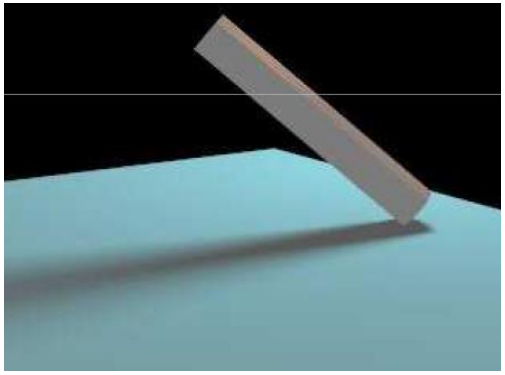
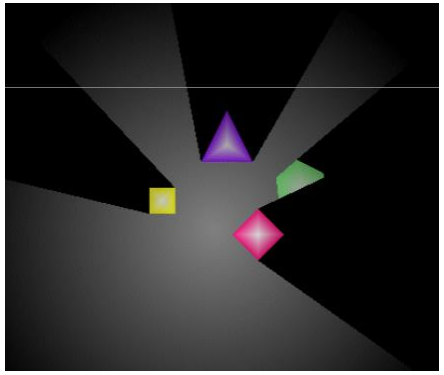
第五章 真实感绘制

5.2 阴影、镜子、环境映射

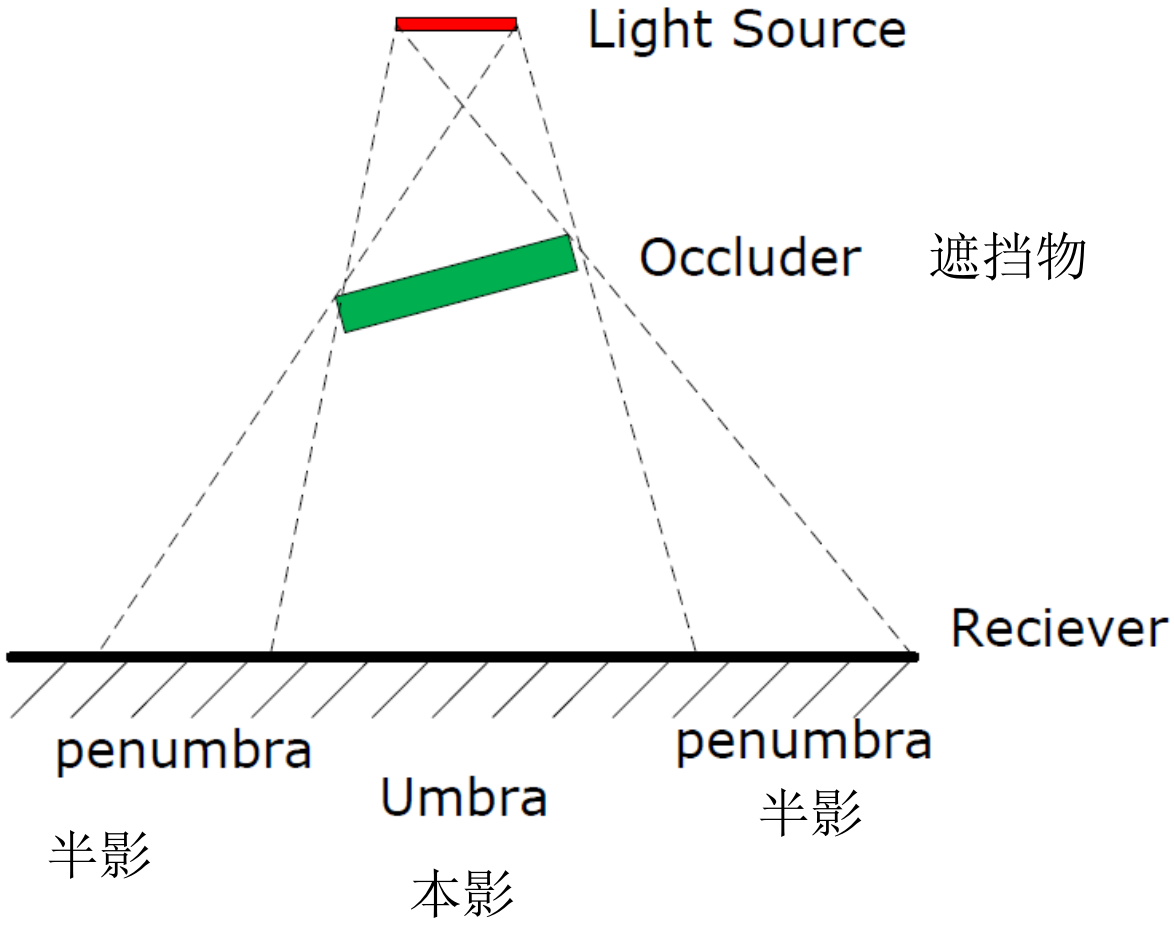
- 阴影（shadow）
- 包含镜子的场景（mirroring）
- 环境映射（environment mapping）

阴影

- 场景中被遮挡了光源的部分
- 场景中物体A上的阴影如果是由物体B造成的，则该阴影是B在A上基于光源点的投影
- 物体有明确的边缘（hard edges），而面光源所投射的影子有柔性边缘（soft edges）

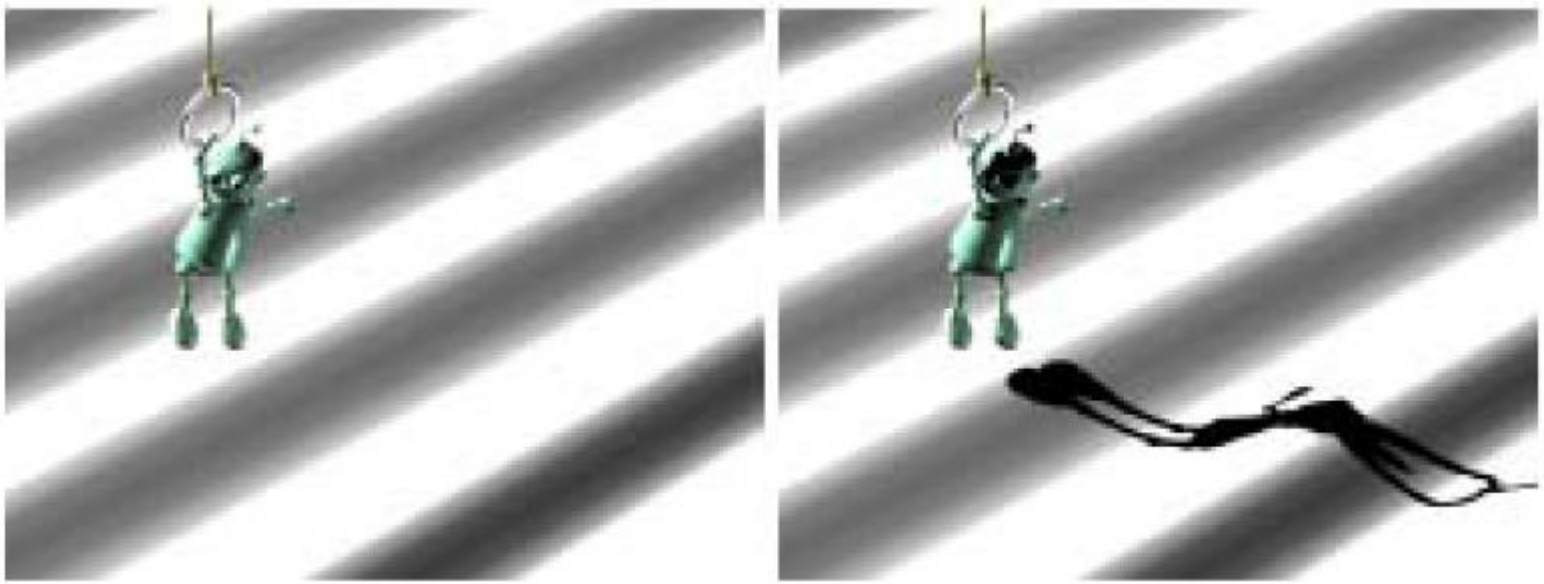




阴影


Light Source
Occluder
遮挡物
Reciever
penumbra
半影
Umbra
本影
penumbra
半影

阴影的重要性

- 能够帮助理解被投射阴影的物体的几何形状



阴影的重要性

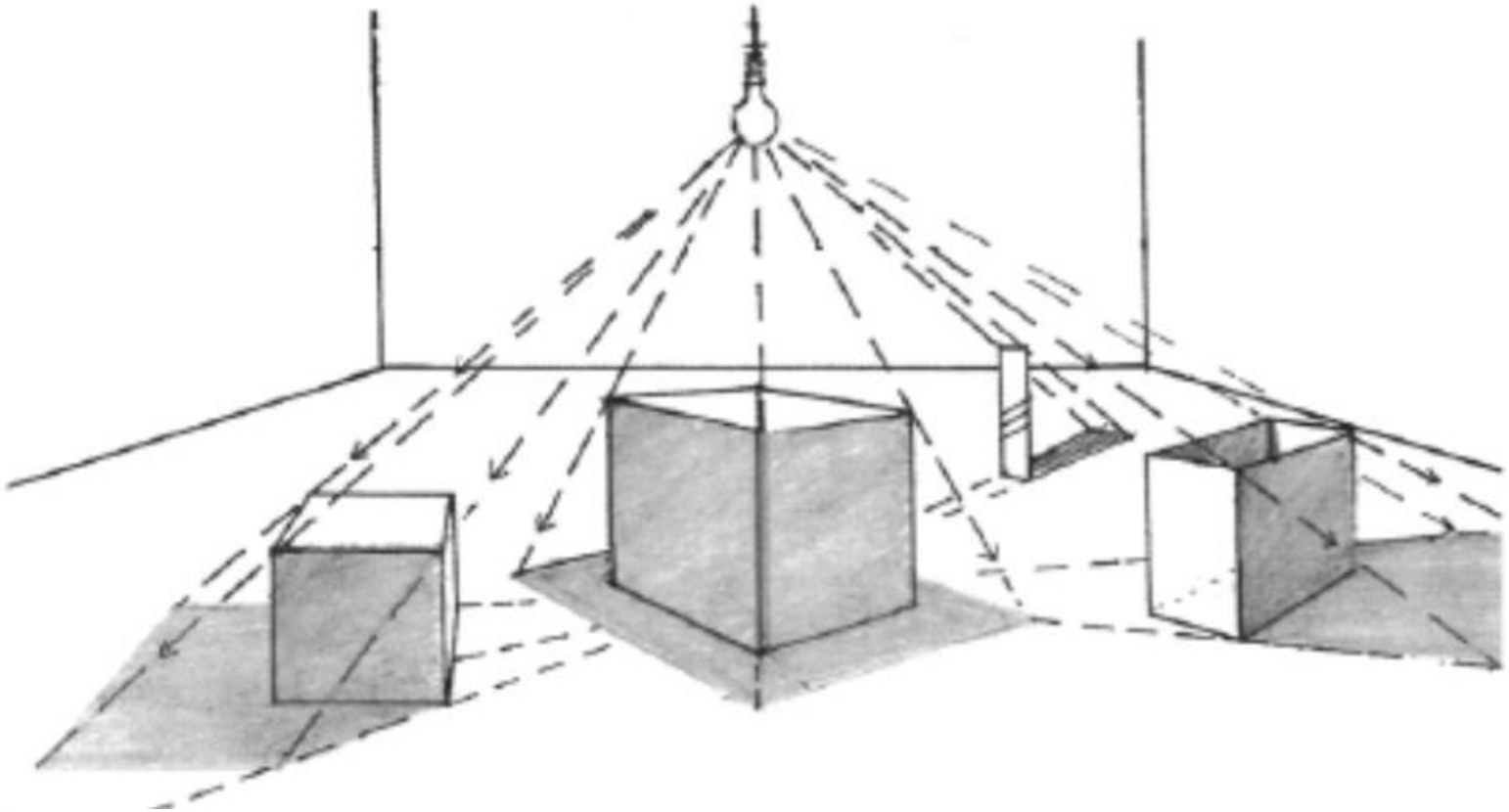
- 能够帮助理解遮挡物的几何形状

阴影的重要性

- 能够帮助理解相对位置

阴影的重要性

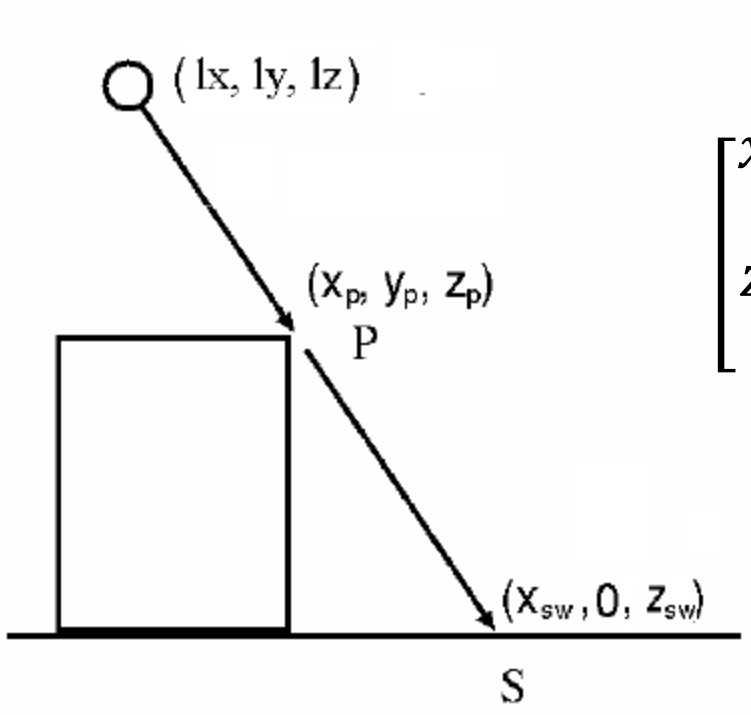
点光源的平面阴影



https://www.oreilly.com/library/view/basic-perspective-drawing/9780470288559/12_chapter08.html

点光源的平面阴影

- Blinn ’88 提出了用齐次变换矩阵求局部点光源的阴影的方法
- 将遮挡物变换成阴影

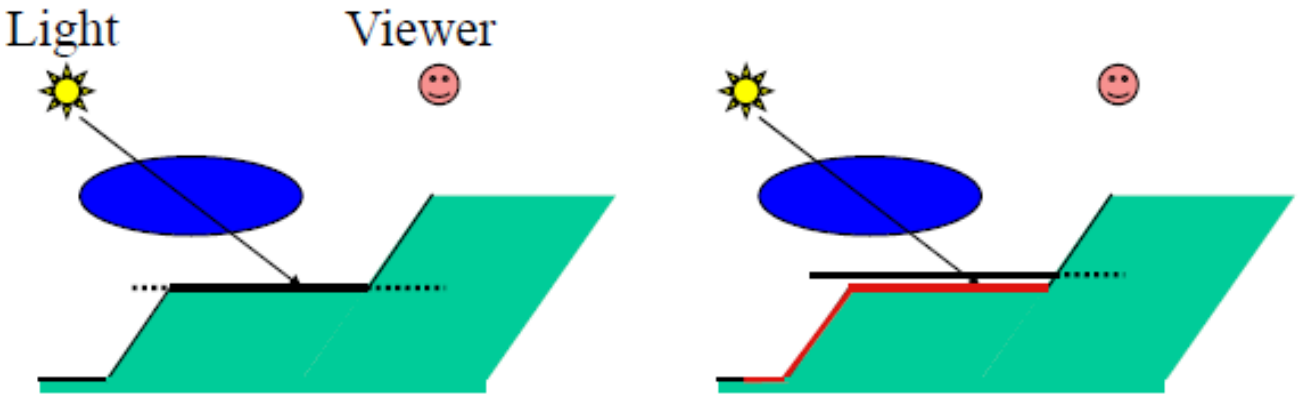


$$\begin{bmatrix} x_{sw} \\ 0 \\ z_{sw} \\ 1 \end{bmatrix} = \begin{bmatrix} l_y & -l_x & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & -l_z & l_y & 0 \\ 0 & -1 & 0 & l_y \end{bmatrix} \begin{bmatrix} x_p \\ y_p \\ z_p \\ 1 \end{bmatrix}$$

绘制平面上的阴影

- 绘制遮挡物
- 将阴影矩阵乘以物体的变换矩阵，得到阴影的形状
- 使用灰色绘制阴影

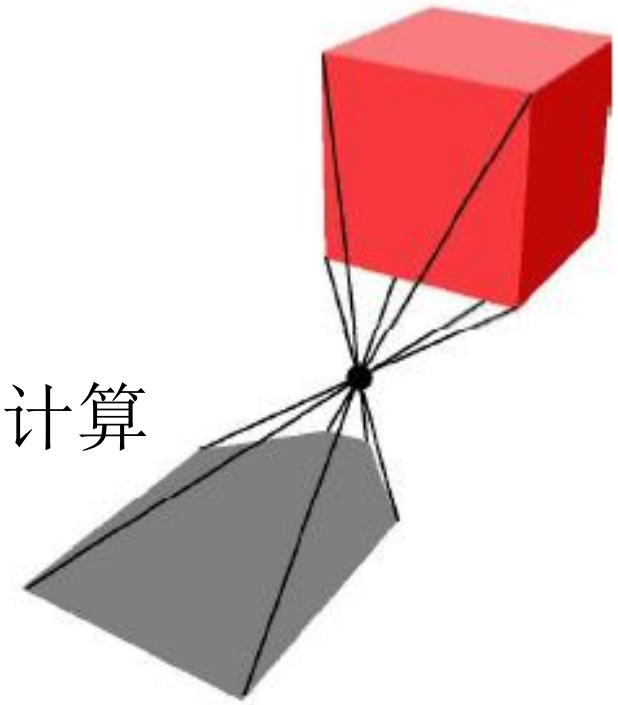
-小技巧：将阴影抬高一点点以避免z-buffer的错误



平面阴影的局限

- 投射面必须是平面
- 阴影必须逐帧渲染（即便阴影没变化）

- “假”阴影
  - 如果光源位于物体下方，将计算出假阴影



阴影纹理（shadow texture）

- 将阴影图像（shadow image）看做一张纹理贴图
- 从光源的角度给遮挡物绘制阴影图像
- 可以阴影投射到非平面的表面上

from light　　shadow texture　　shadows on stairs

阴影纹理（shadow texture）

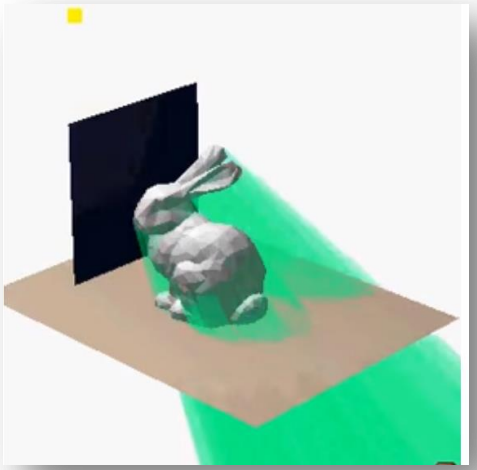
- 优点：
  - 如果接收面没有变化，则不需要重新计算阴影
  - 可以在非平面上投射阴影
- 缺点？
  - 物体不能在自己身上投射阴影

立体阴影（shadow volume）

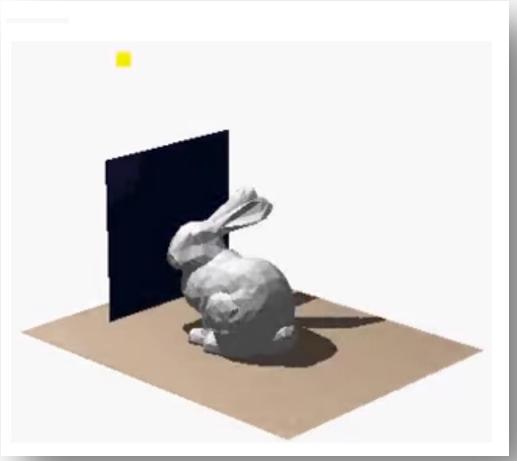
- 也叫volume shadows
- 可以在任意物体上投射的阴影

立体阴影

- 在真实世界中，光源被物体遮挡的位置构成一个立体的形状，而不仅仅是一个二维形状



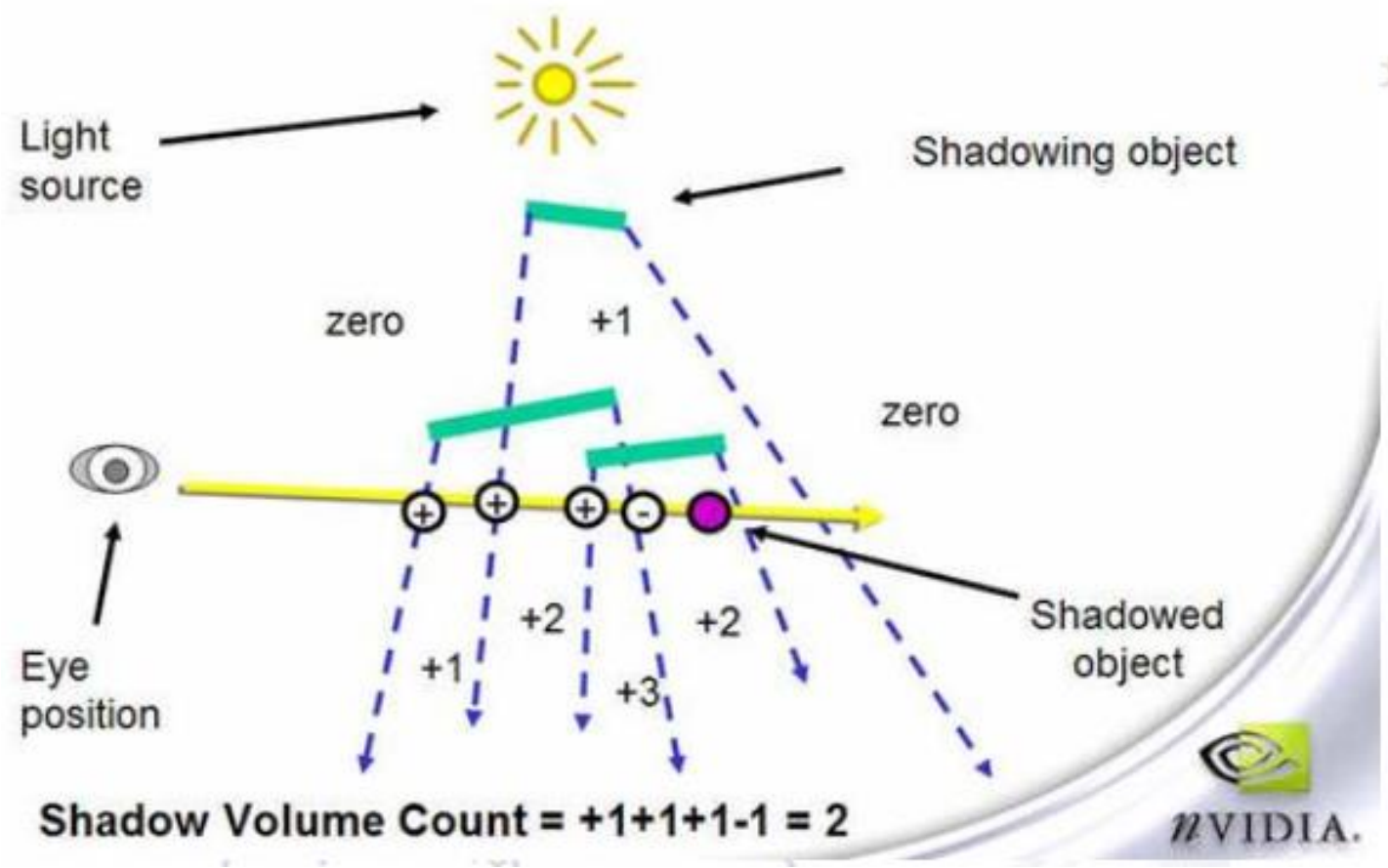
- 立体阴影就是使用“体”（volume）来模拟阴影



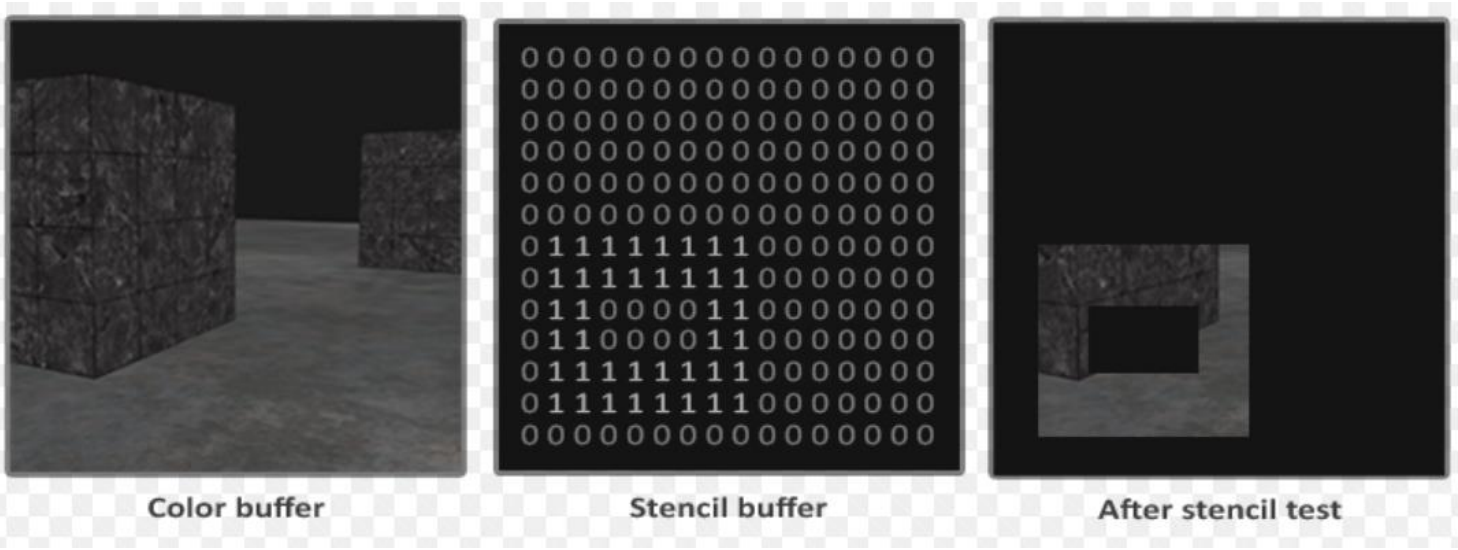
立体阴影的渲染过程

- 两个步骤：
  - 给定光源和投射阴影的物体
  - 每个投射阴影的三角形产生一个阴影体
  - 判断某一点是否在阴影体内
    - 如果在：是阴影
    - 如果不在：用光照模型进行渲染

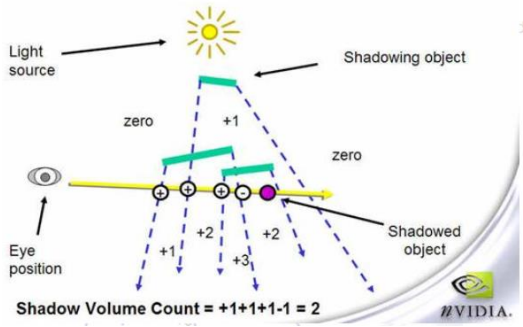
判断是否处于阴影体之中



如何实现？Stencil buffer



立体阴影的实现



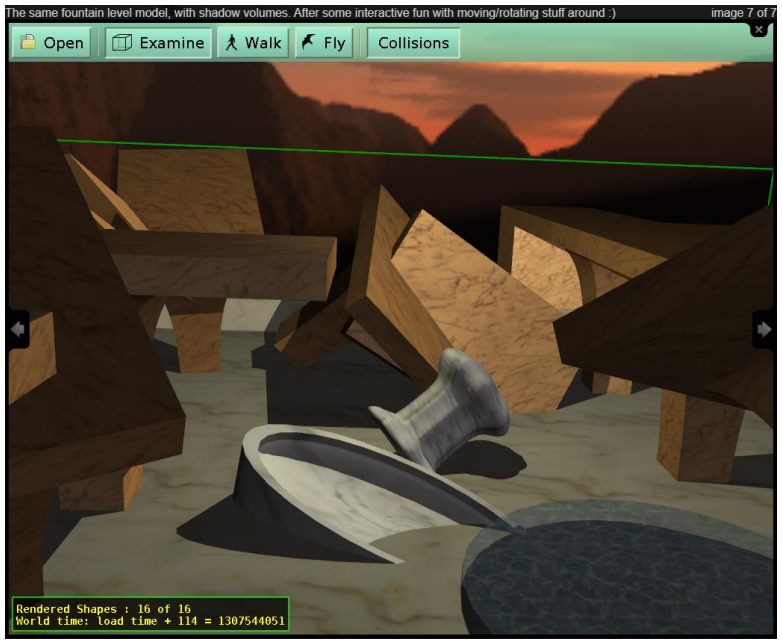
- 获得整个场景的depth map（z-buffer）
- 获得所有阴影体的边界多边形
- 仅用环境光渲染整个场景
- 清空stencil buffer
  - 打开back face culling, 对于每一个阴影体多边形，如果该多边形通过depth test，则stencil value+1
  - 打开front face culling, 对于每一个阴影体多边形，如果当前多边形没有通过depth test，则stencil value-1
- 根据每个像素的stencil value来判断其是否在阴影中（stencil value = 0 在阴影外）
  - 重新使用漫反射、镜面反射渲染stencil value=0的像素

立体阴影

- 优点
  - 更加通用，不用手动指定投影和被投影的物体
  - 容易实现：stencil buffer
  - 可以实现自投影
  - 准确，避免了image based method的不准确的缺点
- 缺点？
  - 当有大量的阴影体时，光栅化（rasterizer）成为瓶颈
  - 视点不能在阴影内

立体阴影

- 例子



https://castle-engine.io/shadow_volumes

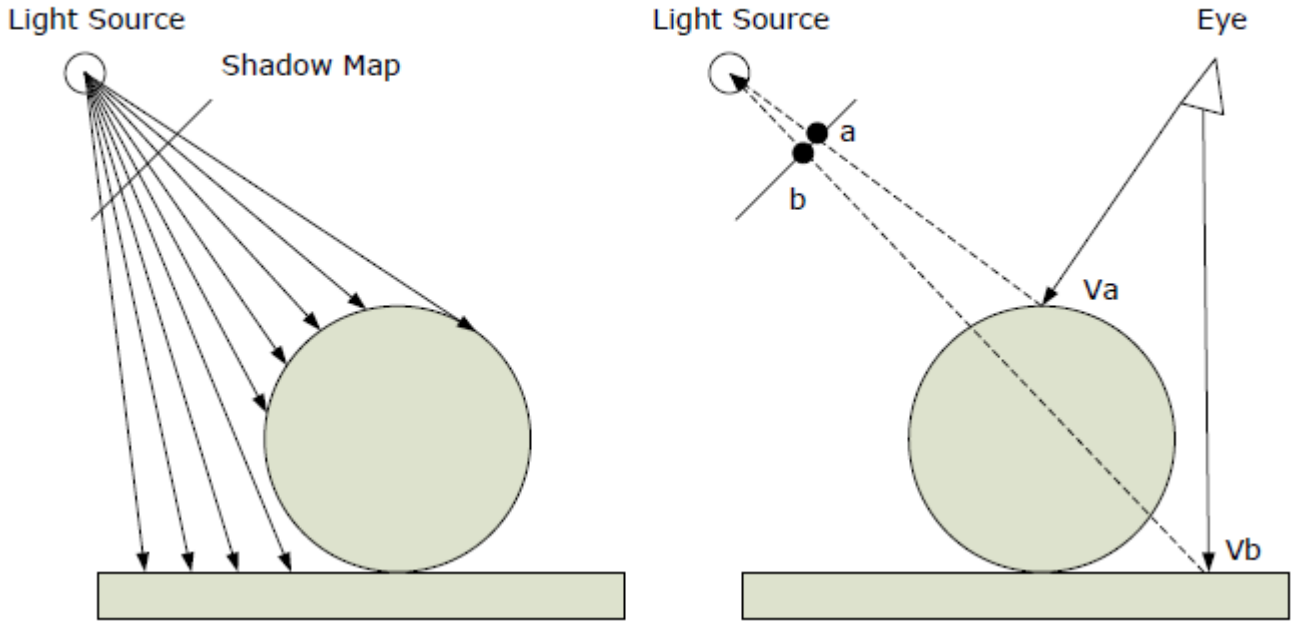
阴影贴图（shadow map）

- 从光源的角度使用Z-buffer
  - Z-buffer存储的是物体到光源的距离
  - 也叫shadow buffer

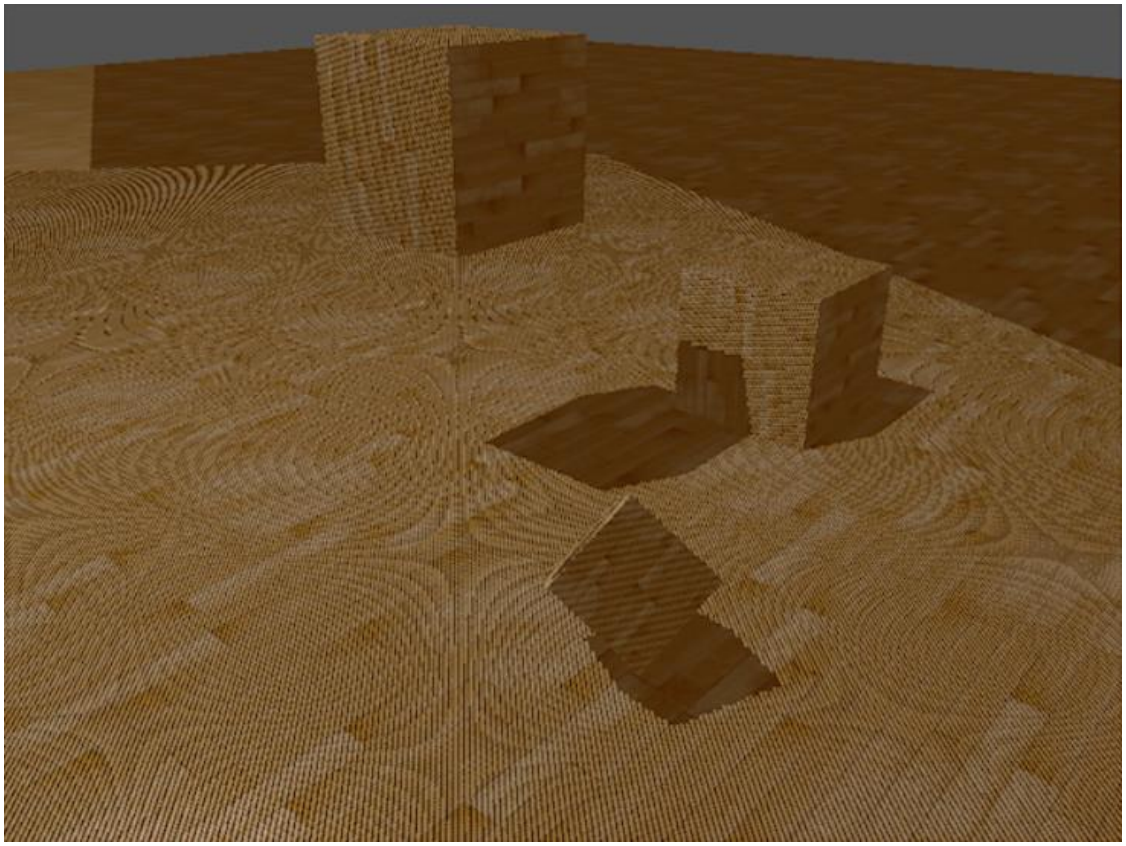
- 从摄像机角度渲染
  - 对于每一点检测其是否在阴影中

阴影贴图（shadow map）

- 检查Va和Vb到light的距离是否大于shadow map中的值

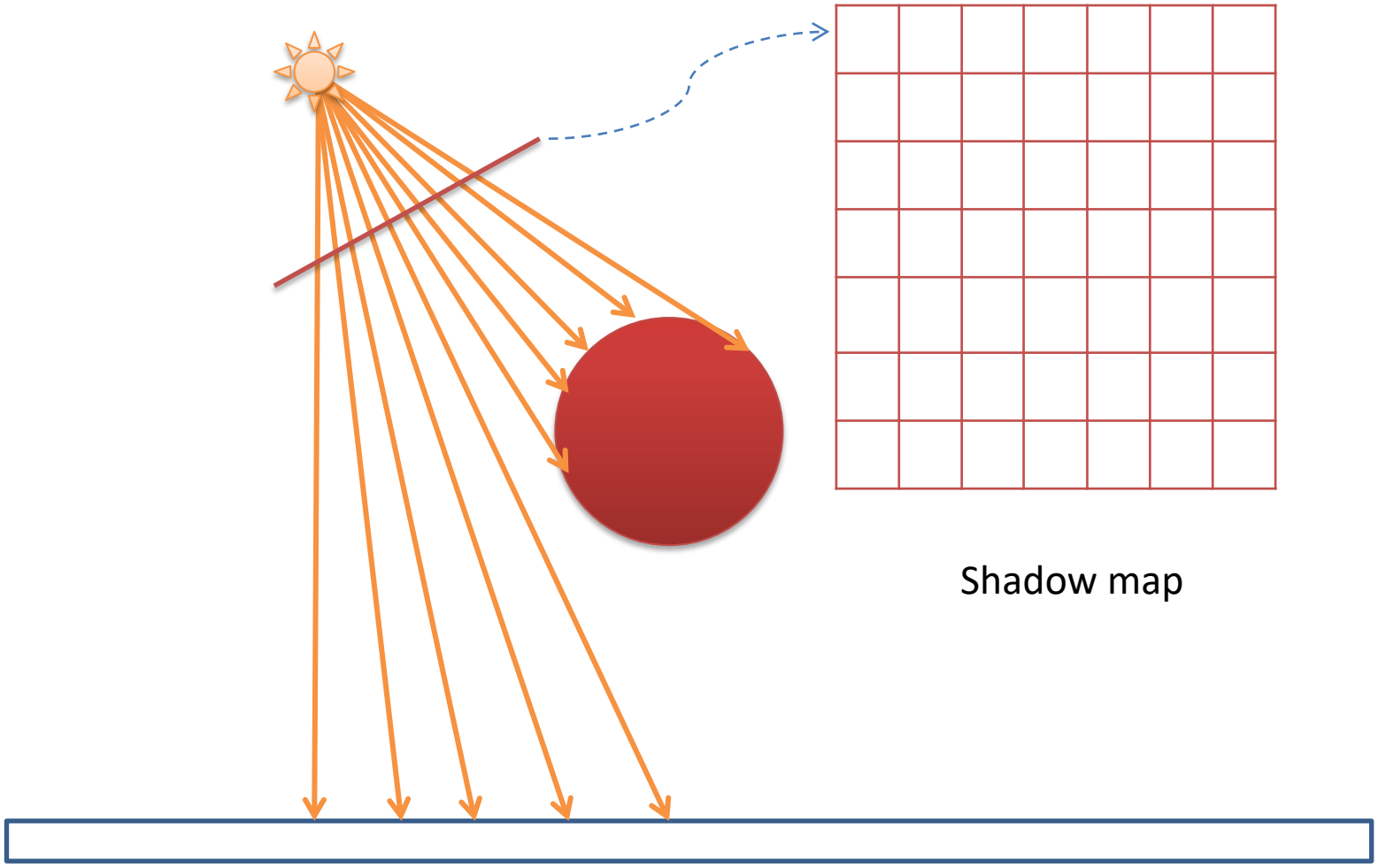


阴影失真（Shadow Acne）

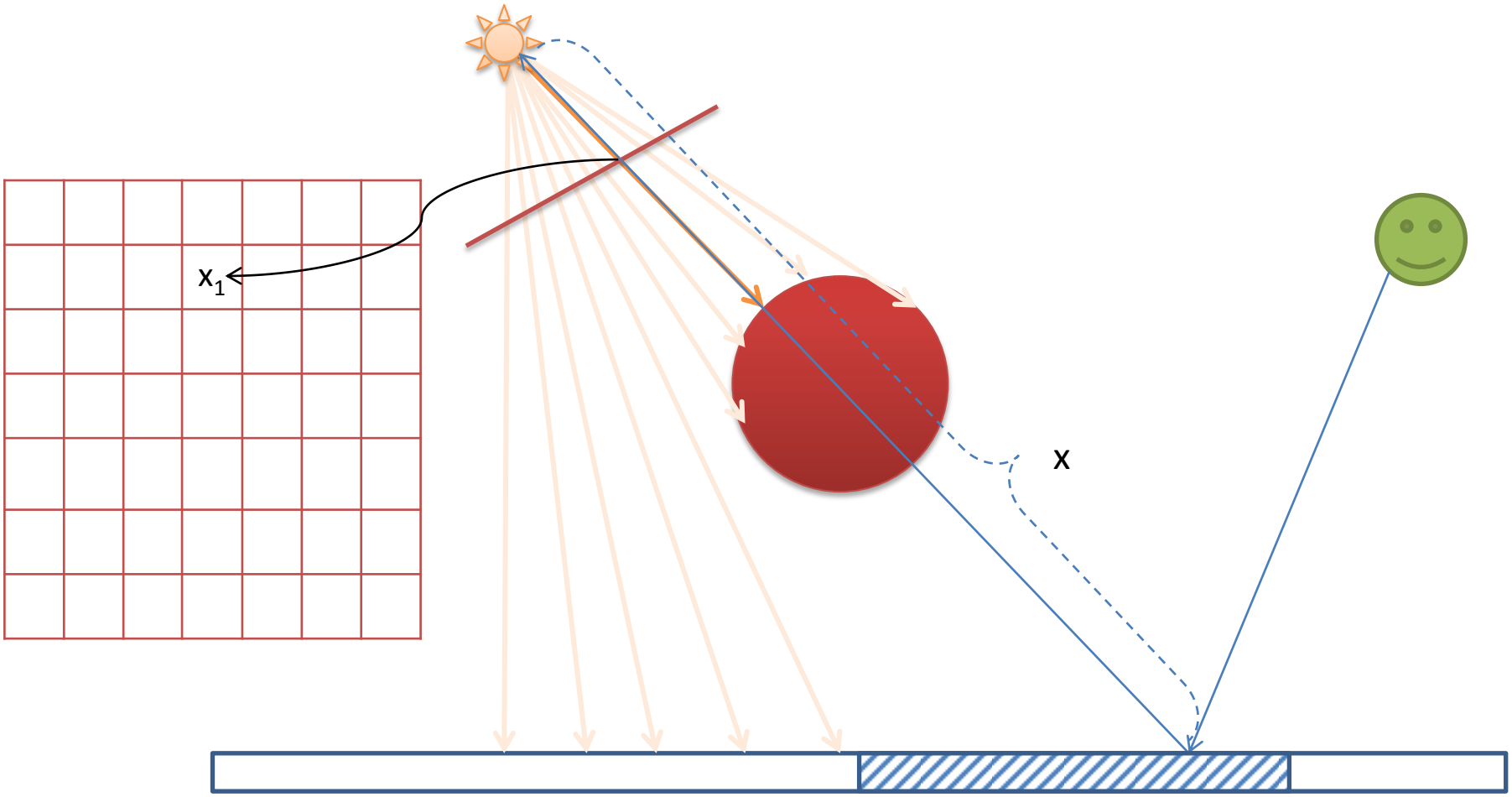


https://learnopengl-cn.github.io/ 视频3

阴影失真（Shadow Acne）



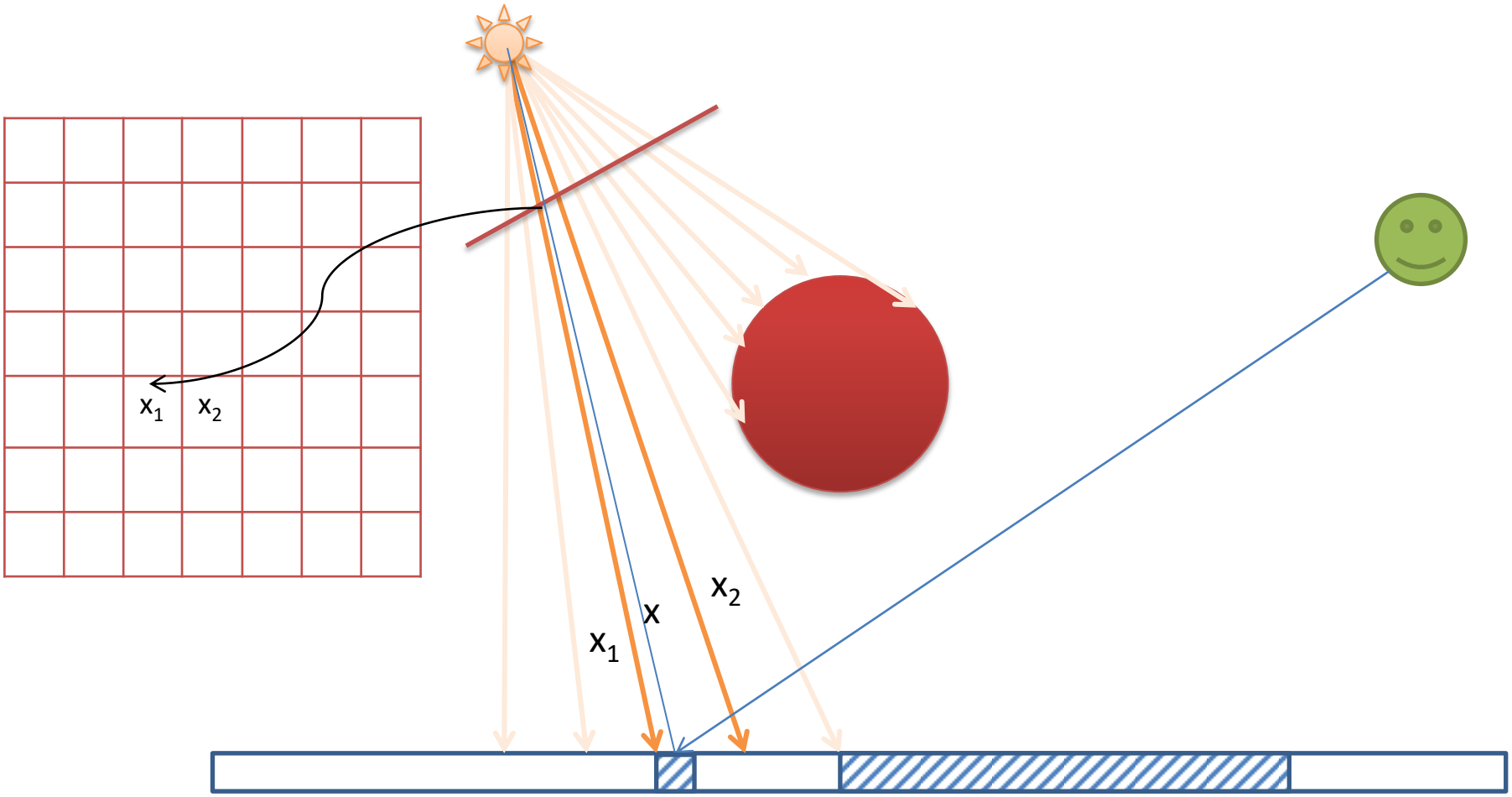
阴影失真（Shadow Acne）



$x>x_1$

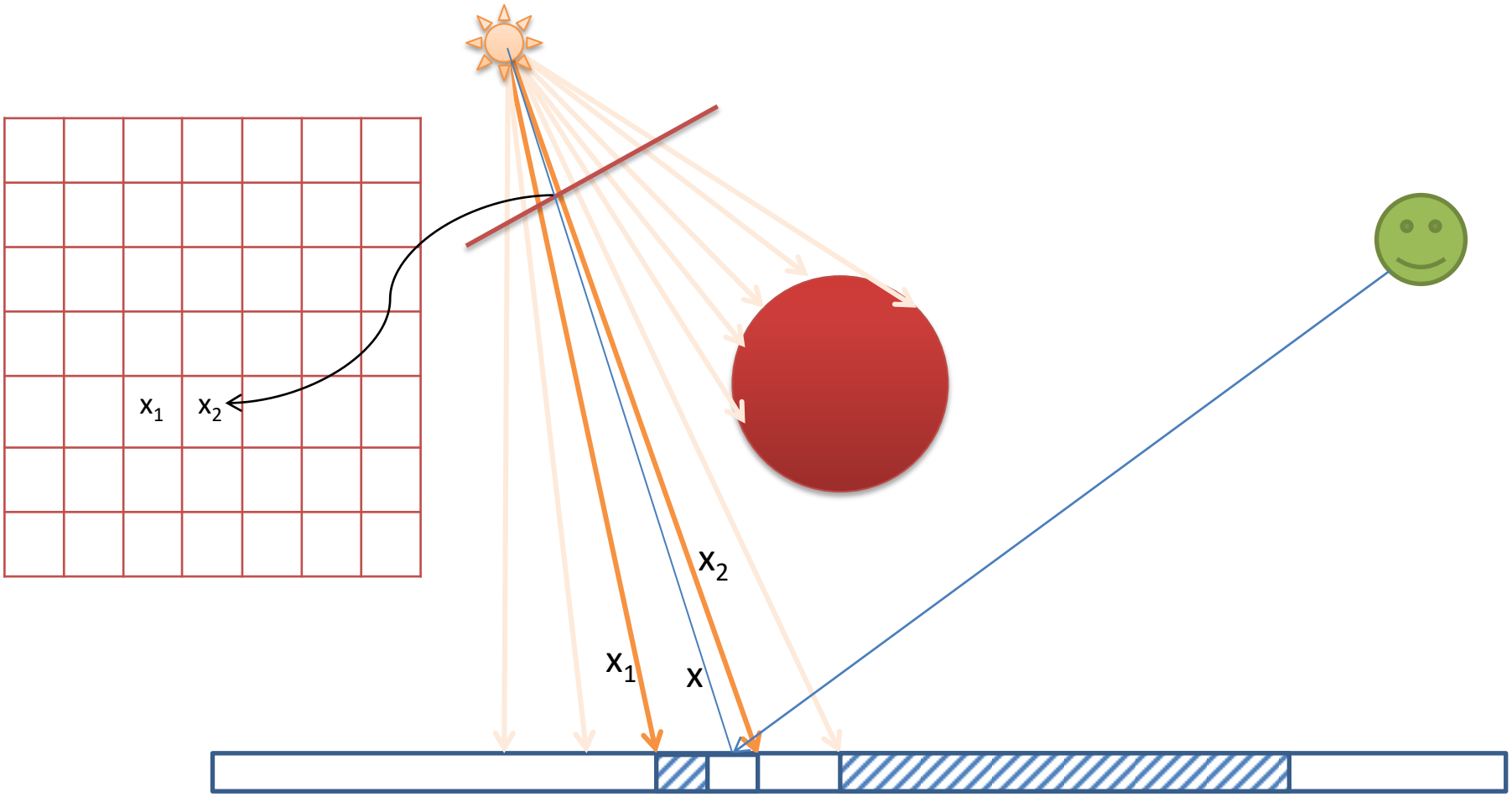
所以该点处于阴影中

阴影失真（Shadow Acne）



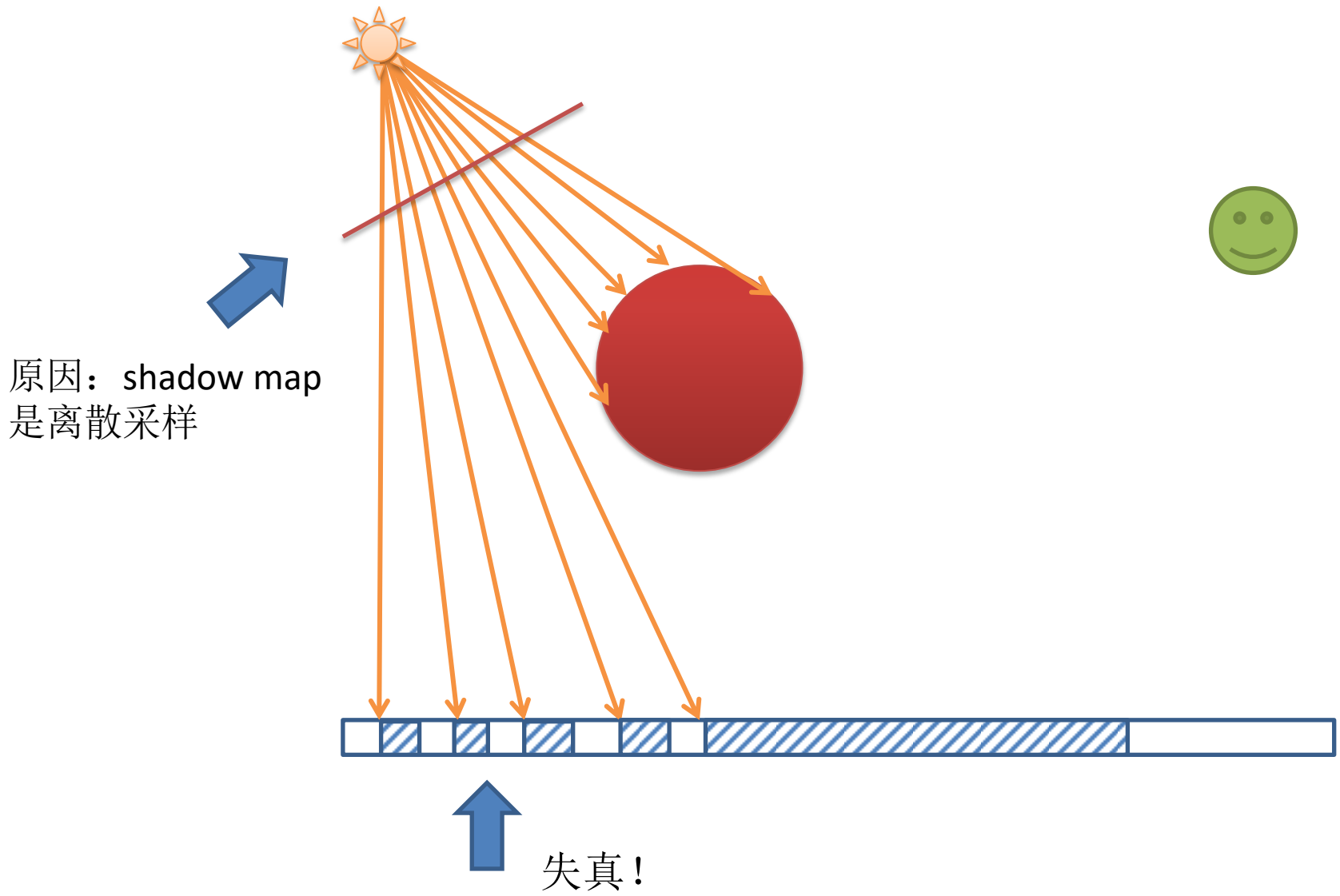
当$x_1<x<x_2$, 且更靠近x_1
查表查到x1，因为x>x1, 所以被认为是阴影

阴影失真（Shadow Acne）

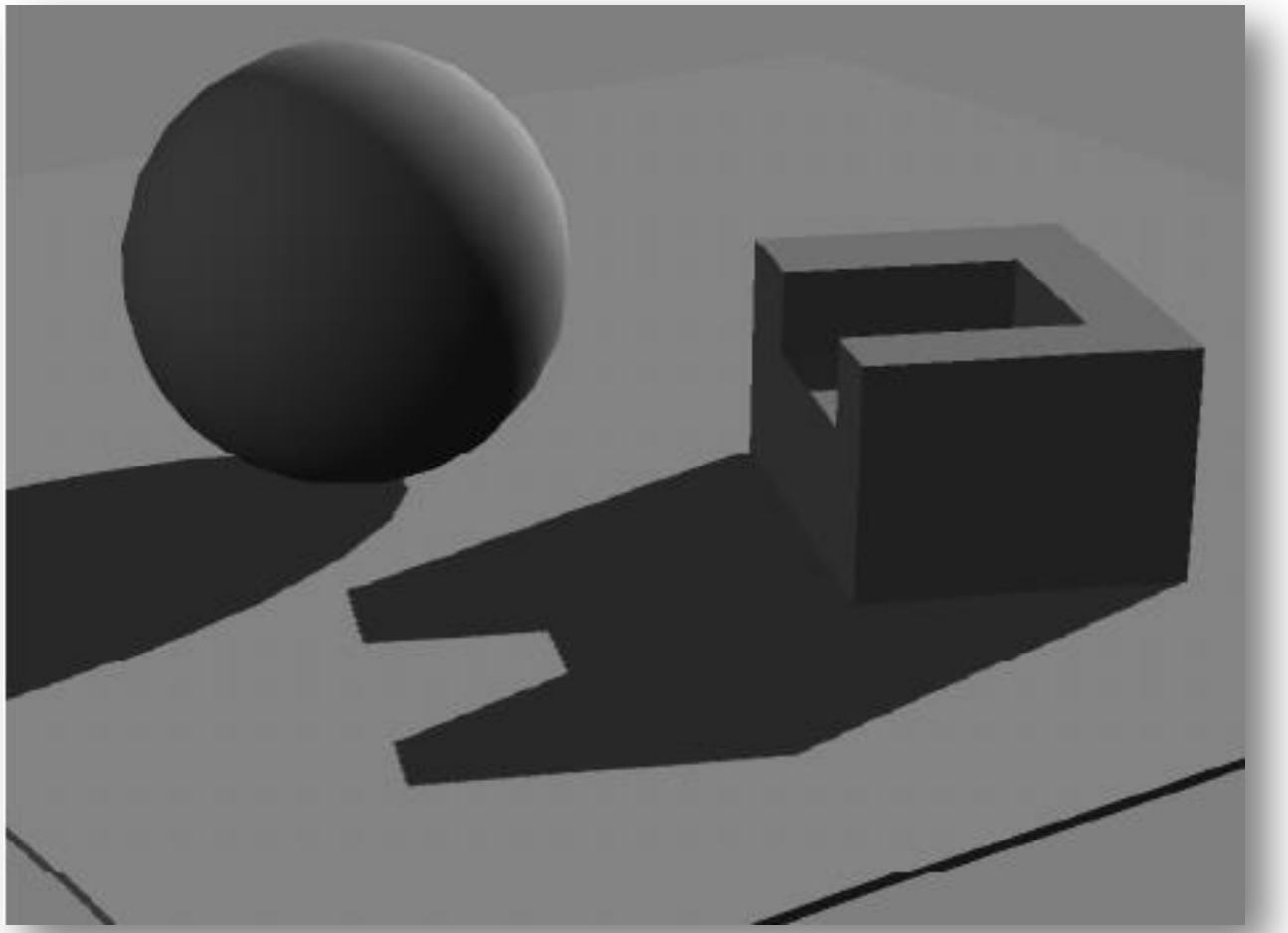


当$x_1<x<x_2$, 且更靠近x_2
查表查到x_2，因为$x<x_2$, 所以被认为不是阴影

阴影失真（Shadow Acne）

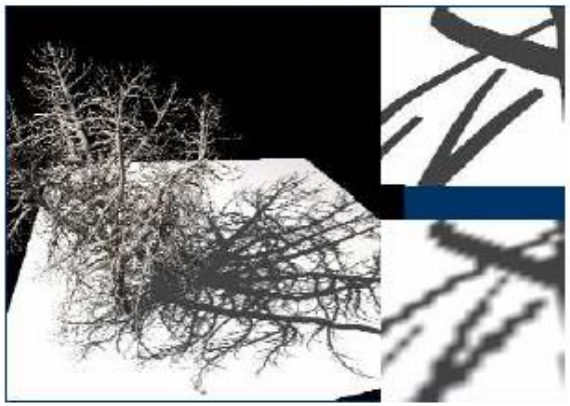


阴影贴图（shadow map）例子



阴影贴图（shadow map）

- 优点
  - 阴影量大的时候比立体阴影速度快（为什么？）
  - 硬件支持
- 缺点
  - 精度低

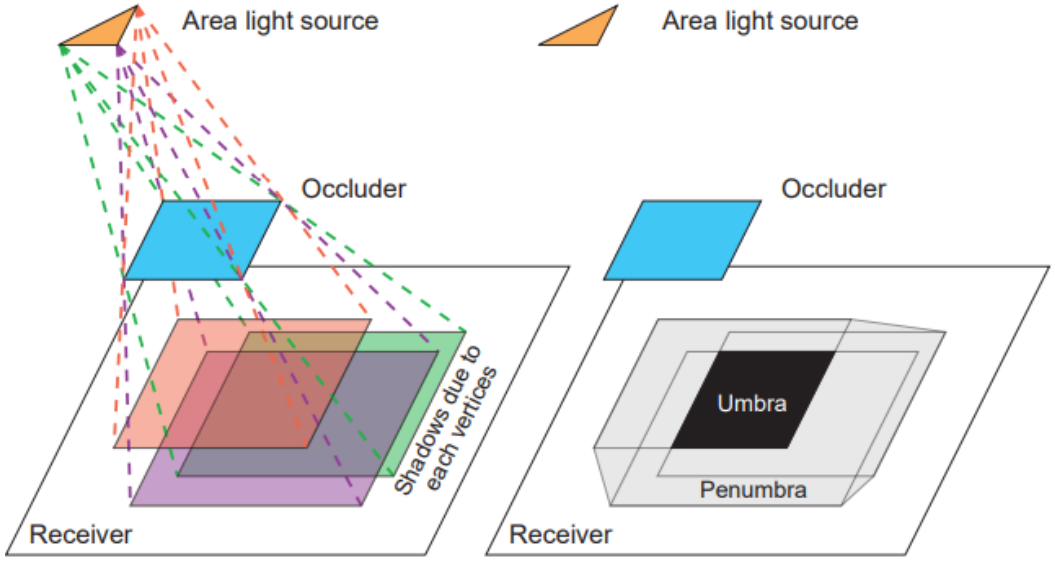


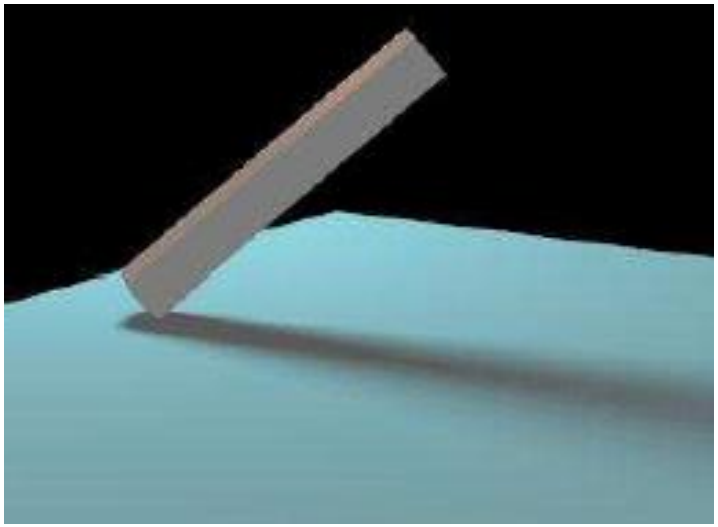
小结

- Shadow map 和 shadow volume都是最常见的阴影计算方法
- Shadow map 用的更多
- 这两种技术都有很多变种
- 更多信息
  - http://www.realtimerendering.com/

柔性阴影（soft shadows）

- 由区域光源（面光源）产生
- 可用多个点光源近似模拟





多个hard shadow叠加

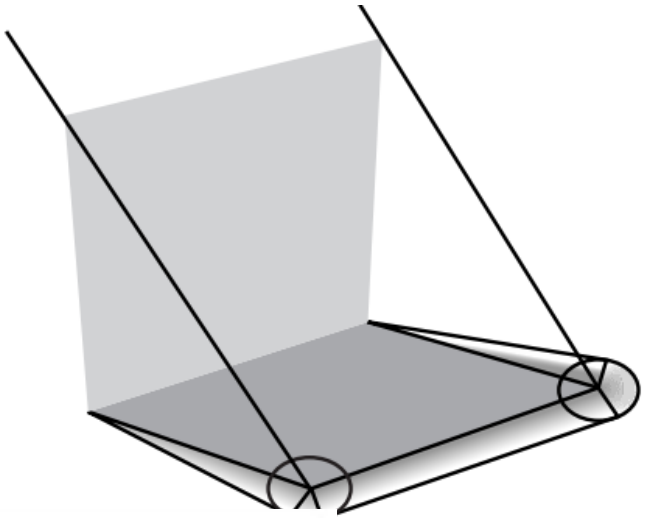
- 需要将每个点光源的效果做叠加
- 需要足够的点光源数才能产生好的效果
- 渲染的次数取决于点光源数
- 比较慢

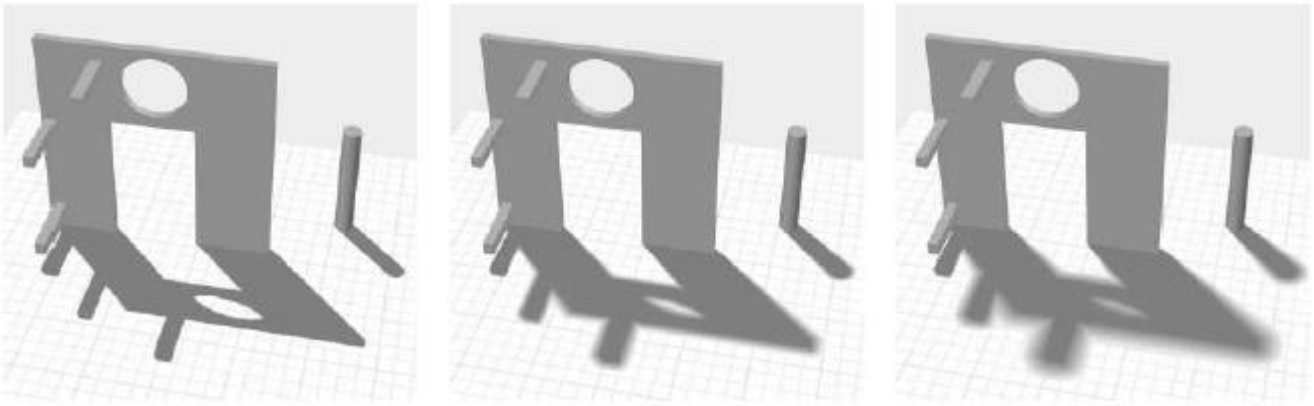
4 samples

1024 samples

Plateaus 方法 [Haines, 2001]

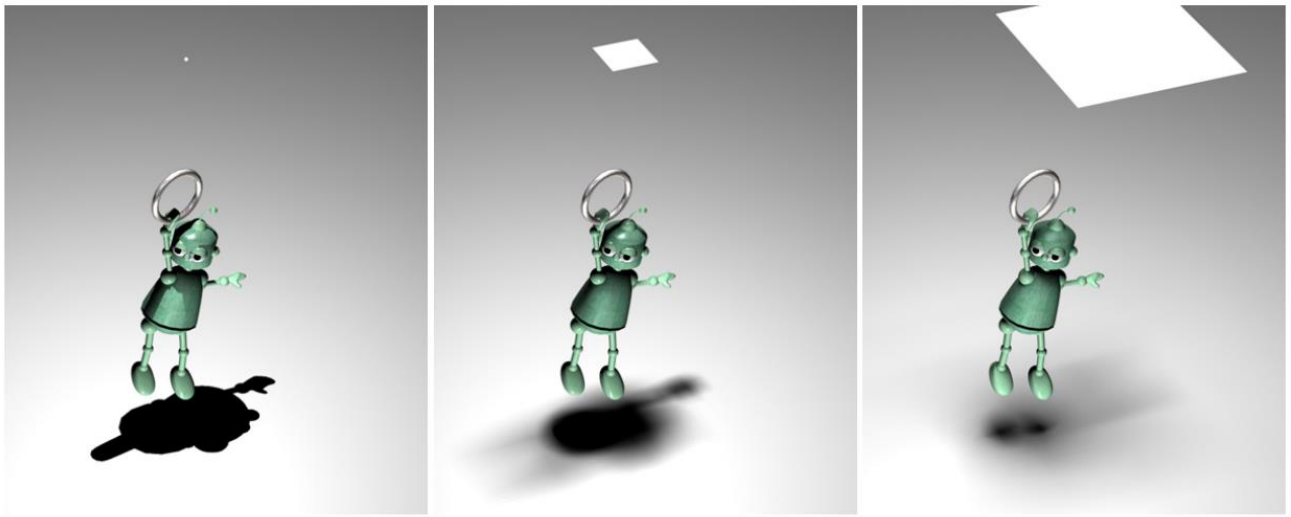
- 先计算hard shadow
- 在便阴影边缘处进行模糊
- 阴影所对应的物体上的点位置越高，则模糊半径越大
- 只能用于平面上阴影的计算





其他方法

- Soft shadow volume
- Moving the projected plane up and down instead of moving the light Source
- ….
- “A Survey of Real-time Soft Shadows Algorithms”



内容提要

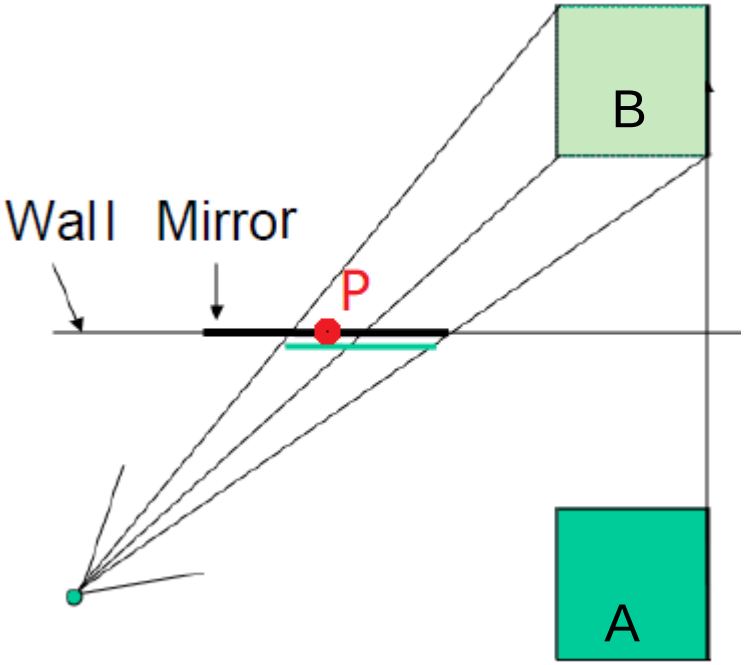
- 阴影（shadow）
- 包含镜子的场景（mirroring）
- 环境映射（environment mapping）

渲染镜子中的物体

镜像（Planar Reflections）

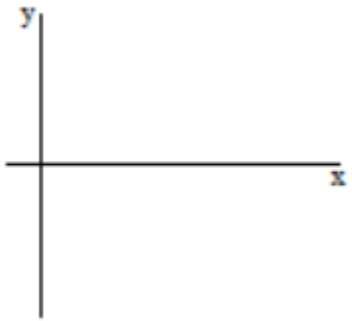
- 简单的方法
  - 渲染出镜子周围的部分
  - 渲染镜子里面的部分

镜子中的物体



- 如果镜子上的点P经过原点，且法向量为（0,1,0），则被反射的物体可以通过以下变换得到

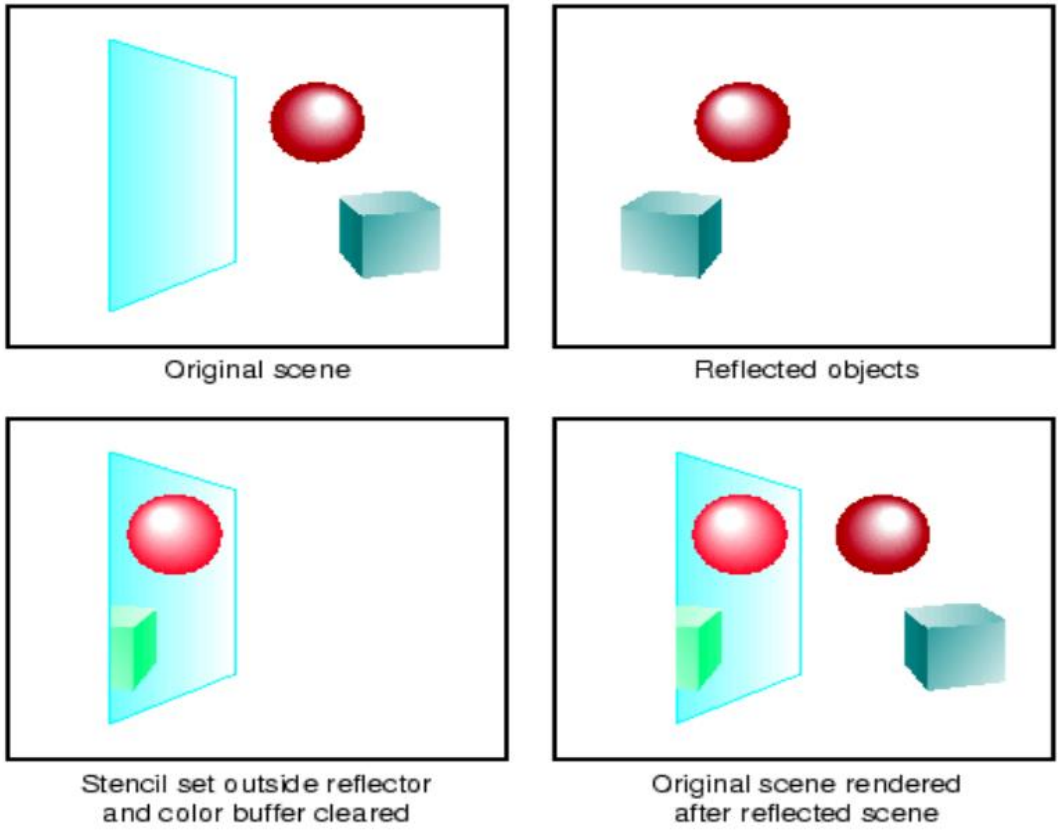
$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



- 如果P是任意点怎么计算？

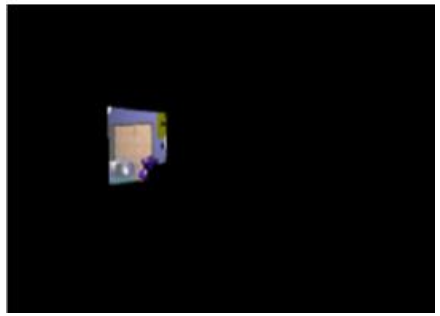
绘制包含镜子的场景

– 使用stencil buffer 来限制镜中世界的绘制范围



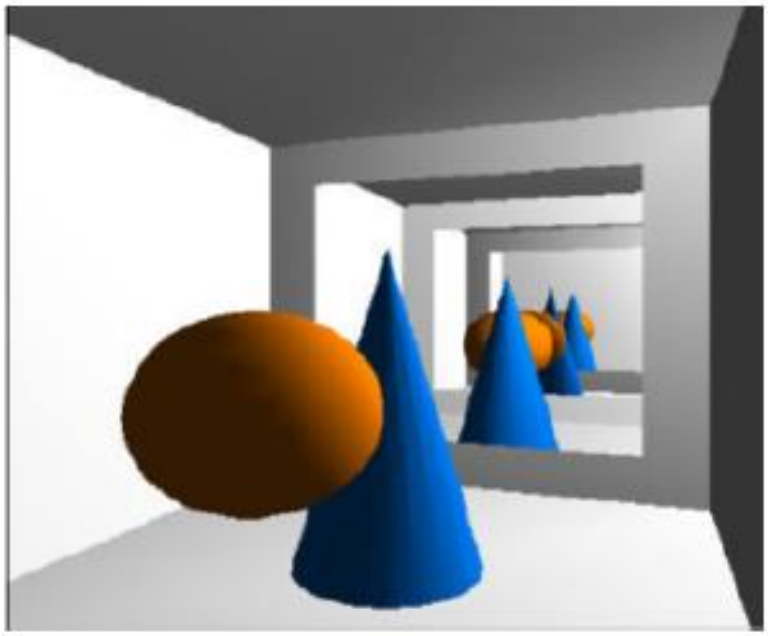
实现步骤

- 剔除镜子，绘制场景的其他部分
- 针对每个镜子：
  - 初始化stencil buffer
  - 绘制镜子，对于镜子内部，stencil被设置为1
  - 初始化depth buffer
  - 绘制镜内世界的物体（reflected objects）
  - 使用stencil buffer，只绘制镜中可见的部分
- 合并境外世界和镜中世界



多个镜子

- 先渲染镜子外的
- 再逐个渲染镜子
- 递归计算

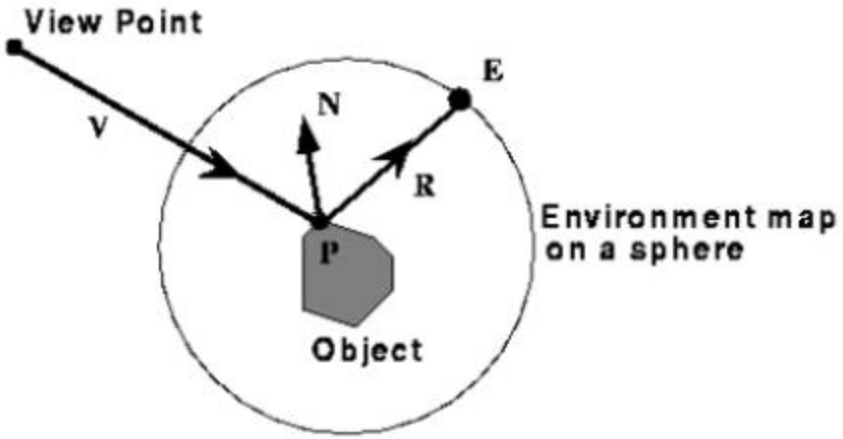


内容提要

- 阴影（shadow）
- 包含镜子的场景（mirroring）
- 环境映射（environment mapping）

环境映射(Environment mapping)

- 一种简单却强大的生成反射效果的方法
- 通过反射向量来模拟反射效果
- 最初提出[Lim Blinn 76]

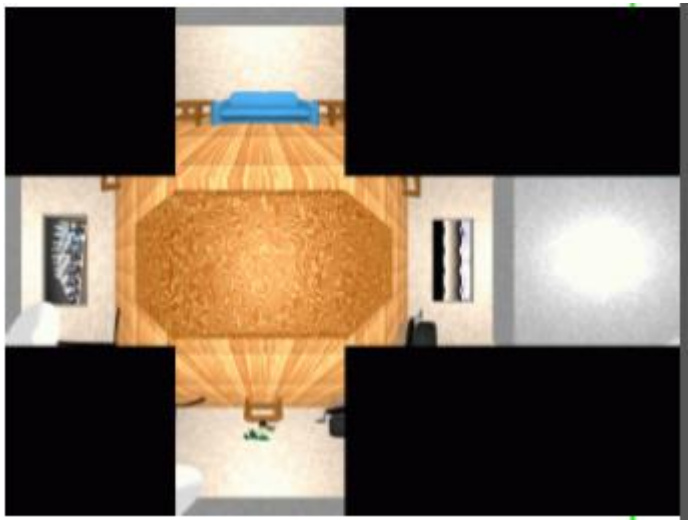


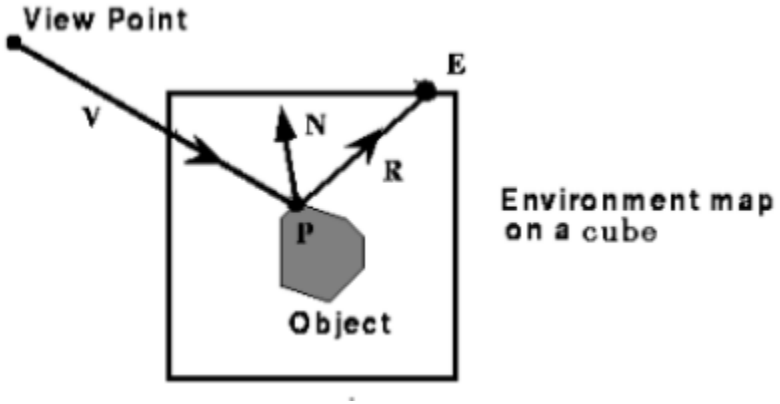
例子

Terminator II

立方体映射（Cubic mapping）

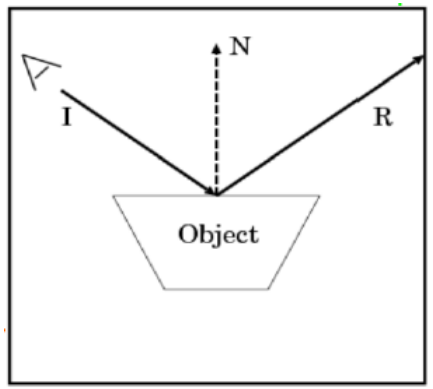
- 最常用的方法
- 用一个包围着物体的立方体的面来表示map





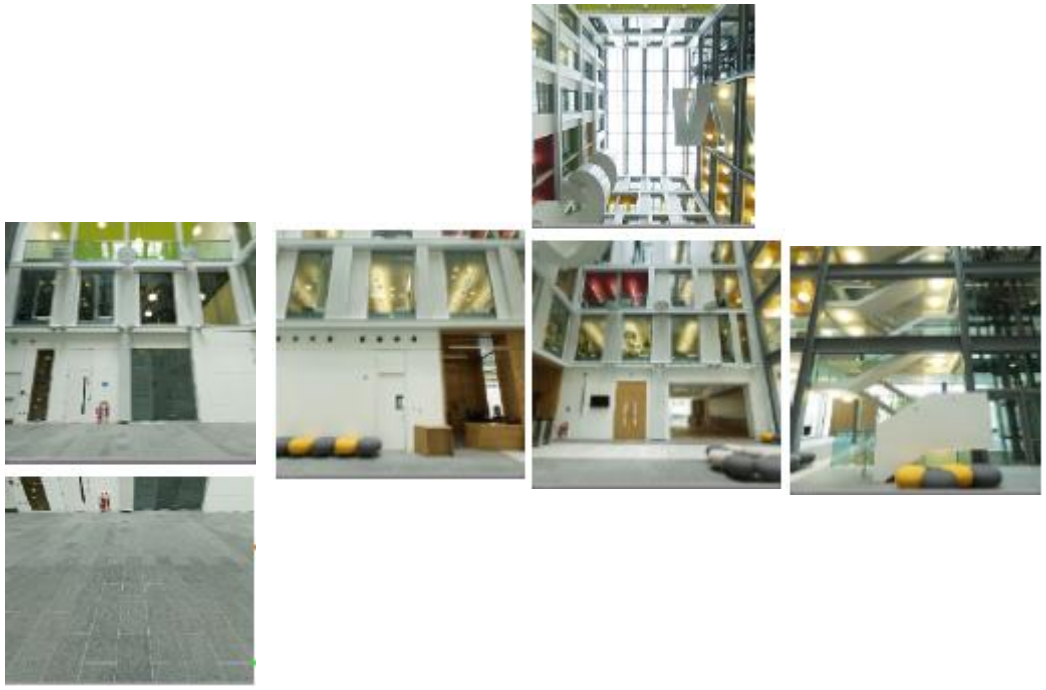
实现过程

- 针对每一个像素
  1. 通过摄像机向量（I）和物体表面的法向量（N）来计算反射向量（reflection vector **R**）
  2. 根据**R**来确定环境贴图上的点和对应于物体上的点
  3. 使用环境贴图上像素点的颜色给物体上的像素上色



生成 Cubic map

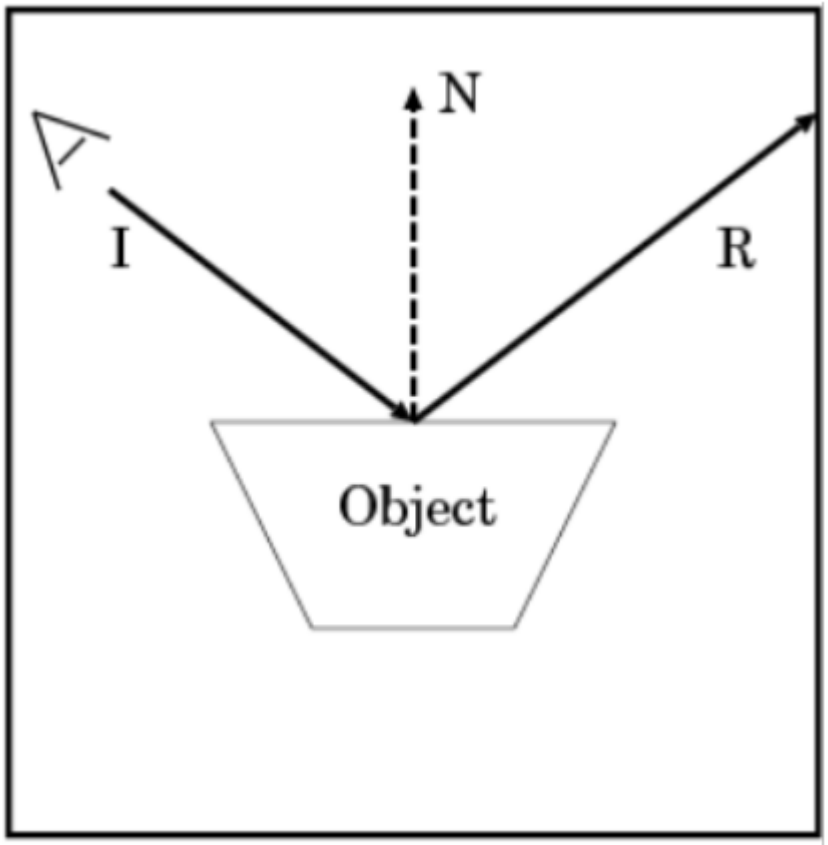
- 在物体的位置给周围环境拍六个照片



计算反射向量

- 法向量N，入射向量I，反射向量R
- 归一化

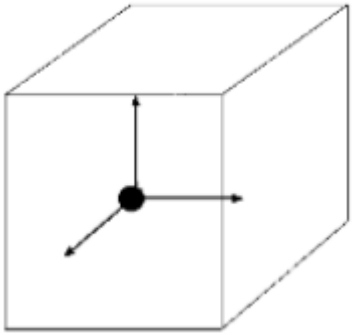
$$R = I - 2N(N.I)$$



- 纹理坐标由R决定
- 假设R的起始点（即反射点）在cubic map的中心点处

检索Cubic maps

- 假设已有R且Cubic map的边与坐标系对齐
- 如何决定使用cube的哪个面？
  - 反射向量R幅值最大的坐标
  - R = （0.3,0.2,0.8）则取z+方向的面

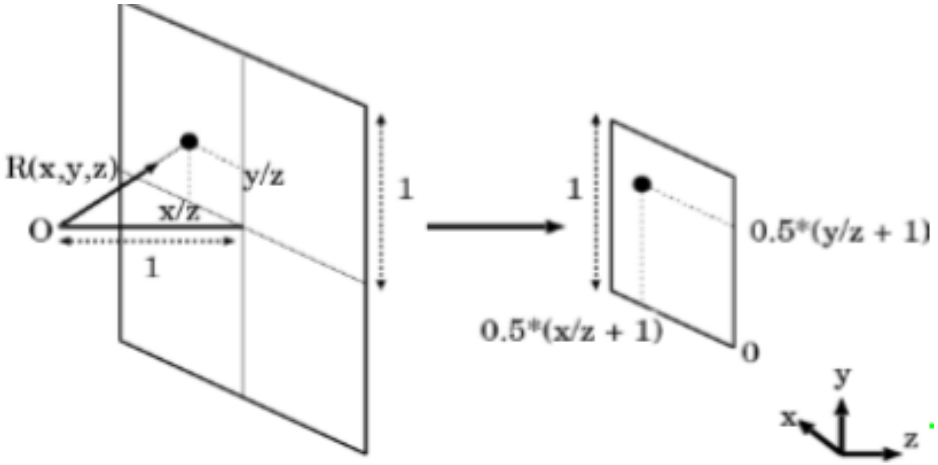


检索Cubic maps

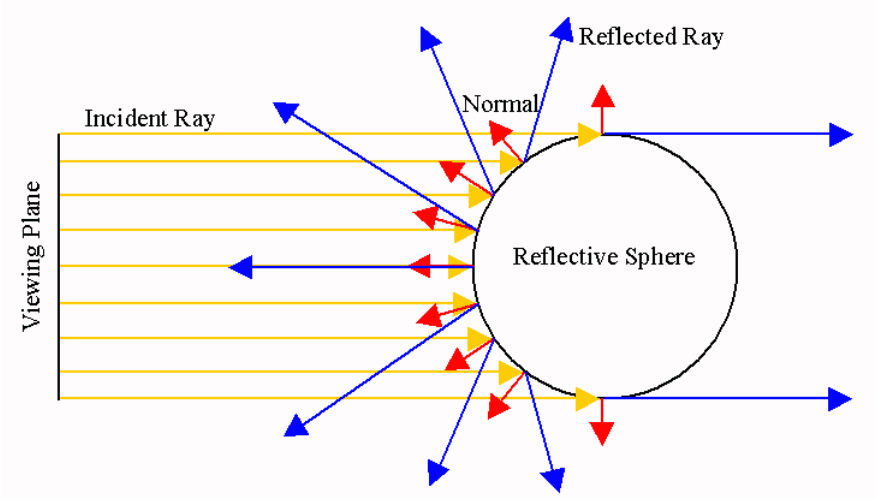
- 如何确定具体的纹理坐标？
  - 使用R向量中最大的值来归一化
  - 每个维度的值 [-1, 1]
  - 再映射到0-1之间

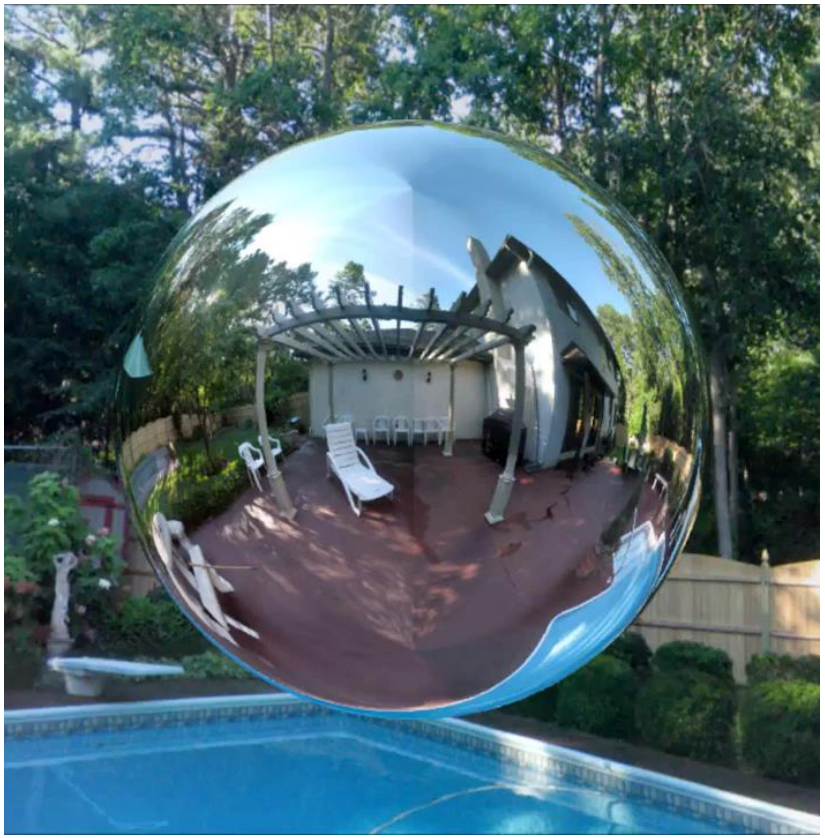
$$(0.3,0.2,0.8) \xrightarrow{\text{yields}}$$

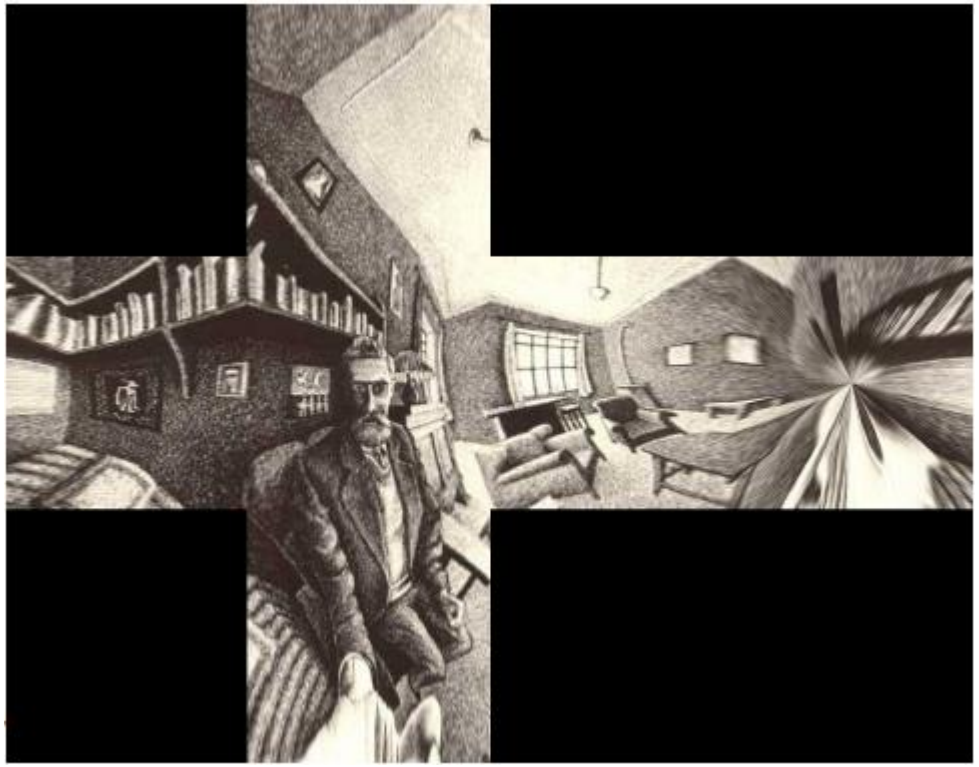
$$\left(\left(\frac{0.3}{0.8}+1\right)\times 0.5, \left(\frac{0.2}{0.8}+1\right)\times 0.5\right)$$
$$= (0.6875,0.625)$$



球面映射 （sphere map）



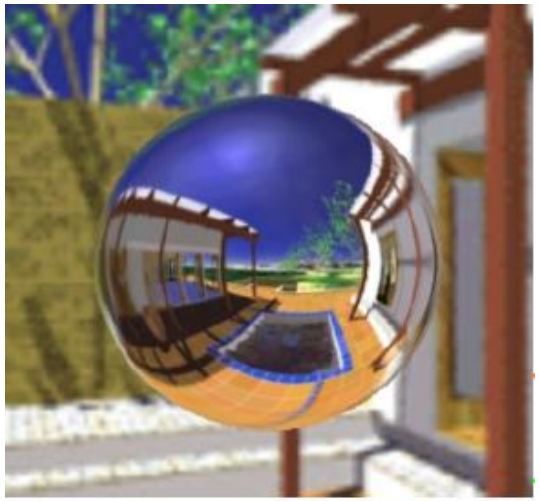




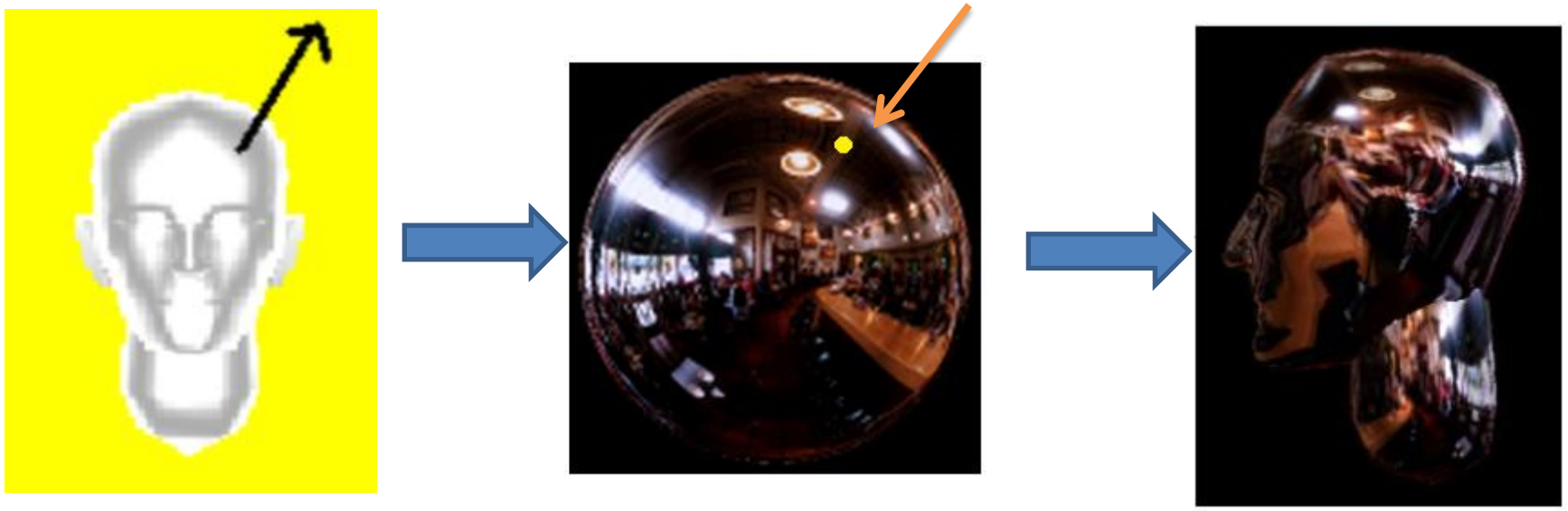
https://en.wikipedia.org/wiki/Hand_with_Reflecting_Sphere

球面映射 （sphere map）

- 怎样得到sphere map？
  - 给反射球照张相
  - 把cubic environment map 映射到一个球面上
  - Ray tracing



球面映射的过程



- 计算model上的反射向量
- 找到map上对应的（u,v）坐标
- 映射回模型上

检索sphere maps

- The normal vector is the sum of the reflection vector and the eye vector

$$N = (R_x, R_y, R_z + 1)$$

- Normalization of the vector gives

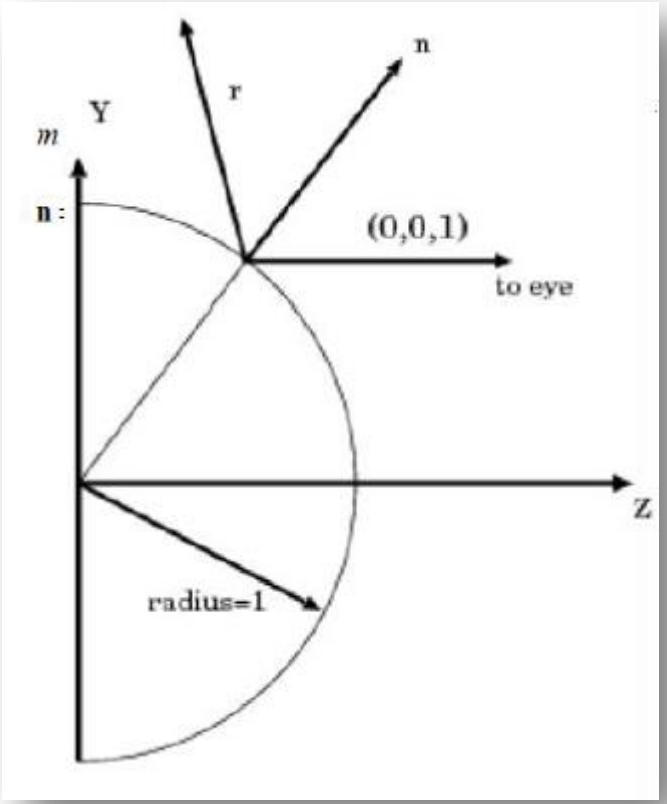
$$m = \sqrt{R_x^2 + R_y^2 + (R_z + 1)^2}$$

$$n = \left(\frac{R_x}{m}, \frac{R_y}{m}, \frac{R_z + 1}{m}\right)$$

- If the normal is on a sphere of radius 1, its x, y coordinates are also location on the sphere map. Finally converting them to make their range [0,1]

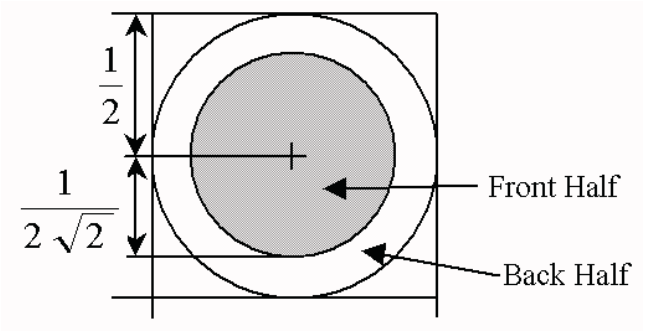
$$u = \frac{R_x}{2m} + \frac{1}{2}, v = \frac{R_y}{2m} + \frac{1}{2}$$

$$m = \sqrt{R_x^2 + R_y^2 + (R_z + 1)^2}$$

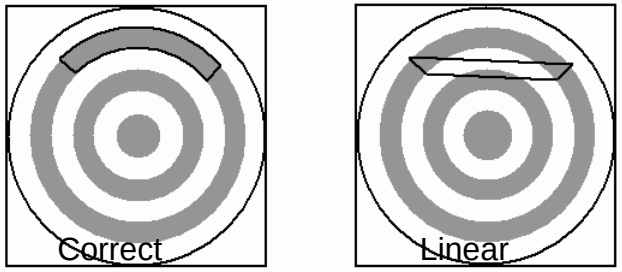


非线性映射

- 问题：
- 采样高度不均匀
- 高度非线性



- 对于(u, v)的线性插值导致错误的纹理
  - 物体表面需要细分



比较和小结

- 立方体映射和球面映射比较
  - 立方体映射有什么好处？
  - 球面映射有什么问题？

内容回顾

- 光照 （illumination）
  - 光线和材质
  - 光源
  - Phong反射模型
- 明暗绘制 （shading）
  - 均匀着色
  - 光滑着色
  - Phong着色
- 阴影（shadow）
  - 投射阴影
  - 阴影纹理
  - 阴影体
  - 阴影贴图
  - 柔性阴影
- 镜面（mirroring）
  - 镜面变换
  - 渲染步骤
  - 多个镜子
- 环境映射（Environment mapping）
  - 立方体映射
  - 球面映射

参考及引用文献

- http://vcc.szu.edu.cn/index.html
- https://cg.cs.tsinghua.edu.cn/course/
- http://homepages.inf.ed.ac.uk/tkomura/Teaching.htm
- A Survey of Real-time Soft Shadows Algorithms