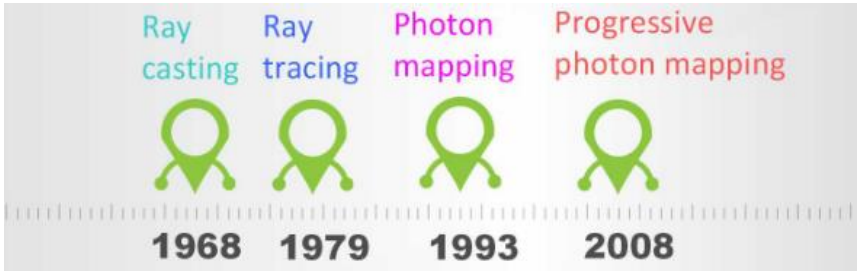
大作业说明

主要考核点

- 基本：实现带有反射、折射、支持三角网格模型导入的光线追踪引擎
- 其他：参数曲面的渲染（Bezier曲面，B样条曲面），光子映射，分布式光线跟踪（景深、运动模糊），算法加速，体积光等等

光线跟踪类别

- 光线投射(Ray casting)　光线跟踪(Ray Tracing)
- 路径追踪(Monte Carlo Ray Tracing / Path Tracing)
- 分布式光线跟踪(Distributed Ray Tracing)
- 光子映射(Photon mapping)
- 渐进式光子映射(Progressive Photon mapping)
- 随机渐进式光子映射(Stochastic Progressive Photon Mapping)

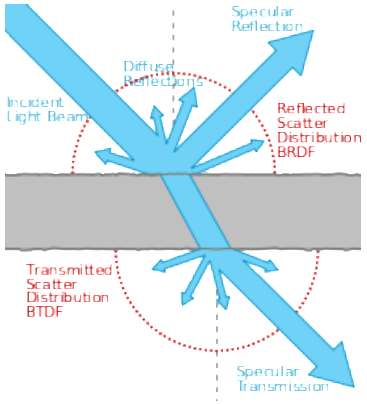


经典的光线跟踪算法(单点光源)

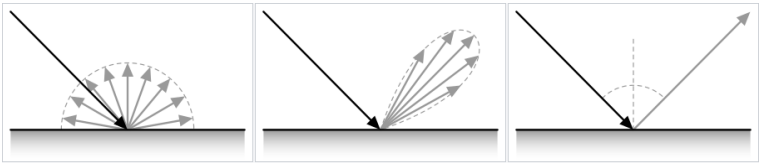
```
RayTracer(Ray, Depth, weight, &Color) {
  Color = 0;
  if ( weight衰减到小于给定值 ) return;  // 递归终止：给定衰减阈值.
  Color = Background;
  计算Ray与场景中最近的物体的交点P;          // 可以采用多种加速方法.
  if ( 没有交点 ) return;                  // 递归终止：返回背景色.
  if ( P非阴影点 )                         // P与光源间是否有物体阻挡.
    Color = 局部光强;                      // 如：书P149，局部光强计算公式.
  if ( Depth > 1 ) {                       // 递归终止：一定次数.
    if ( 当前面是镜面 ) {
      计算反射光线ReflectedRay;
      RayTracer(ReflectedRay, Depth-1, weight*w_r, &RefColor);
      Color += w_r * RefColor;
    }
    if ( 当前面是透射面 ) {
      计算透射光线TransmittedRay;
      RayTracer(TransmittedRay, Depth-1, weight*w_t, &TransColor);
      Color += w_t * TransColor;
    }
  }
}
```

其他算法选择

- Path Tracing
  - Color Bleeding
  - Physically Unbiased



- Distributed Ray Tracing
  - AA, Depth, Glossy, Motion Blur

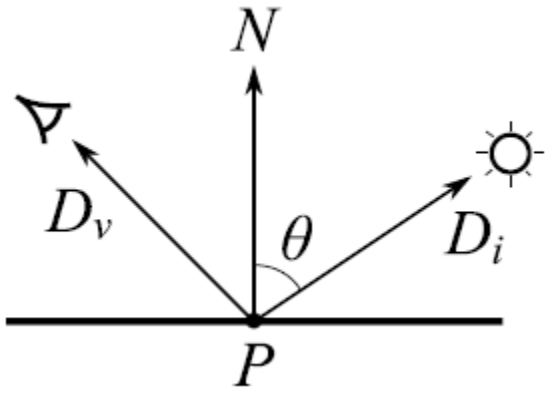


- Photon Mapping
  - Caustics

Path Tracing

- http://www.kevinbeason.com/smallpt/
- 路径追踪不在漫反射表面终止，而是在击中光源时终止
- 光线击中漫反射表面后随机选择出射方向，根据BRDF函数近似计算亮度积分

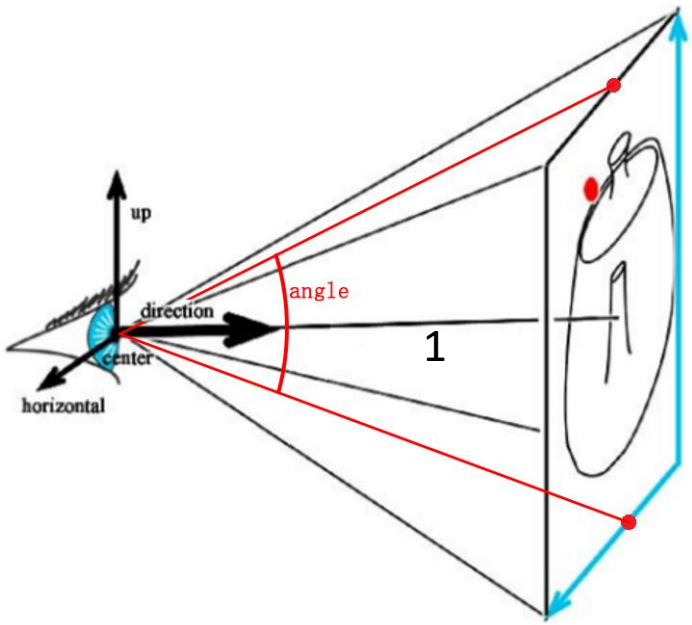
Render Equation



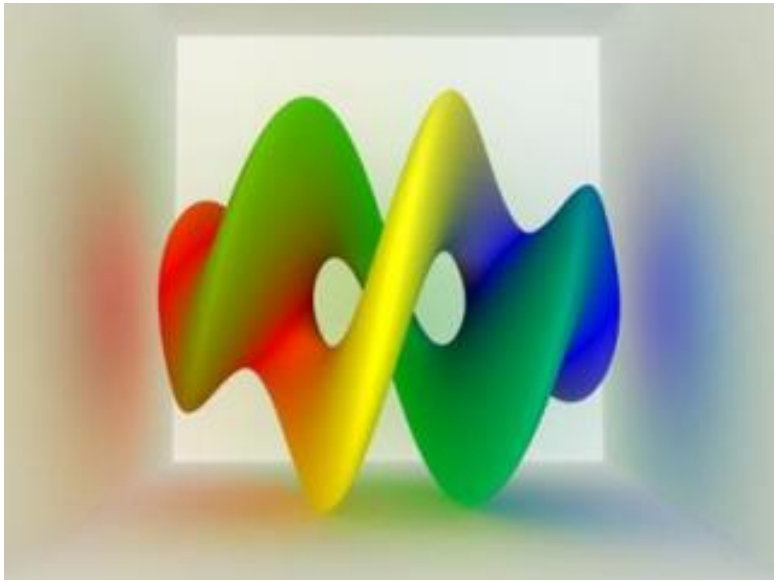
$$L(P \to D_v) = L_e(P \to D_v) + \int_\Omega F_s(D_v, D_i) |\cos\theta| L(Y_i \to -D_i)\, dD_i$$

Path Tracing Algorithm

```
1:  for each pixel (i,j) do
2:      Vec3 C = 0
3:      for (k=0; k < samplesPerPixel; k++) do
4:          Create random ray in pixel:
5:              Choose random point on lens P_lens
6:              Choose random point on image plane P_image
7:              D = normalize(P_image − P_lens)
8:              Ray ray = Ray(P_lens, D)
9:          castRay(ray, isect)
10:         if the ray hits something then
11:             C += radiance(ray, isect, 0)
12:         else
13:             C += backgroundColor(D)
14:         end if
15:     end for
16:     image(i,j) = C / samplesPerPixel
17: end for
```



Path Tracing



Photon Mapping

- http://graphics.ucsd.edu/~henrik/papers/
- 从光源出发发射一定数量的光子（photon tracing），在漫反射表面吸收
- 从视点出发追踪光路（ray tracing），收集终点处一个邻域内的光子以估计该点颜色

Photon Mapping: Construct Photon Map

- 传播
  1. 反射
  2. 折射
  3. 漫反射: 按照 BRDF (对于 Lambert 反射体就是 cos) 的概率分布随机选一个方向
  4. 每次碰到漫反射物体储存位置, 方向跟颜色
  5. 按照概率在反射, 折射, 漫反射以及被吸收中随机选一种.

Photon Mapping: Rendering

- 光线追踪

进行正常的光线跟踪. 但是碰到漫反射物体时, 使用光子图来计算颜色: 寻找碰撞点邻域内的光子, 根据 BRDF 计算这些光子产生的平均色光.

$$L_r(x,\omega) = \frac{1}{\pi r^2}\sum_{p=1}^{N} f(x,\omega_p,\omega)\Delta\Phi_p(x,\omega_p)$$

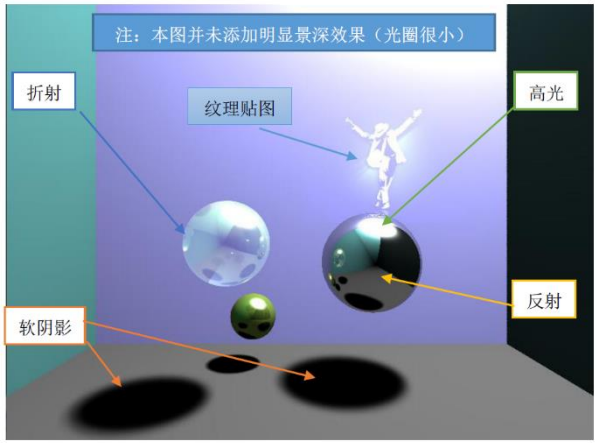
Photon Mapping

- 储存 photon mapping 的数据结构: kd 树 (其他空间数据结构也可)
- 如果场景中漫反射表面特别少, 光子一直反射怎么办: 设置最大撞击次数, 或者俄罗斯轮盘赌
- 需要很多的光子才能达到足够好的效果: Progressive photon mapping (PPM) 和 Stochastic progressive photon mapping (SPPM)

Photon Mapping

要求

- 按时完成，可查阅开源资料和往届文档，不允许裸抄
- CPU上实现
- 至少实现光线跟踪（反射+折射+阴影）
- 图片需要大于480P （640x480），建议使用无损png等格式进行存储。
  - 不建议在多张图上分别实现多个效果
- 严禁使用PhotoShop及其他画图工具进行后处理。
- 不要在最终结果上疯狂注释。
- 不要构造难以辨认的场景。
  - 例如使用纹理贴图贴一张光线跟踪结果。



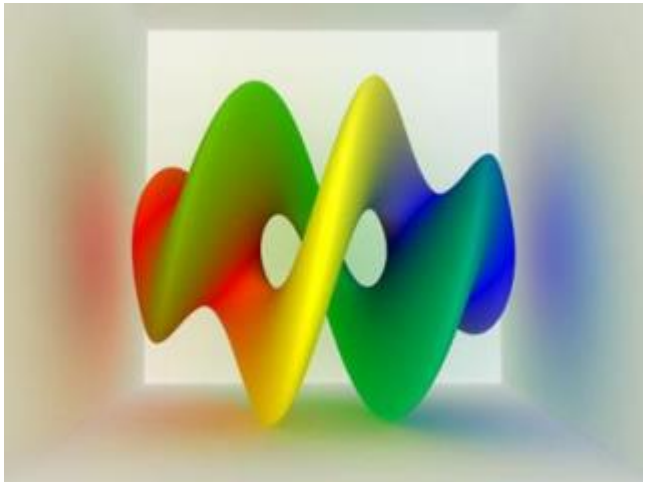
得分项总述

- 算法选型（RT，PT，PM，PPM，……）

- 求交加速（包围盒、层次包围盒、kd-tree、octree、hash）
- 抗锯齿（边缘超采样、DRT等方法，主要针对模型边缘）
- 景深（Adaptive Blur、DRT等方法，模拟相机焦距）
- 软阴影（面光源采样）
- 纹理贴图（UV纹理映射、凹凸贴图）
- 复杂网格模型/场景（网格读写、求交加速、法向插值等）
- 其他（体积光等）

1. 算法选型

- 仅需实现一种渲染法。
  - 算法难度和运算量逐步提高。
  - 如果选择了较新的算法，应该体现出比老算法强在哪里。
    - 做Path Tracing： 漫反射，特殊材质下的Color Bleeding。
    - 做Photon mapping：Caustics – 由反射/透射引起的漫反射。
    - 做Progressive Photon mapping ： 高精准度。



2. 时间代价

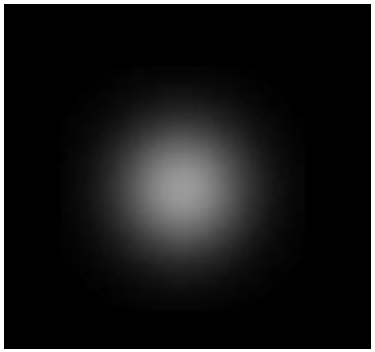
- 最终评分主要看效果，做的加速(包围盒、包围球、Octree、Kd-tree)请在报告中注明。

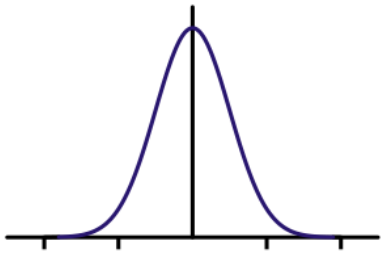
- 参数曲面（尤其是高次）和复杂网格求交的时间代价比一般曲面要高很多，完成作业时注意留足渲染时间。

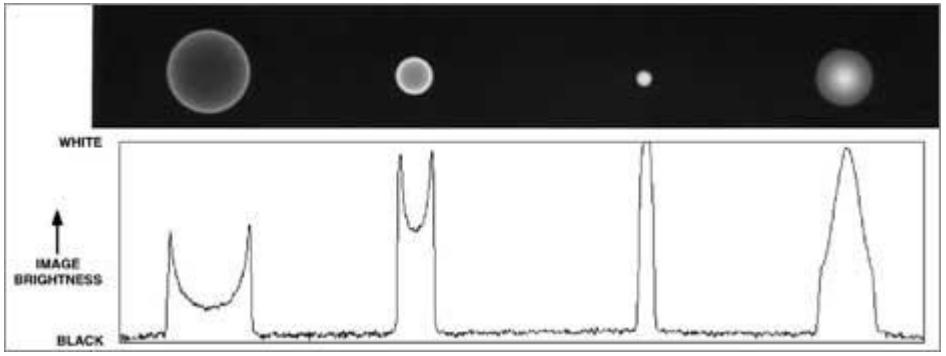
- 多项附加功能会导致渲染时间乘法式叠加，夜里挂程序请保护计算机的安全。

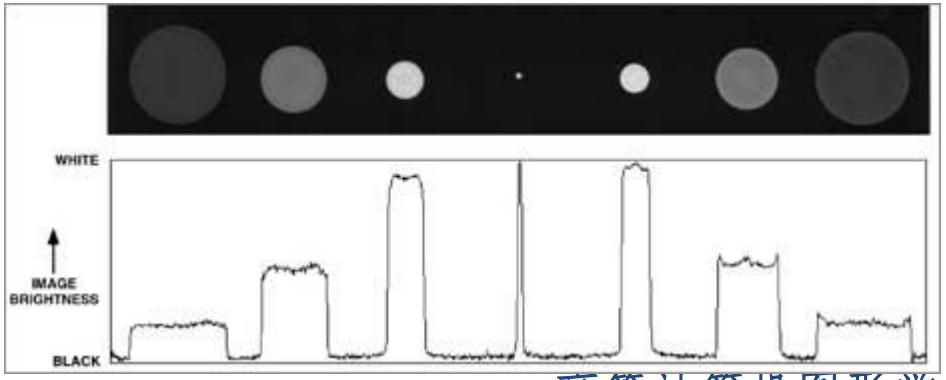
3. 景深

- 景深三要素：光圈、焦距、物距



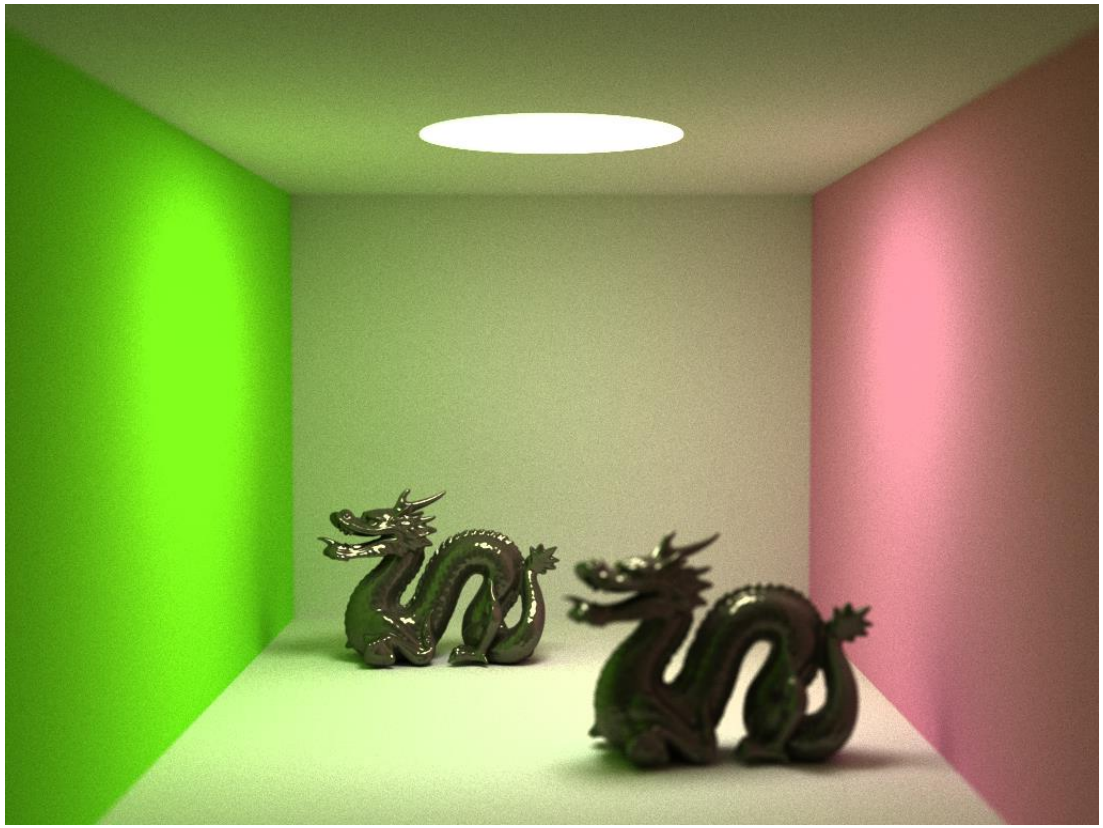






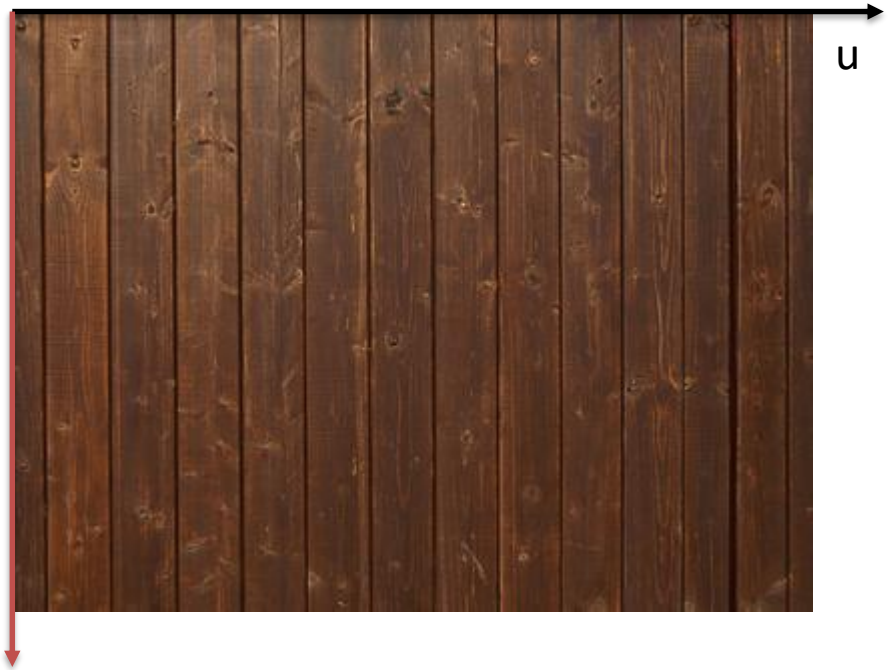
3. 景深

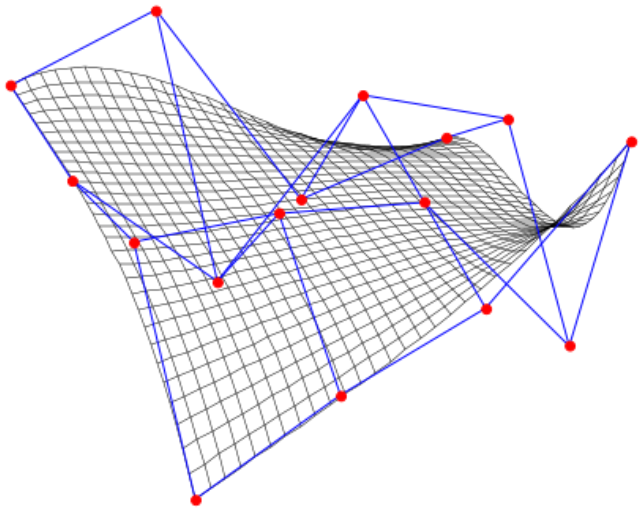
- Path Tracing带景深



4. 纹理映射

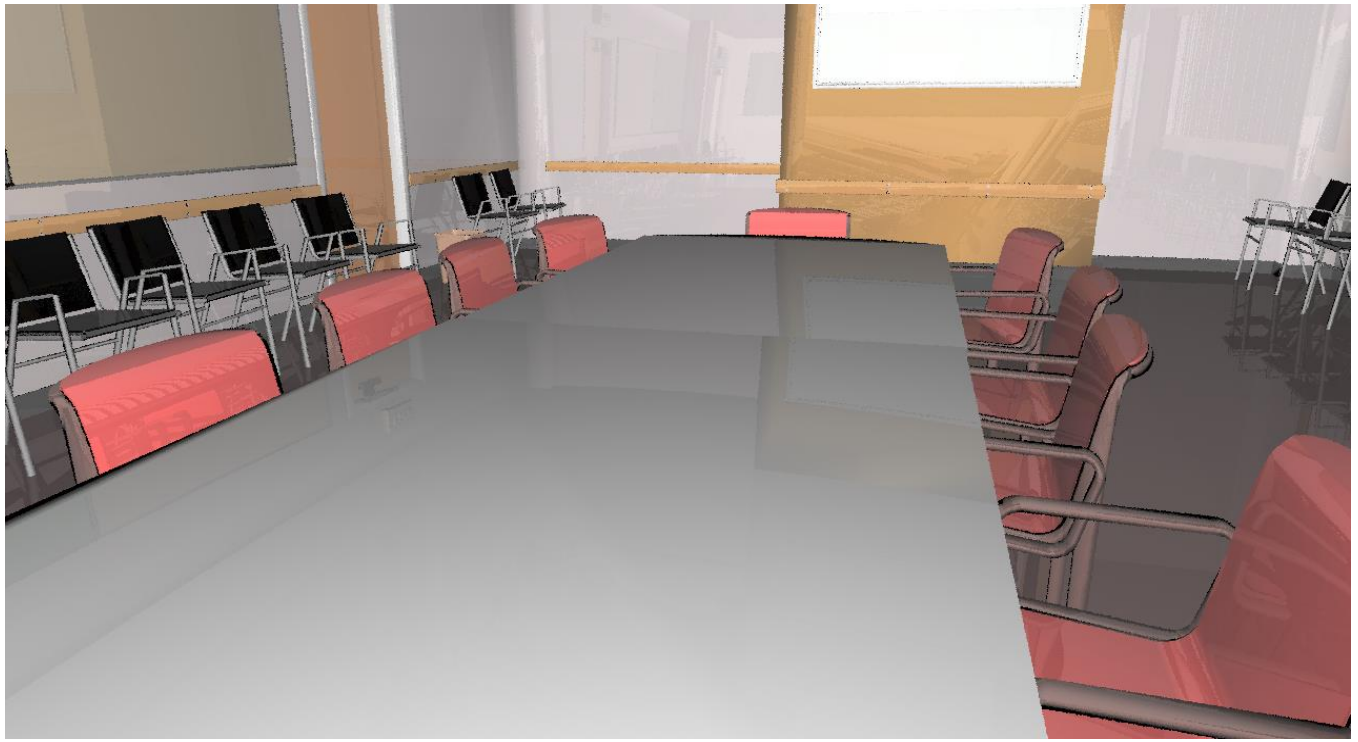
- 与参数曲面求交时，UV纹理坐标和求出来的参数值完美对应。
- 一般的面片，也可以将求交结果转化为参数坐标后读出需要取的纹理颜色值。





5. 复杂网格/场景

- 建议做1~2个参数曲面或复杂网格模型即可。



X. 体积光

- 纹理与体积光

最终提交和验收

- Result = 一张/多张渲染好的图片
- Code
- Report
  - 开头应明确列出自己的所有得分点，避免遗漏。
  - 再分别列出每个得分点对应的代码段，还可以辅以说明，方便查验。
  - 报告最后再附上实验结果图片。

- 第18周工作日进行当面验收，原则上不允许远程验收和推迟。
  - 验收以交流为主，验收时用的图像不作为最终评分标准
  - 之后可以补交最终结果。
- 第18周周日晚上23:59前提交。

评分细则

- 占总评45分。
  - 实现光线追踪（反射、折射、三角网格，有明显bug扣分）
  - 实现光子映射 / 渐进式光子映射
  - 景深 / 软阴影 / 抗锯齿 / 贴图 / 凹凸贴图/运动模糊等（路径追踪得分含在内）
  - 实现参数曲面解析法求交
  - 复杂网格模型
  - 主观分: 设计和构图
  - 其他额外效果: 体积光、色散、体渲染等

- 评分
  - 实现的功能：多写不扣分
  - 功能的完成度和呈现质量：影响每项功能的具体得分
  - 整体渲染效果
  - 实验报告：按要求描述清楚即不会扣分

Thank You ！
Any Questions?