



中山大學 软件工程学院
SUN YAT-SEN UNIVERSITY SCHOOL OF SOFTWARE ENGINEERING

Lecture 15: 着色和阴影

SSE315: 计算机图形学
Computer Graphics

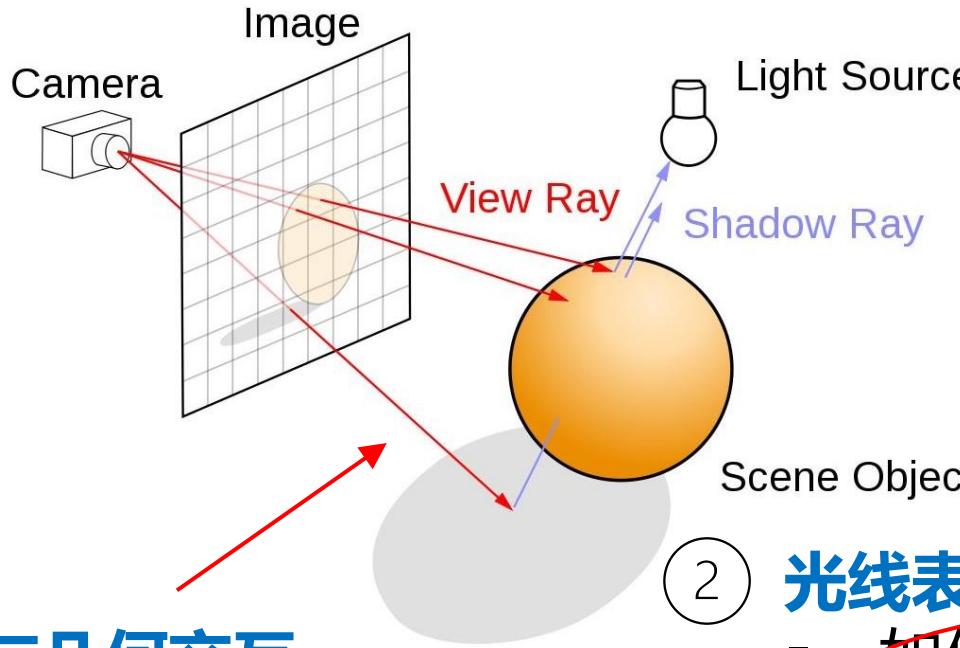
陈壮彬

软件工程学院

chenzhb36@mail.sysu.edu.cn

光线追踪 Ray tracing

分成 **光线与几何交互** 和 **光线表示与可视化** 两部分



① 光线与几何交互

- 查找光线照射在几何中的位置
- 如何加速上述过程

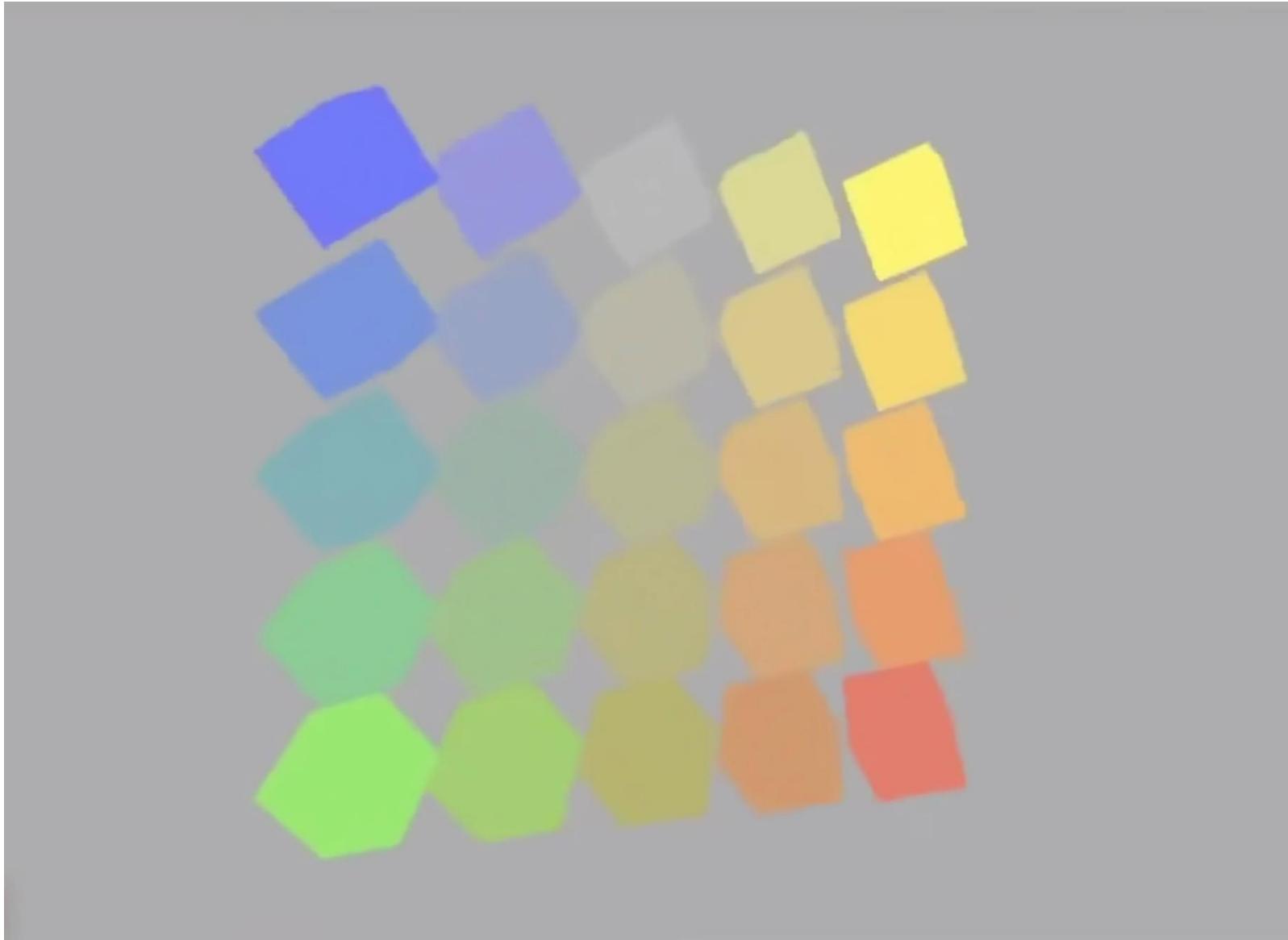
看到哪里

② 光线表示与可视化

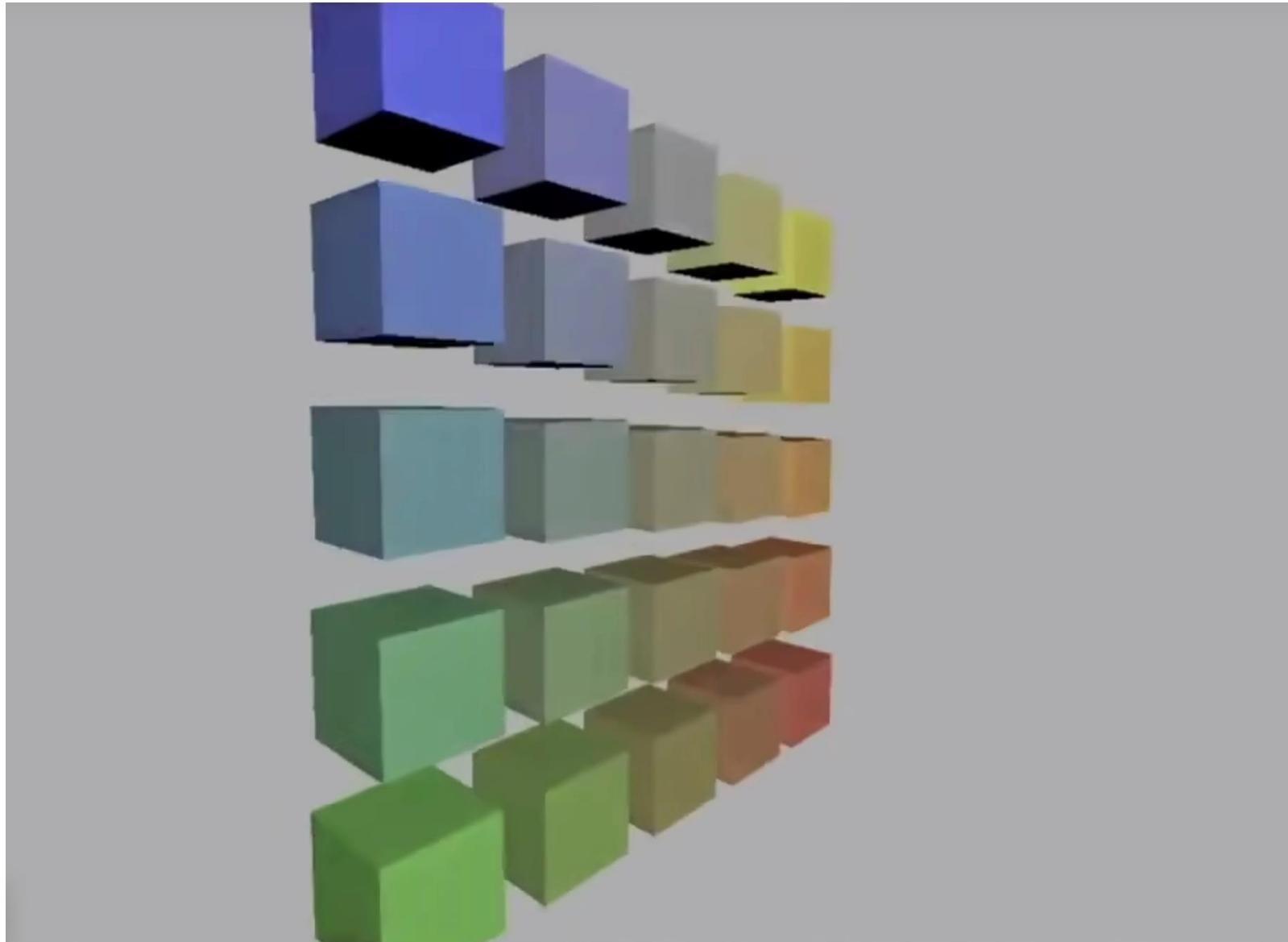
- 如何建模光线
- **光线与物体的相互作用，包括反射、折射和散射等**
- 全局光照

看到什么

旋转立方体 (逆时针还是顺时针?)



我们期望看到的效果



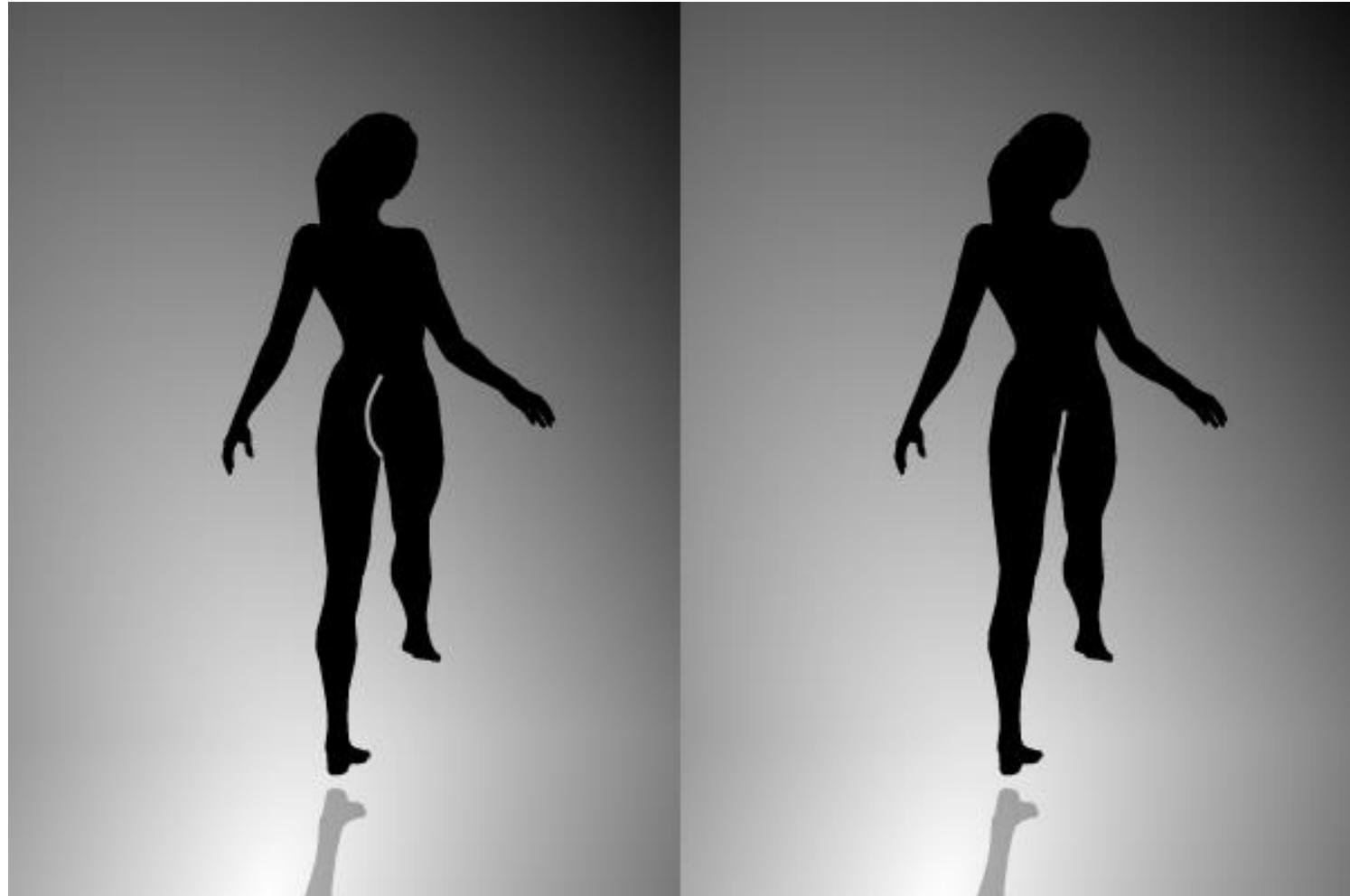
旋转舞者 Spinning Dancer

□ 缺乏深度信息



旋转舞者 Spinning Dancer

□添加深度信息



Today's topics

□ 着色 (Shading) 介绍

□ 着色模型

- Blinn-Phong 着色 (Blinn-Phong shading)
- 平面着色 (Flat shading)
- Gouraud 着色 (Gouraud shading)

□ 阴影贴图 (Shadow mapping)

着色 Shading

口韦氏词典

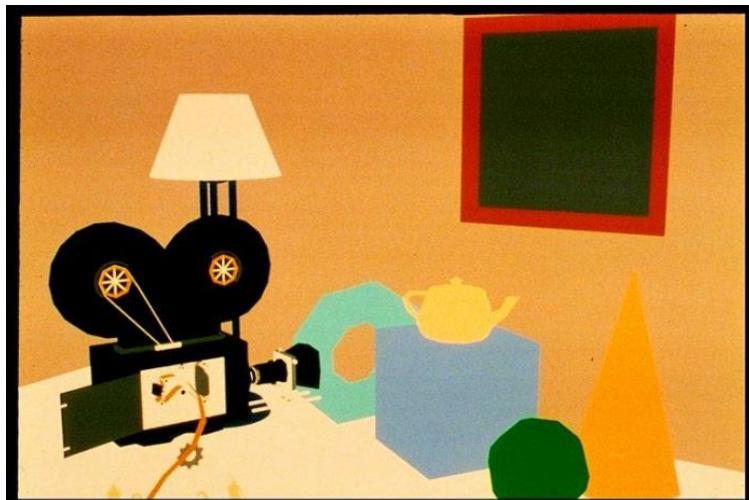
shad·ing, ['ʃeɪdɪŋ], noun

The **darkening** or **coloring** of an illustration or diagram with parallel lines or a block of color.

着色 Shading

口计算光线对物体表面的影响，包括物体的材料属性、表面方向以及光源的位置和强度

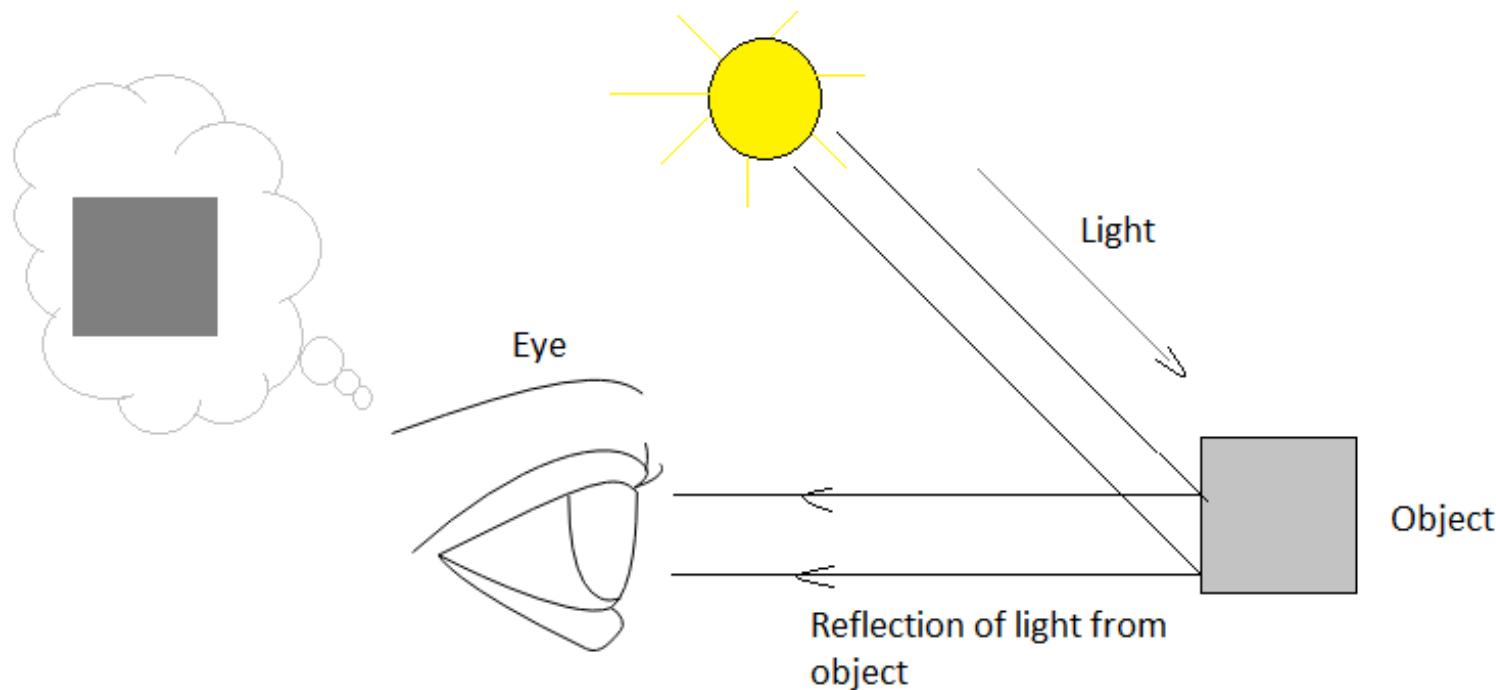
口主要考虑光线如何影响物体表面的**颜色**和**明暗**，使物体表面看起来**更加真实**和**有立体感**



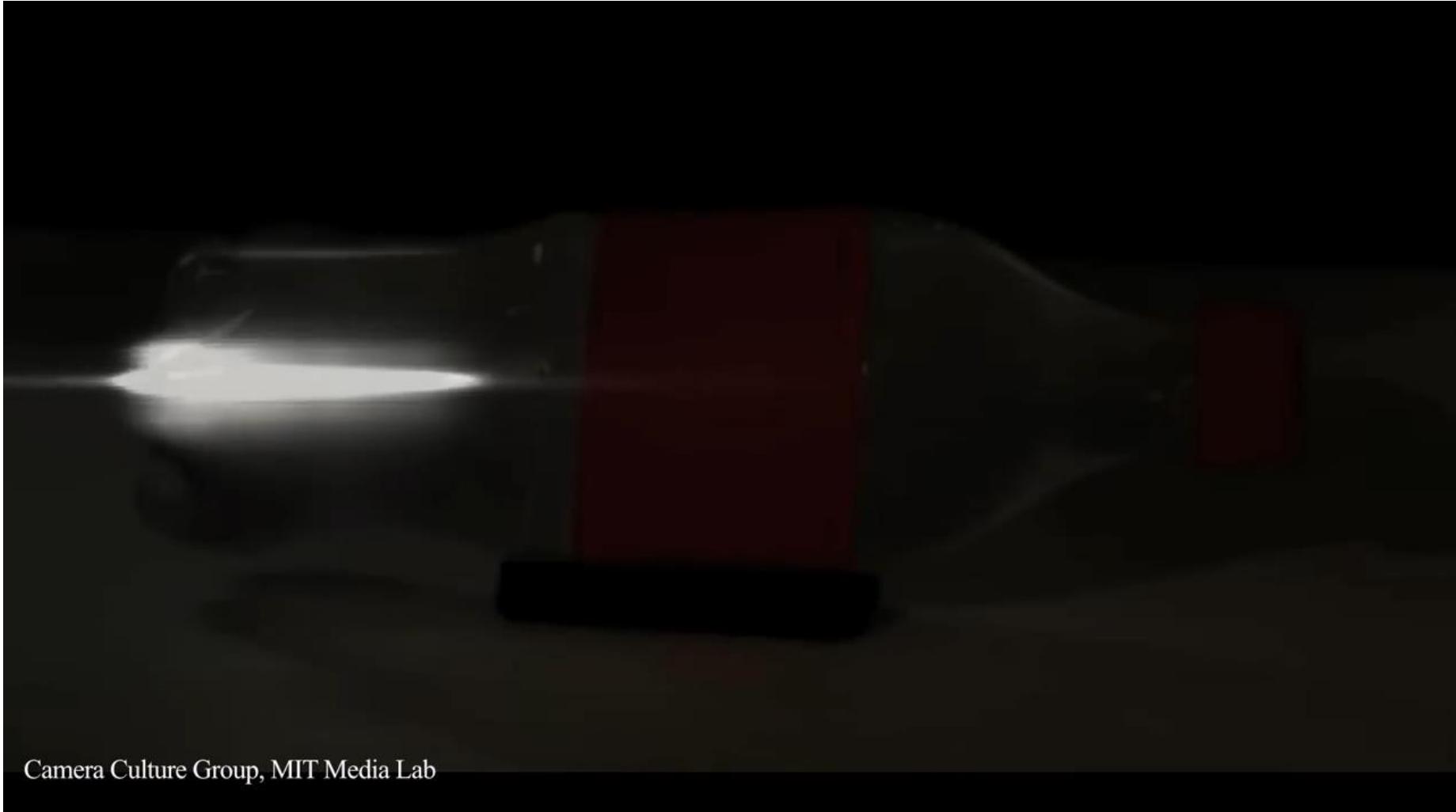
我们为什么能看到物体？ – 光子

口物体能被看见的原因是因为光源发出的光线照射到物体上，然后反射到我们的眼睛中

口从物理学上讲，是因为光子 (Photons, 光的基本单位) 从物体表面反射到我们的眼睛



光线穿过物体

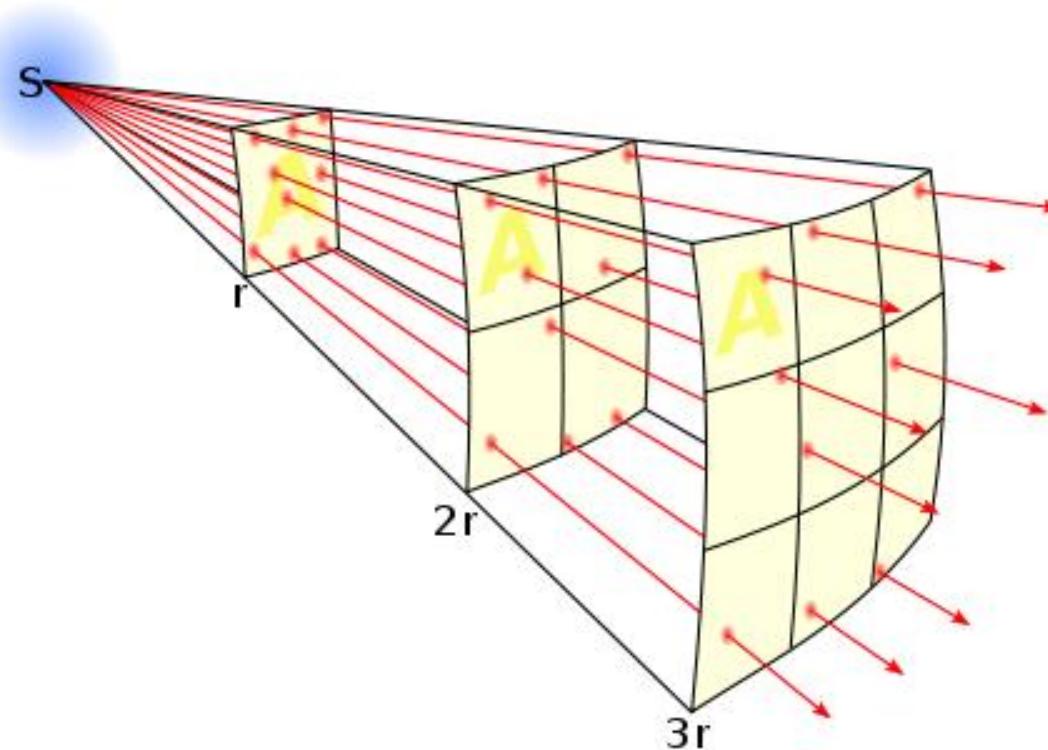


Camera Culture Group, MIT Media Lab

光强

口光强 (Light's intensity) 是指光源在单位时间内，通过单位面积，沿着垂直于该面积的方向发出的光子数目

口光强越大则传递到能量越多 (光线更亮或热量更多)

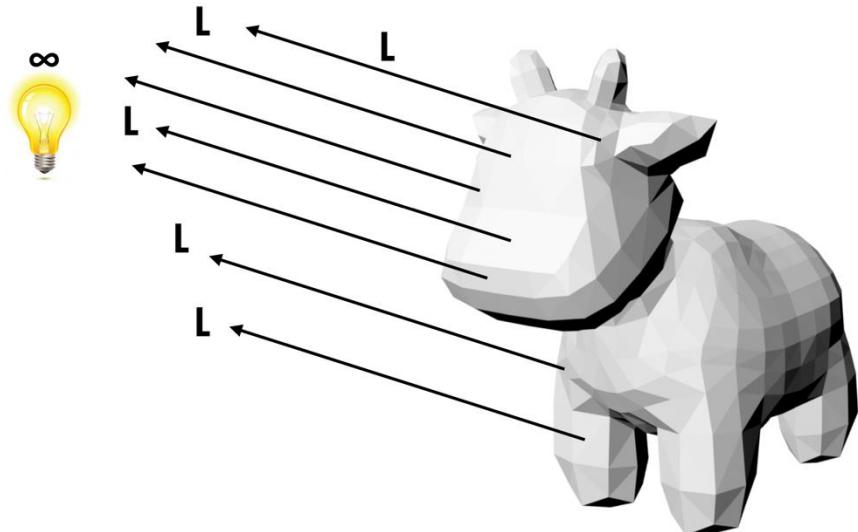
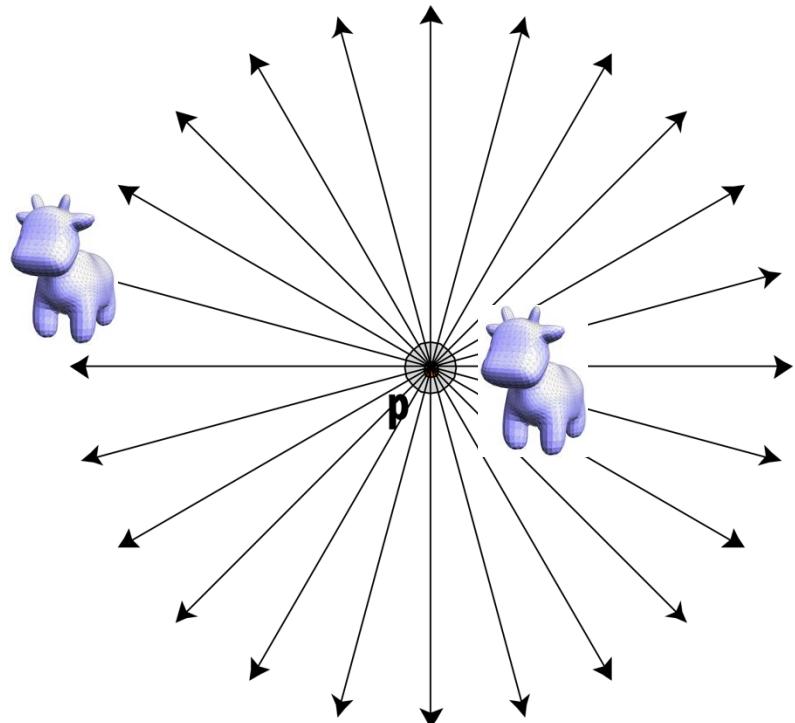


光源的简单抽象模型

口光强固定 (即单位时间内向周围发出的光子数量相同)

- 离光源越远，物体越暗 (因为接收到的光子变少了)

口光源无穷远，光线是离散且相互平行的

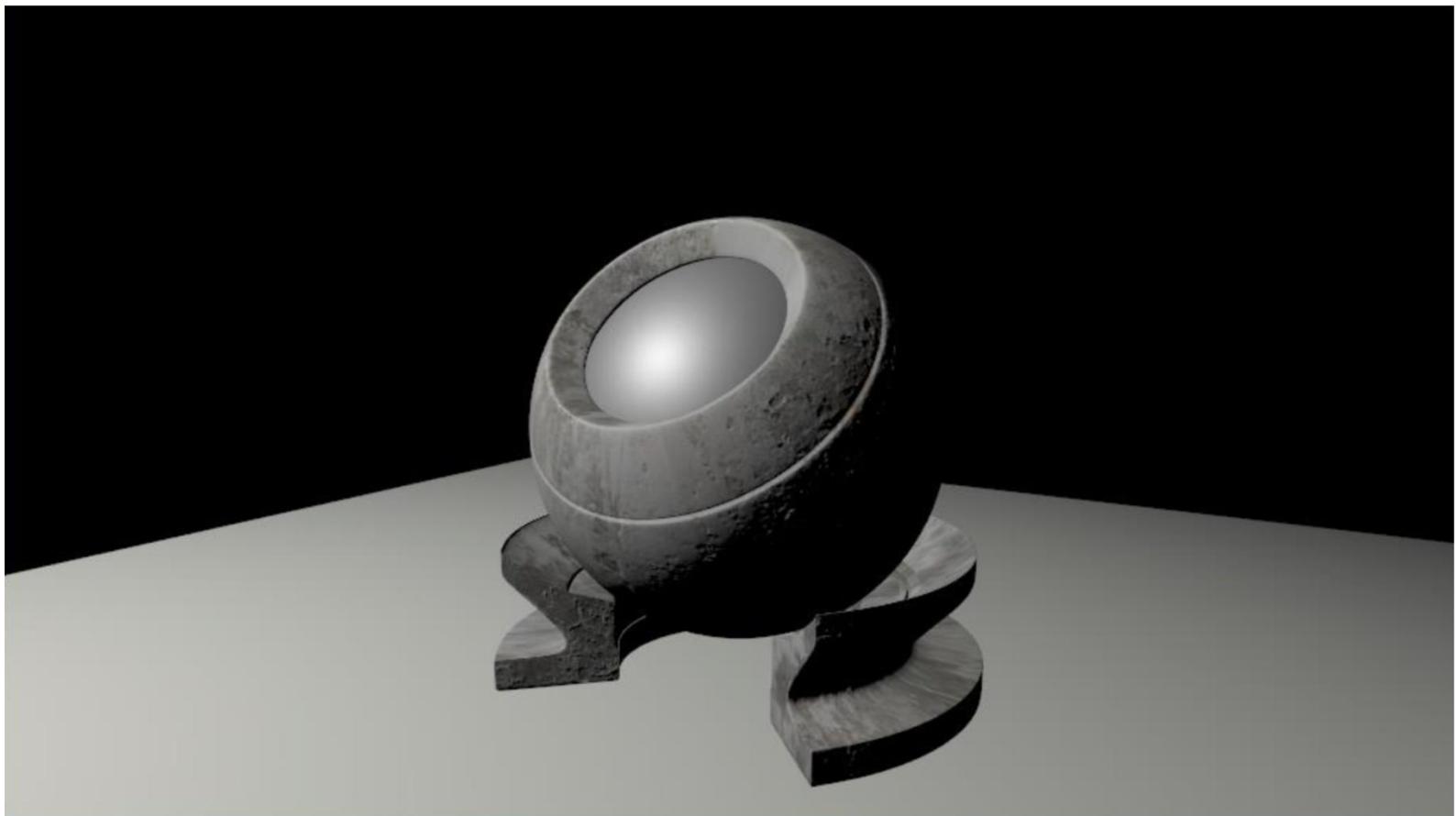


我们不久后将更细致地讨论光 (即辐射度量学, Radiometry) 14

着色 vs. 添加阴影

口着色 (Shading)

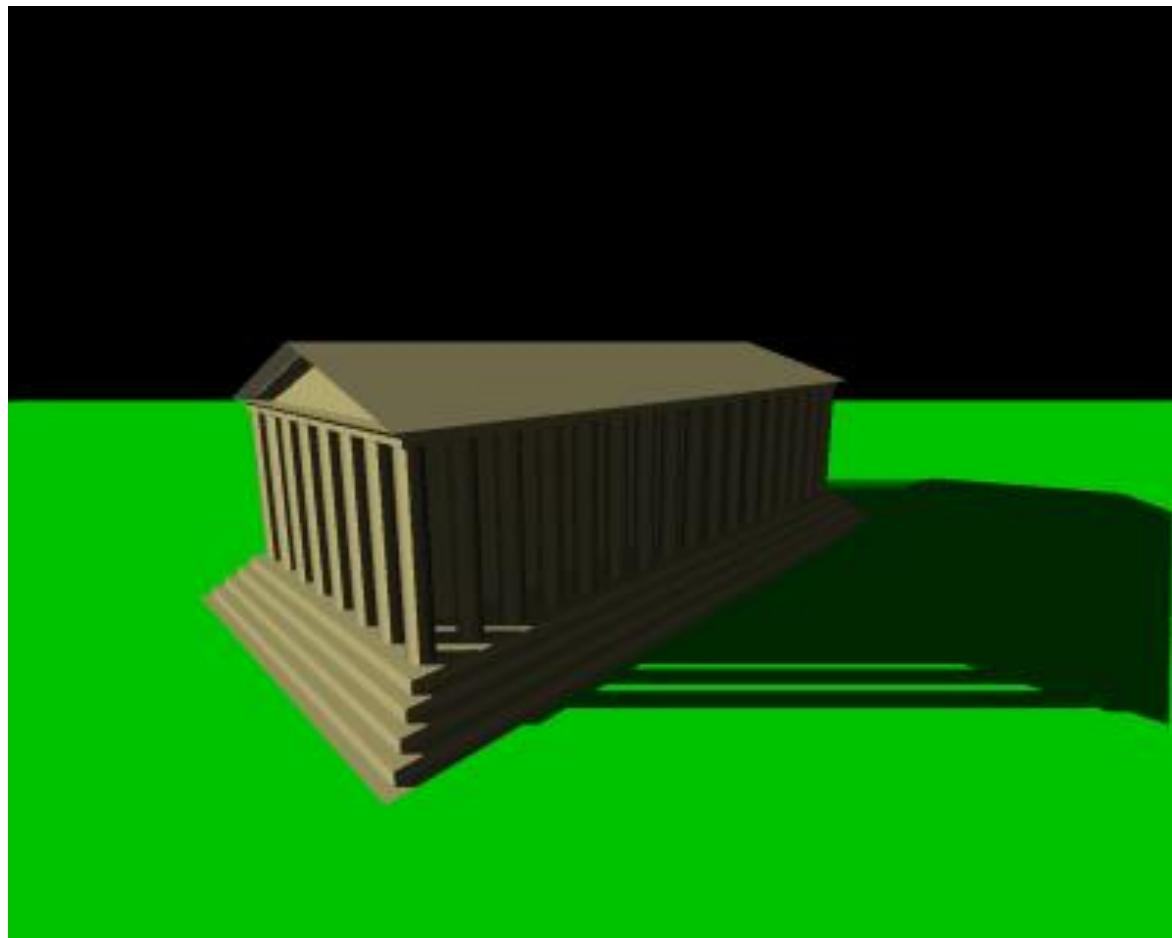
- 研究光线的局部行为，模拟光线对物体表面的影响 (颜色和明暗)，不考虑其他物体



着色 vs. 添加阴影

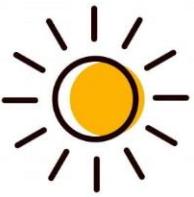
□添加阴影 (Adding shadow)

- 研究光线的全局行为，判断物体是否被遮挡 (位于阴影区域)



Blinn-Phong 反射模型

Blinn-Phong Reflectance Model



光线感知

高光 (Specular highlights)

大部分光线沿着特定方向散射出去，形成高亮的点

漫反射光 (Diffuse reflection)

光线以各个方向均匀地散射出去

环境光 (Ambient lighting)

光线从所有方向均匀地照射在物体上，可简化为固定光照

BP 反射模型同时考虑了这三种类型的光以生成真实的效果

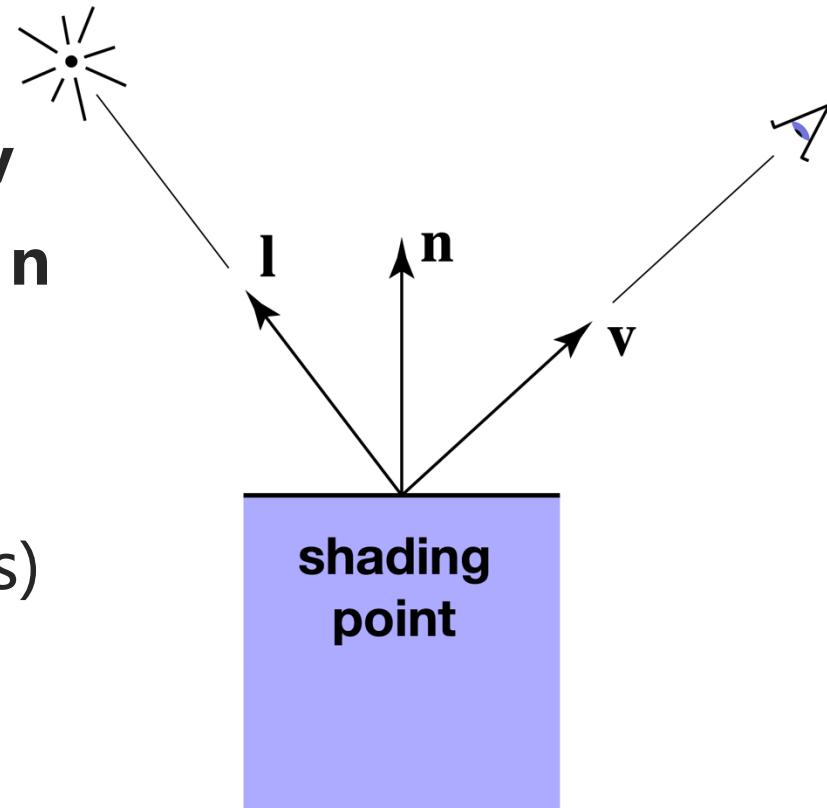


光线反射模型

计算在特定着色点 (shading point) 反射到摄影机的光线

输入

- 观察方向 (viewer direction), v
- 表面法向量 (surface normal), n
- 光照方向 (light direction), l
 - 针对每一个光源
- 表面参数 (surface parameters)
 - 颜色 (color)
 - 光泽度 (shininess)



仅表示方向，为简化计算，所有向量均为单位向量

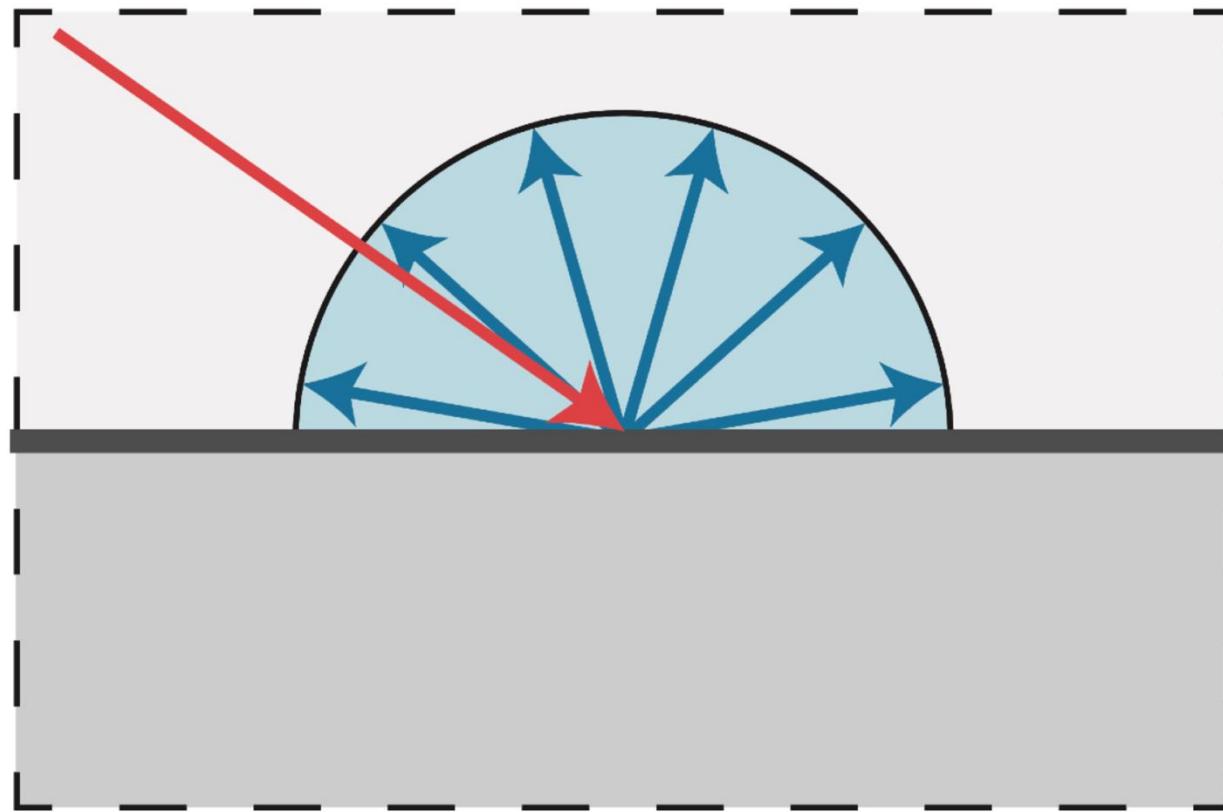
漫反射光

Diffuse Reflection

漫反射

口光向各个方向均匀散射

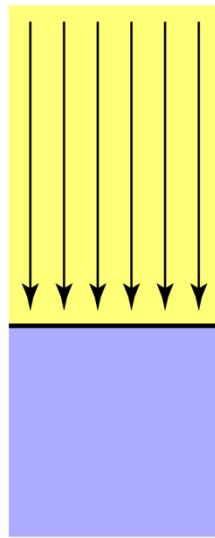
- 所有观察方向看到的表面反射光都相同



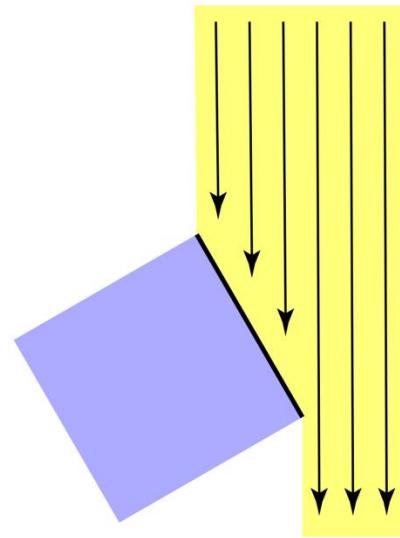
漫反射

□不同方向具体接收到多少光(能量)?

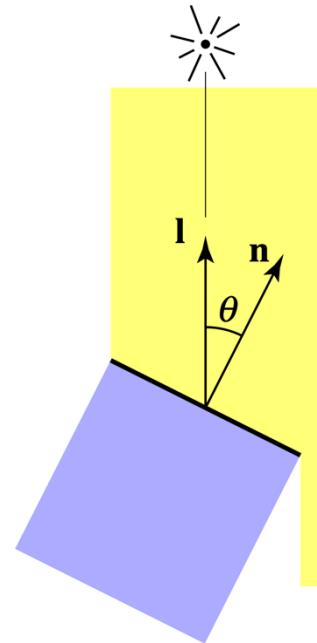
- 朗伯余弦定律 (Lambert's cosine law)



垂直光照方向
接收全部的光

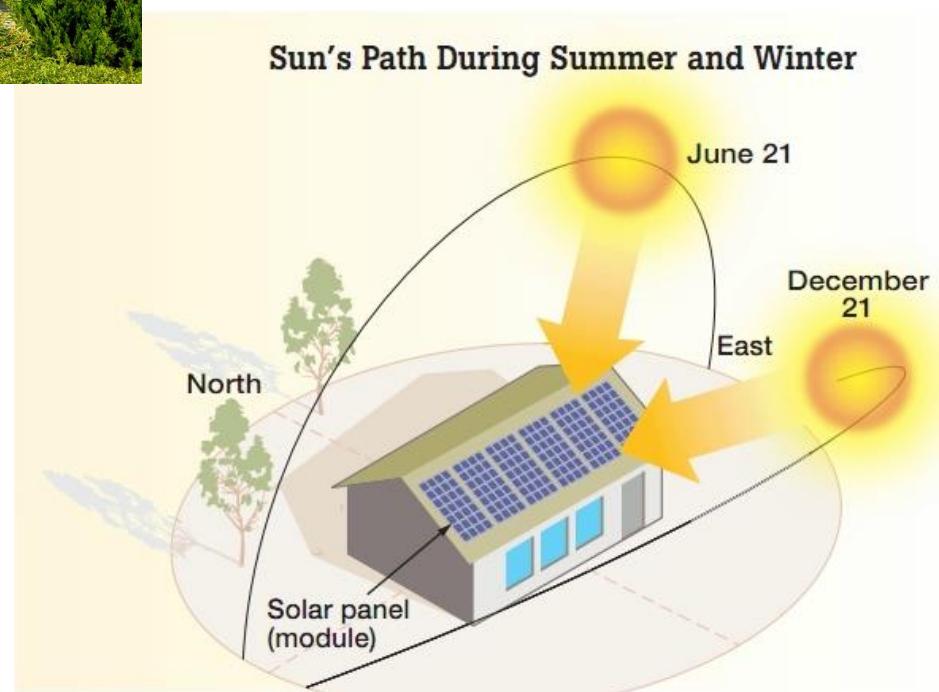


旋转 60° 接收
一半的光



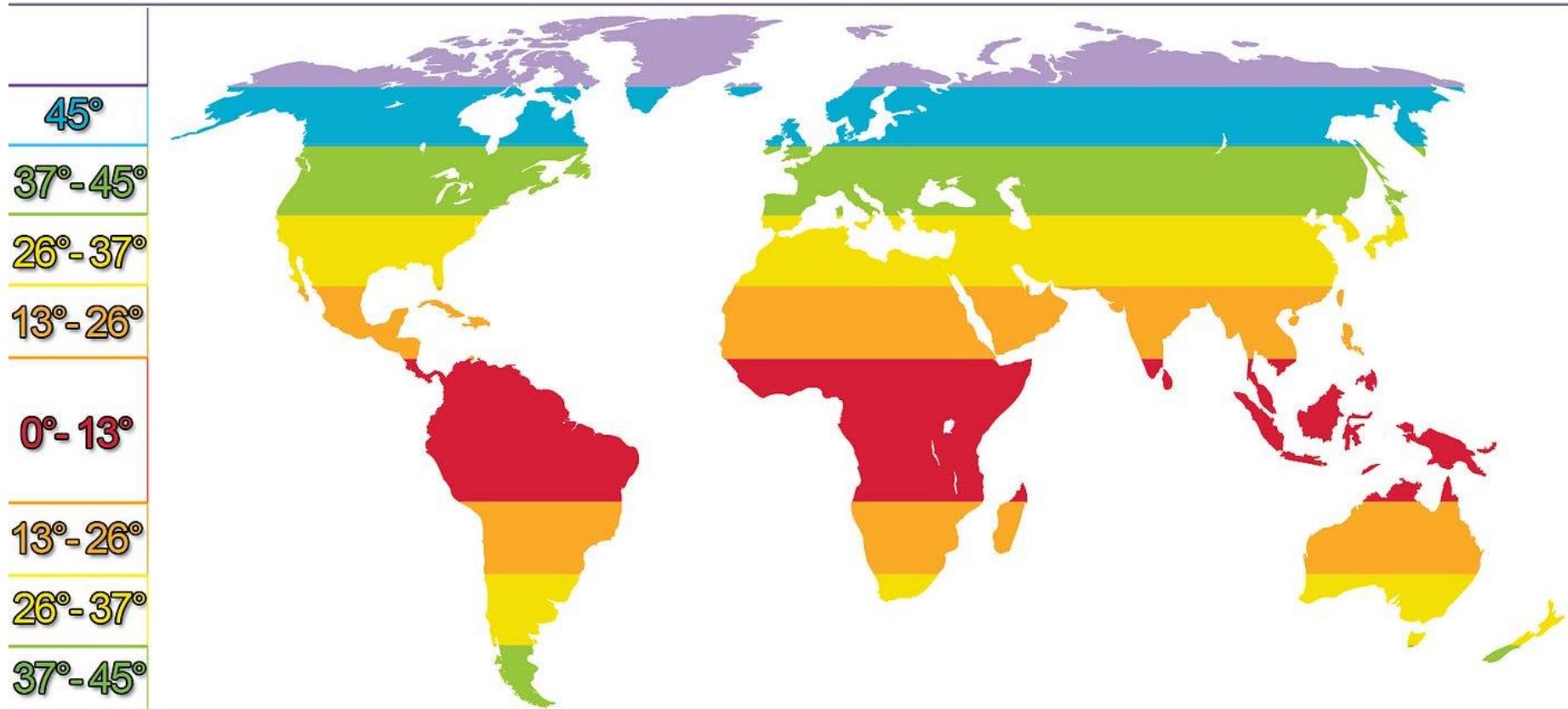
旋转 θ , 接收的
光与 $\cos(\theta) =$
 $l \cdot n$ 成比例

太阳能板 Solar panel

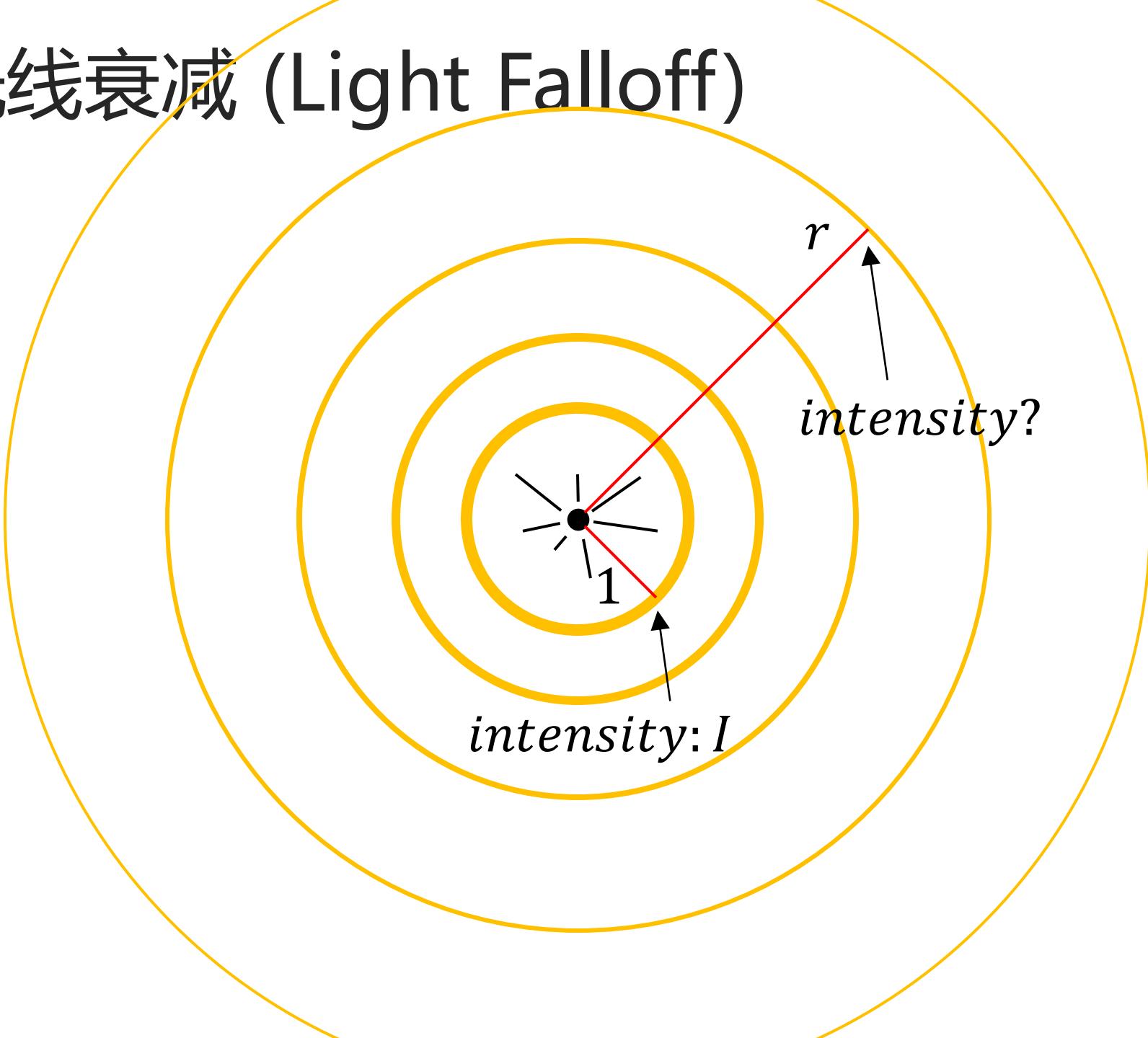


全球太阳能板的最佳角度

Optimal angle for fixed solar panels depending on installation position

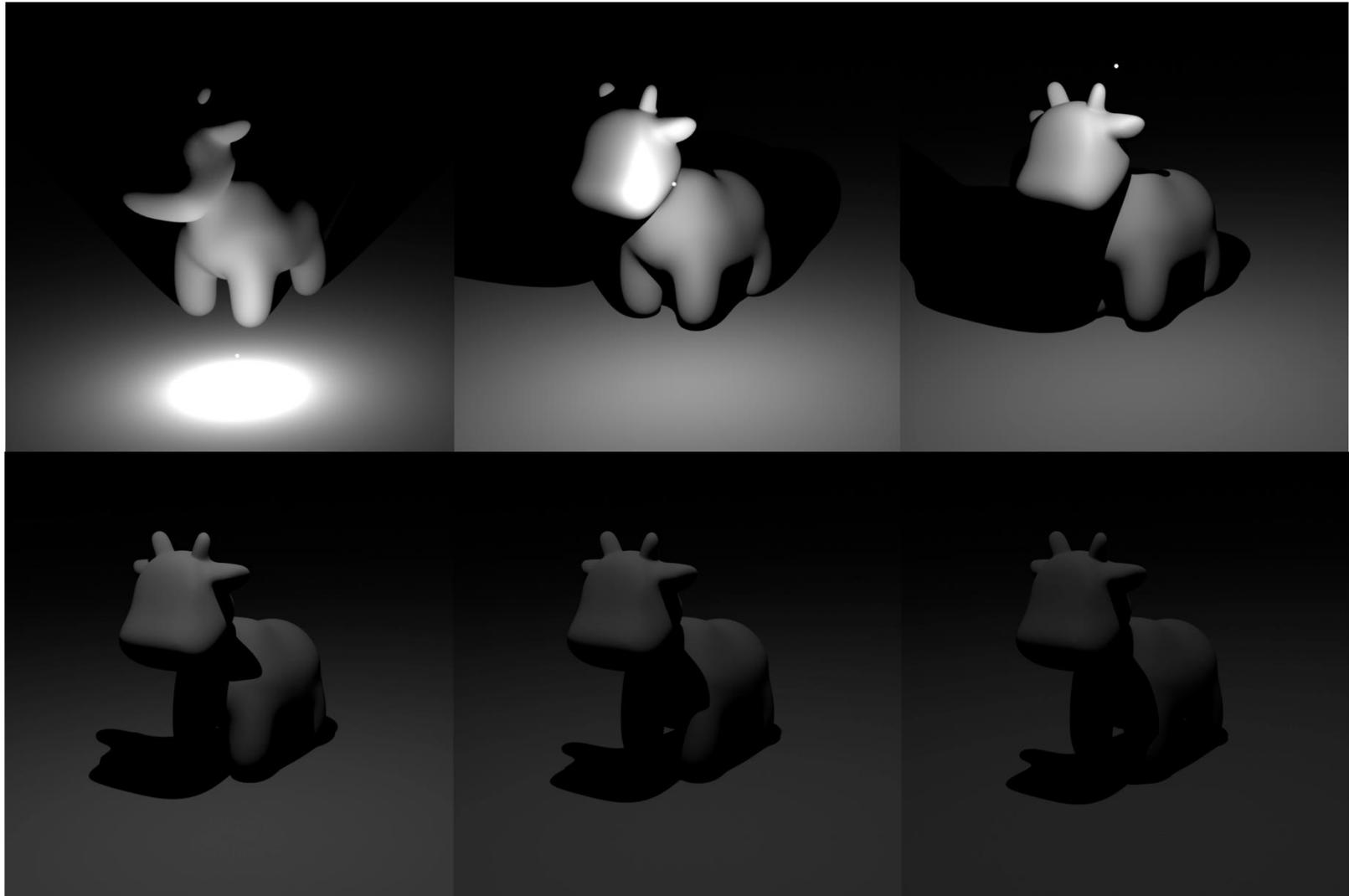


光线衰减 (Light Falloff)



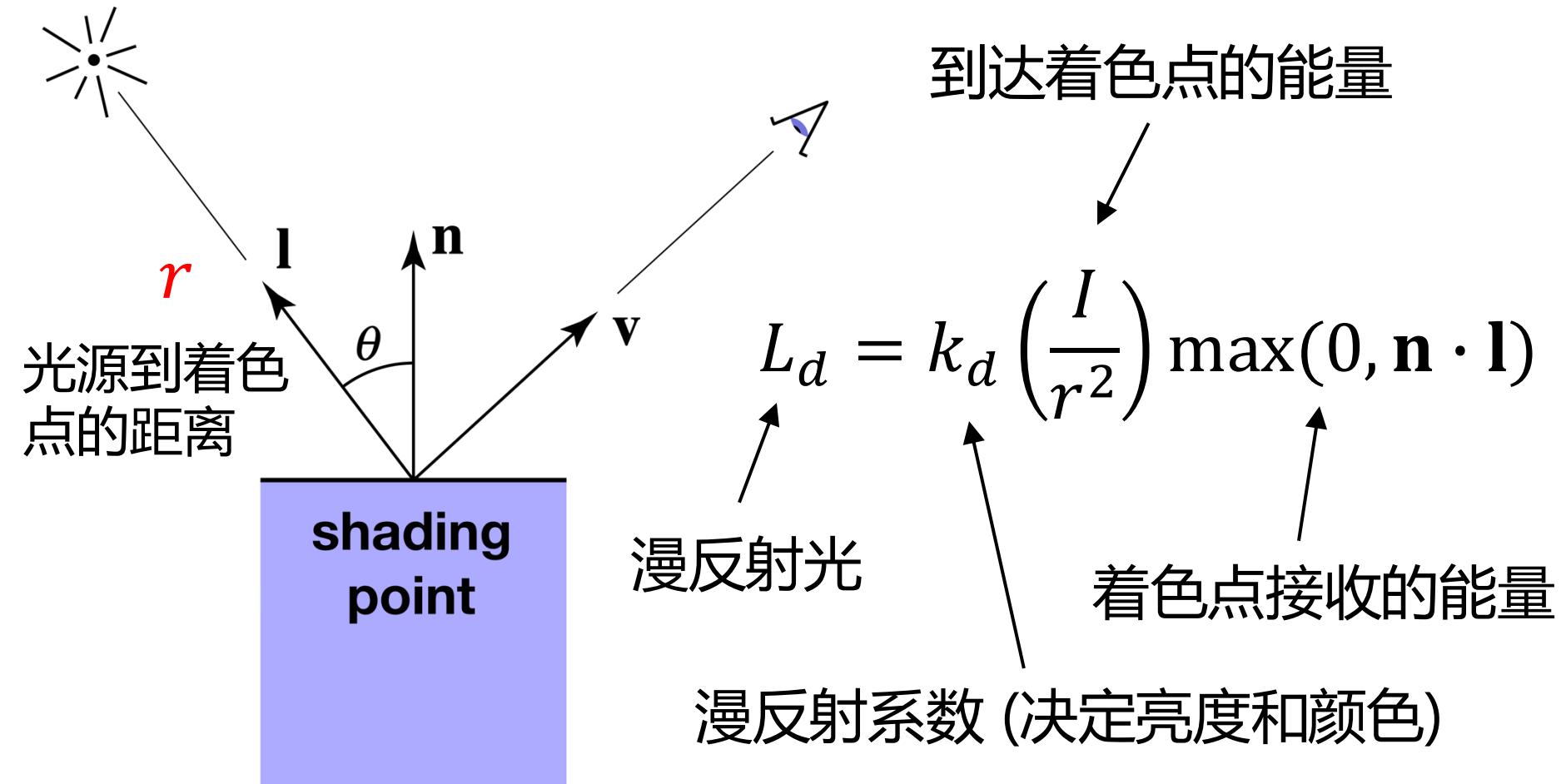
光线衰减可视化

□ 离光源越远，物体越暗



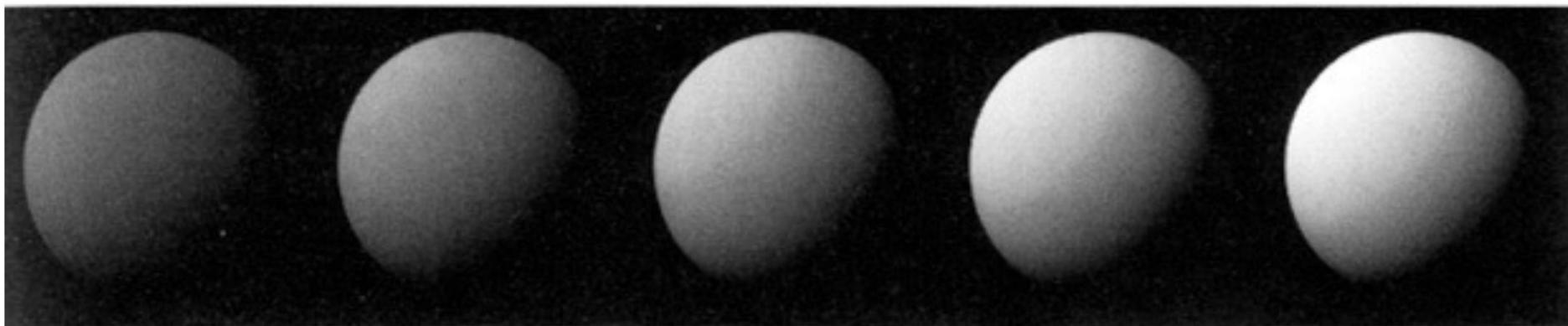
漫反射着色

口着色与观察方向无关



漫反射着色效果

- 口对 RGB 通道分别定义 k_d , 会产生不同的颜色
- 口从不同角度看到的结果是一样的



$$k_d \longrightarrow$$

高光
Specular highlights

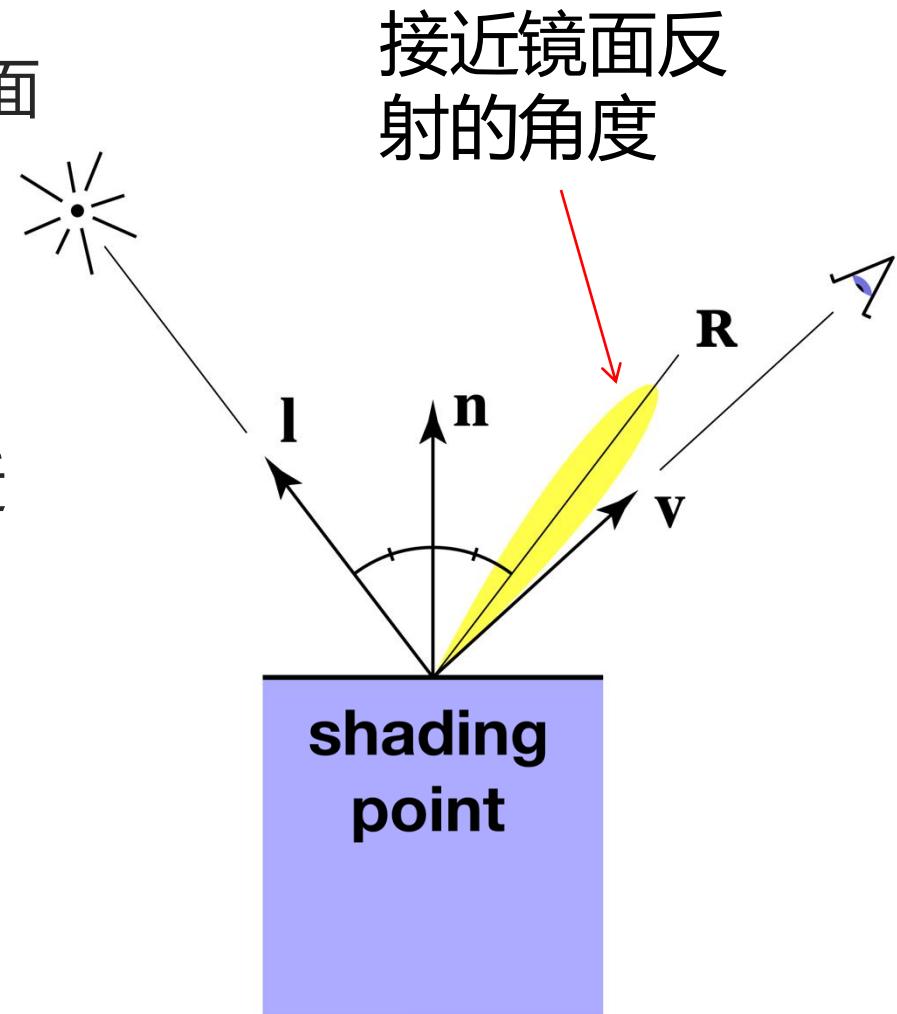
高光着色

口什么时候会出现高光?

- 物体表面比较光滑，比如镜面

口强度取决于观察方向

- 接近镜面反射方向的时候才能看见高光，即 v 和 R 接近

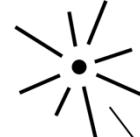


高光着色

□ Blinn-Phong 模型对高光的处理

- 观察角度与镜面反射角度接近 \Leftrightarrow 半程向量与法向量接近

□ 用点积衡量向量接近程度

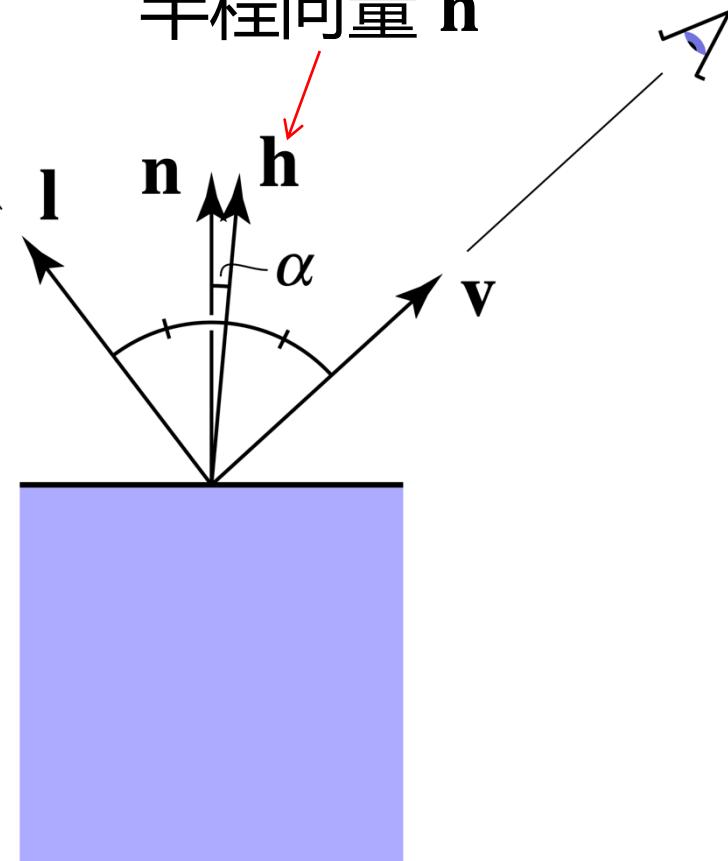


角平分线方向

$$\mathbf{h} = \text{bisector}(\mathbf{v}, \mathbf{l}) = \frac{\mathbf{v} + \mathbf{l}}{\|\mathbf{v} + \mathbf{l}\|}$$

归一化

半程向量 \mathbf{h}



镜面反射光 镜面反射光系数

$$L_s = k_s \left(\frac{I}{r^2} \right) \max(0, \cos \alpha)^p$$

$$= k_s \left(\frac{I}{r^2} \right) \max(0, \mathbf{n} \cdot \mathbf{h})^p$$

Blinn-Phong 高光着色模型 – 讨论

□ Blinn-Phong 没有考虑光线与平面的角度

- 经验模型，做了简化

$$L_s = k_s \left(\frac{I}{r^2} \right) \max(0, \mathbf{n} \cdot \mathbf{h})^p \cancel{\max(0, \mathbf{n} \cdot \mathbf{l})}$$

□ 为什么不直接计算观察角度与镜面反射角度的相似性？

- 也可以，即 Phong 模型，Blinn-Phong 是改进的模型
- 计算半程向量比计算镜面反射向量（给定入射向量和法向量）要容易

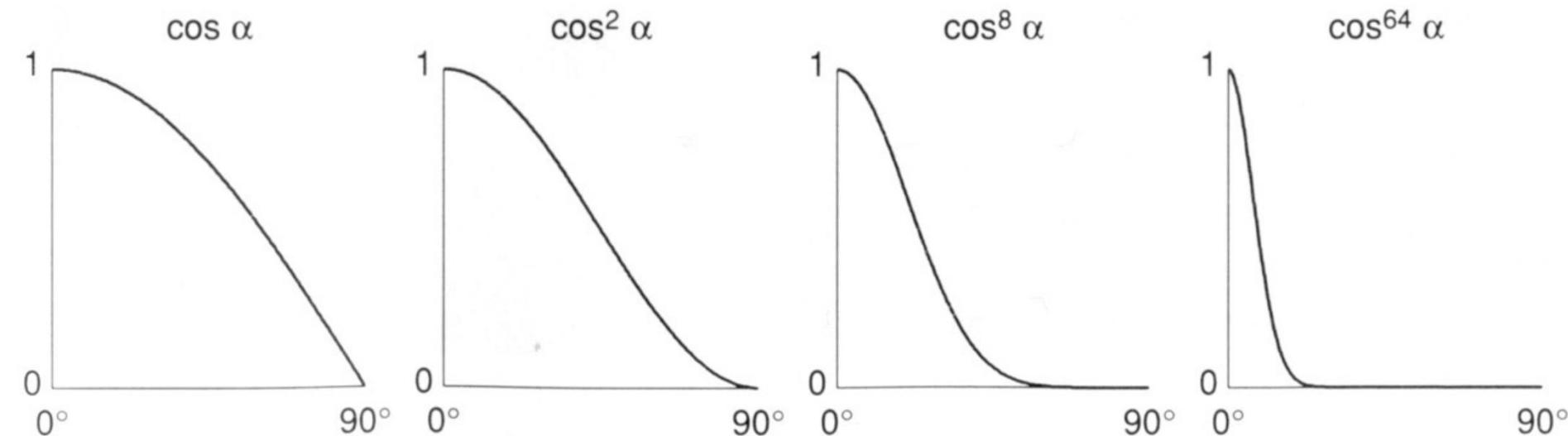
□ 不考虑从物体到观察者的光线衰减

- 简单来说，是因为人眼对这种光线衰减不敏感

□ 为什么有一个指数 p ？

余弦指数

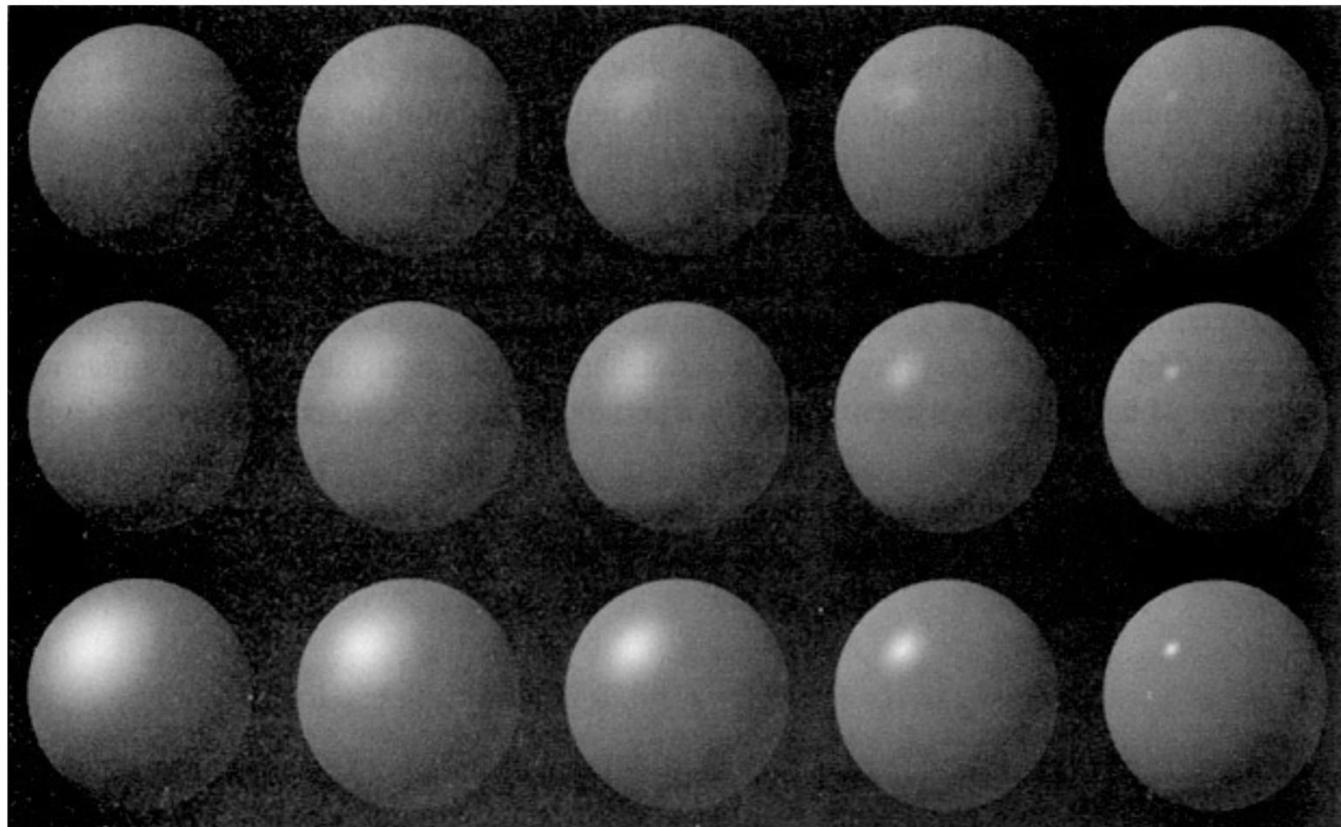
- 标准的余弦相似度会产生很大的高光区域
- 增加指数 p 使反射波变窄
- 在 Blinn-Phong 模型下 p 为 100-200



Blinn-Phong 高光着色

$$L_s = k_s \left(\frac{I}{r^2} \right) \max(0, \mathbf{n} \cdot \mathbf{h})^p$$

k_s



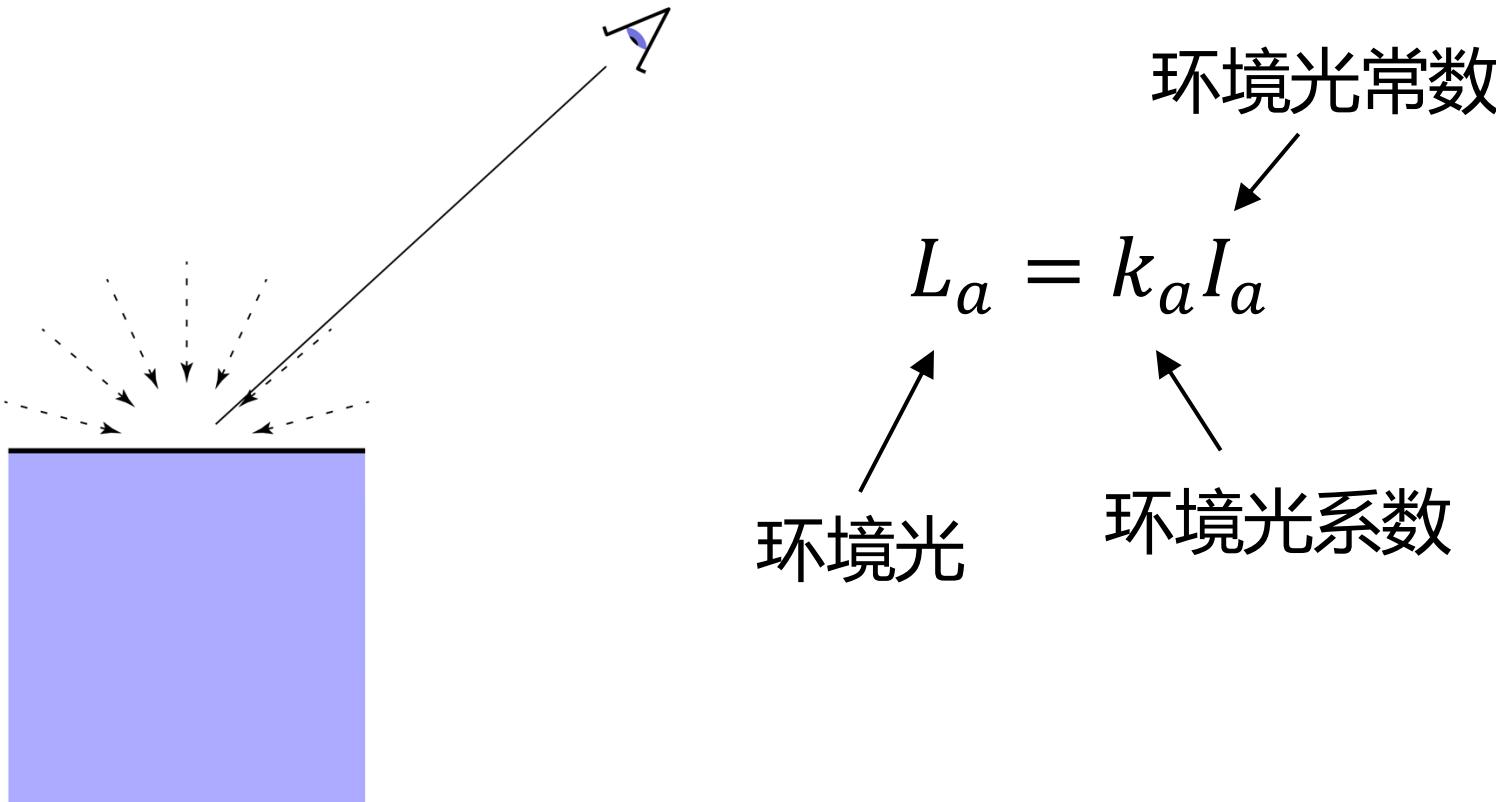
p —→

环境光
Ambient lighting

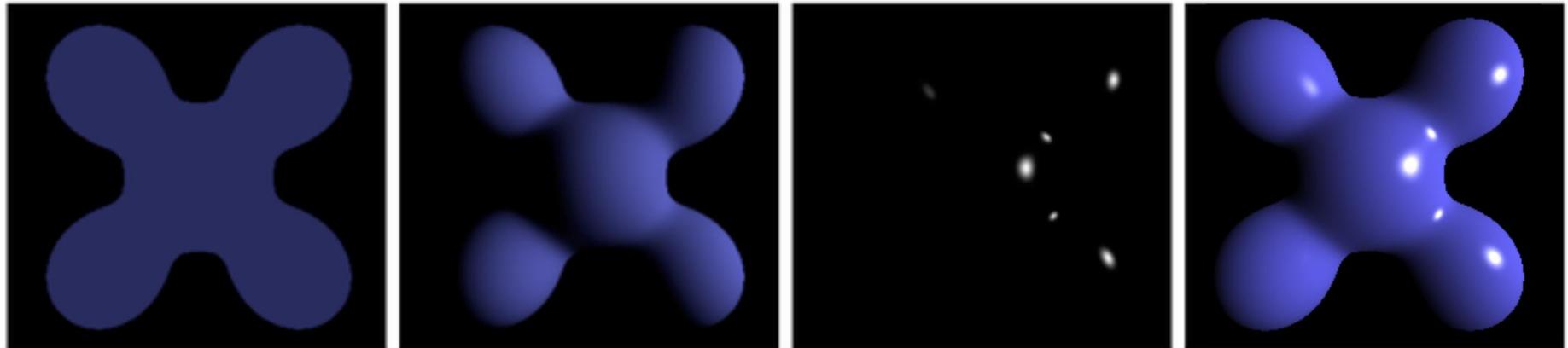
环境光着色

捕捉精确的环境光非常复杂，因此 Blinn-Phong 模型假定环境光着色**不依赖于任何光线**

增加一个固定的常数 I_a 以近似环境光 (呈现一点物体原本的颜色，而不是黑色)



Blinn-Phong 反射模型



Ambient + Diffuse + Specular = Blinn-Phong
Reflection

$$L = L_a + L_d + L_s$$

$$= k_a I_a + k_d \left(\frac{I}{r^2} \right) \max(0, \mathbf{n} \cdot \mathbf{l}) + k_s \left(\frac{I}{r^2} \right) \max(0, \mathbf{n} \cdot \mathbf{h})^p$$

Blinn-Phong 反射模型 – 讨论

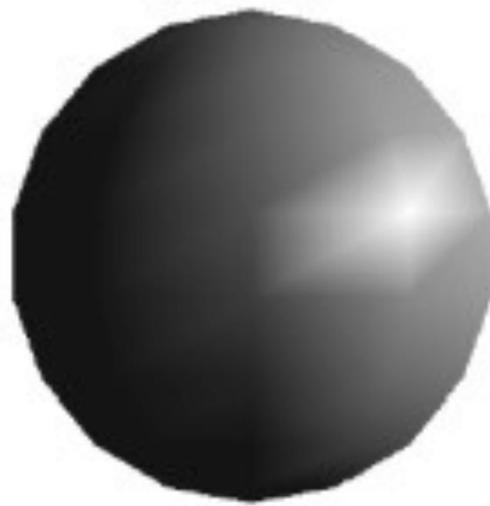
- Blinn-Phong 反射模型是一个相对**简单的经验模型**，不太涉及物理真实性
- 环境光照、漫反射和镜面反射的反射系数是通过实验或观察得到的，**不具有物理上的准确性**
- 是一个**局部光照模型**，无法准确地模拟复杂的光照现象，例如全局光照、间接光照、颜色漂移等
- 适用于**实时渲染**和需要**快速计算**的场合

着色频率 Shading frequencies

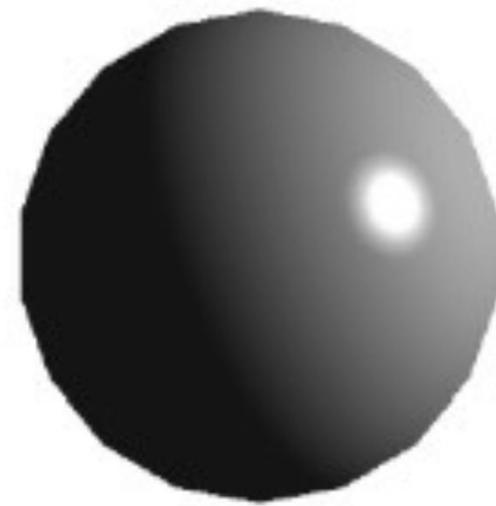
口把着色应用在哪些点上



对每个
多边形
进行着色



对每个顶点
进行着色

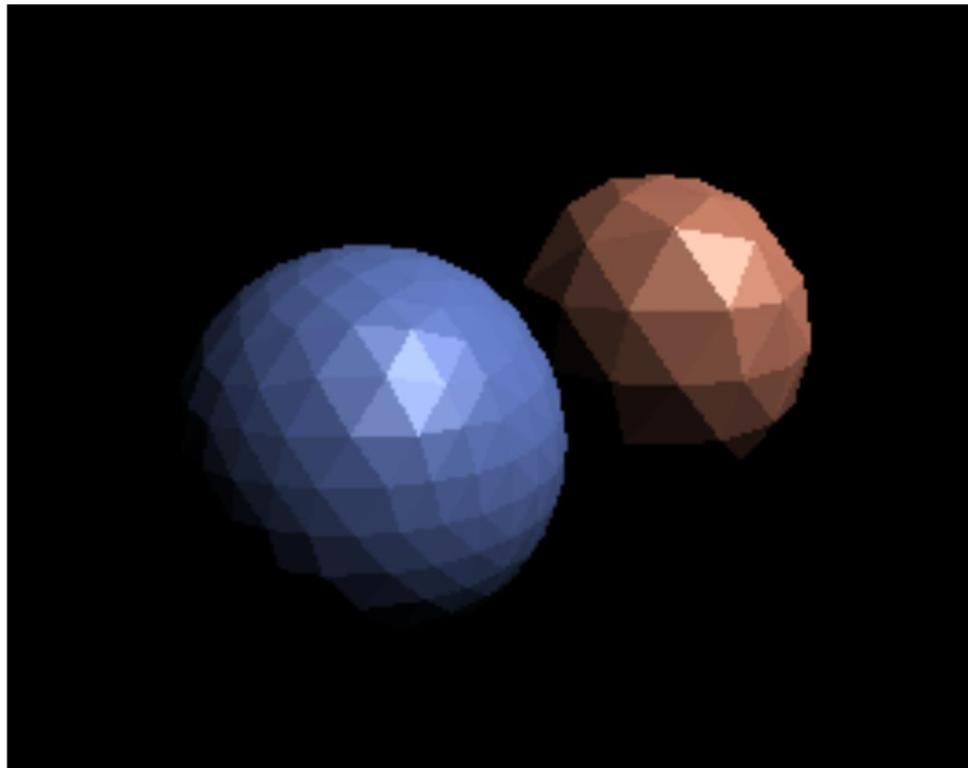


对每个像素
进行着色

对每个三角形进行着色 (平面着色)

□ 平面着色 Flat shading

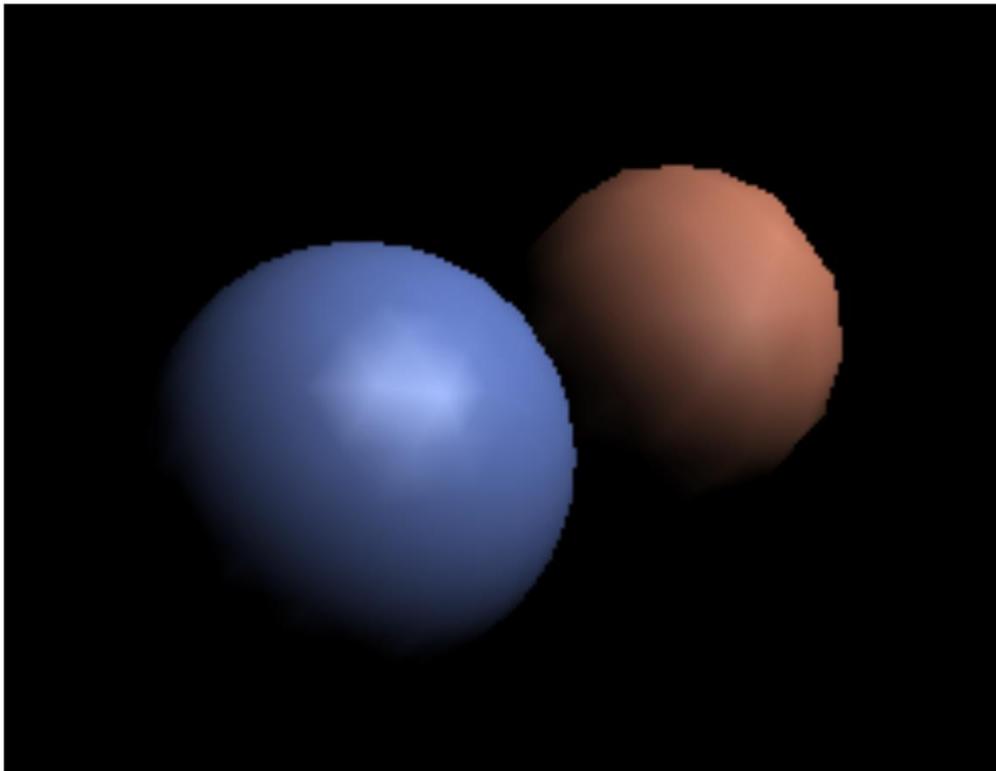
- 三角形是一个平面 (有一个法向量)
- 不适合平滑的曲面



对每个顶点进行着色 (高德洛着色)

□ 高德洛着色 (Gouraud shading)

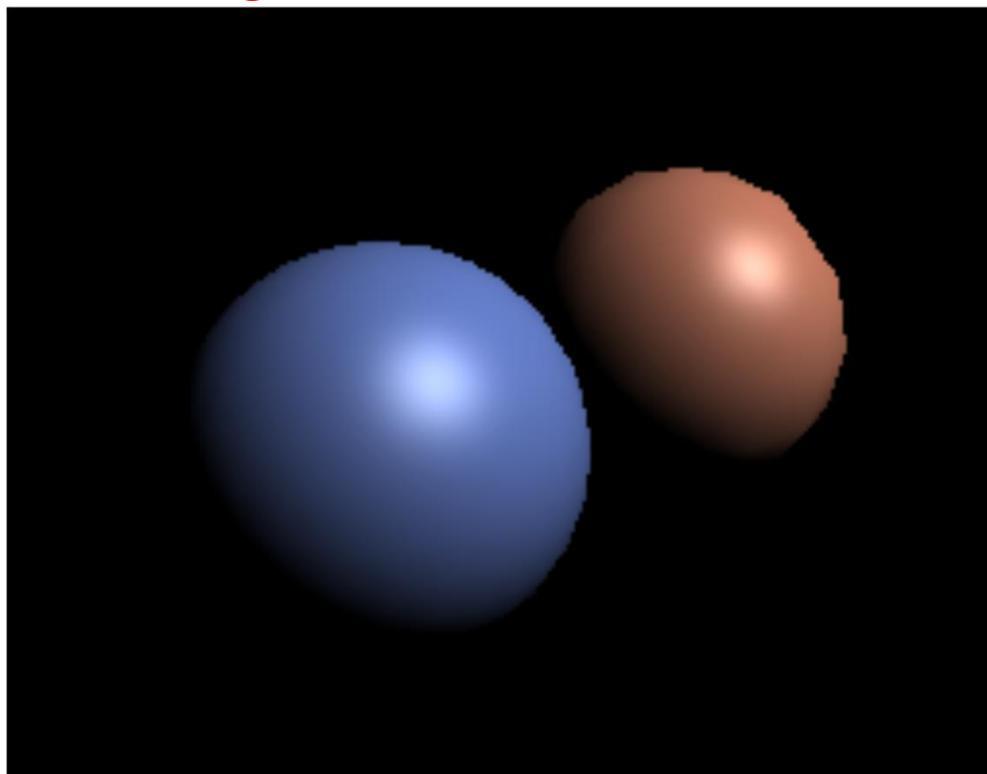
- 基于重心坐标对三角形内部的点进行着色
- 需要计算三角形每个顶点的值
 - 怎么求每个顶点的法向量?



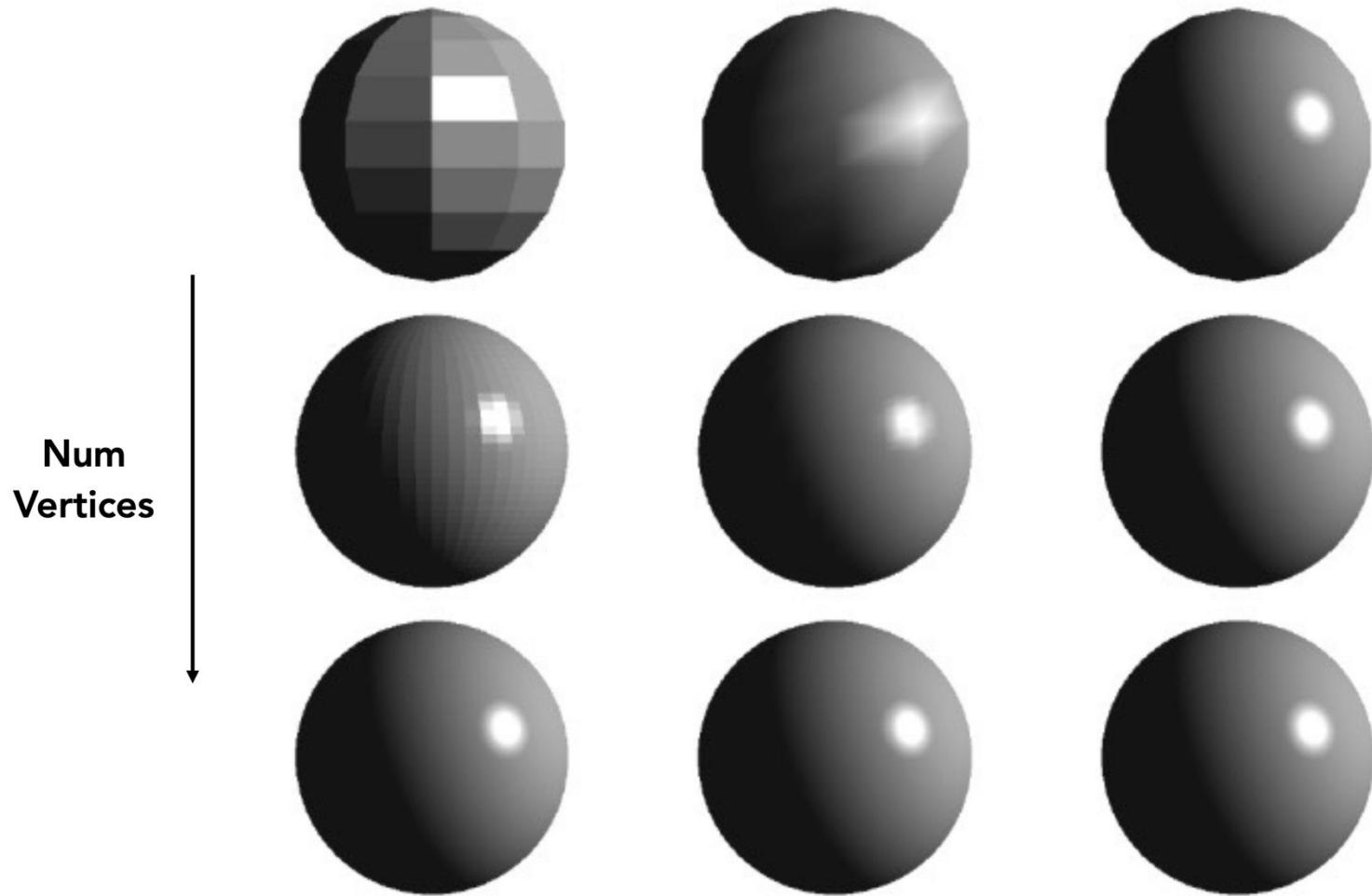
对每个像素进行着色 (Phong 着色法)

□ Phong 着色法 (Phong shading)

- 计算每个像素位置的法向量
- 计算每个像素的着色值
- 不是 Blinn-Phong 反射模型 (由同一个人发明)



着色频率：面、顶点或像素



Shading freq. : Face
Shading type : Flat

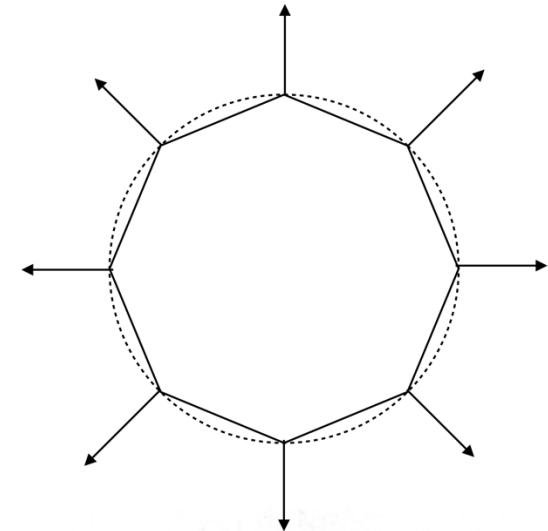
Vertex
Gouraud

Pixel
Phong

计算每个顶点的法向量

□ 从基础几何体获取顶点法线的最佳方式

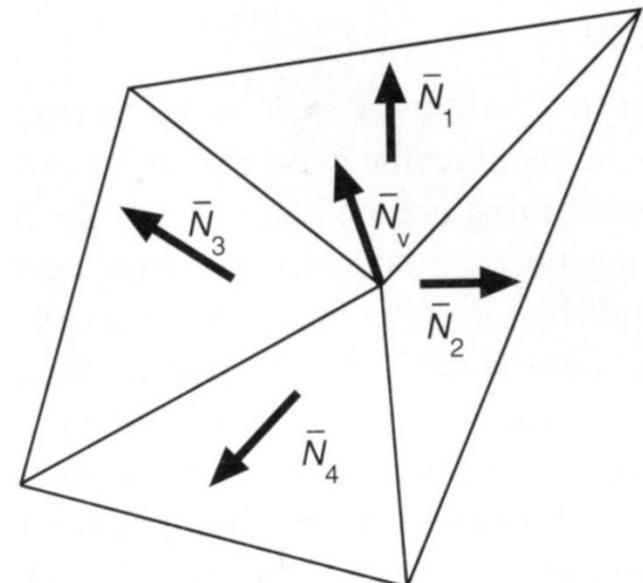
- 比如考虑一个球体



□ 否则必须从三角形面推断顶点法线

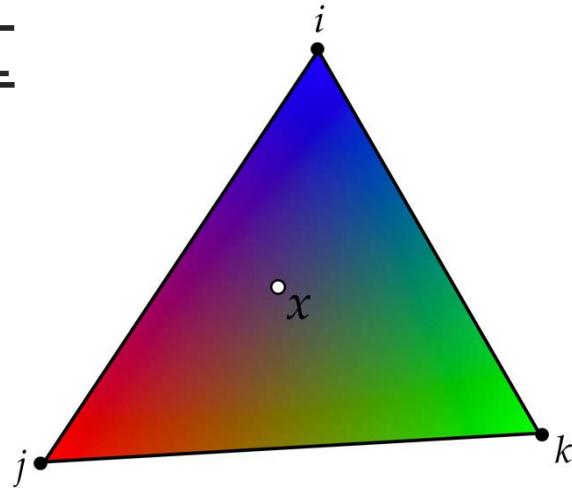
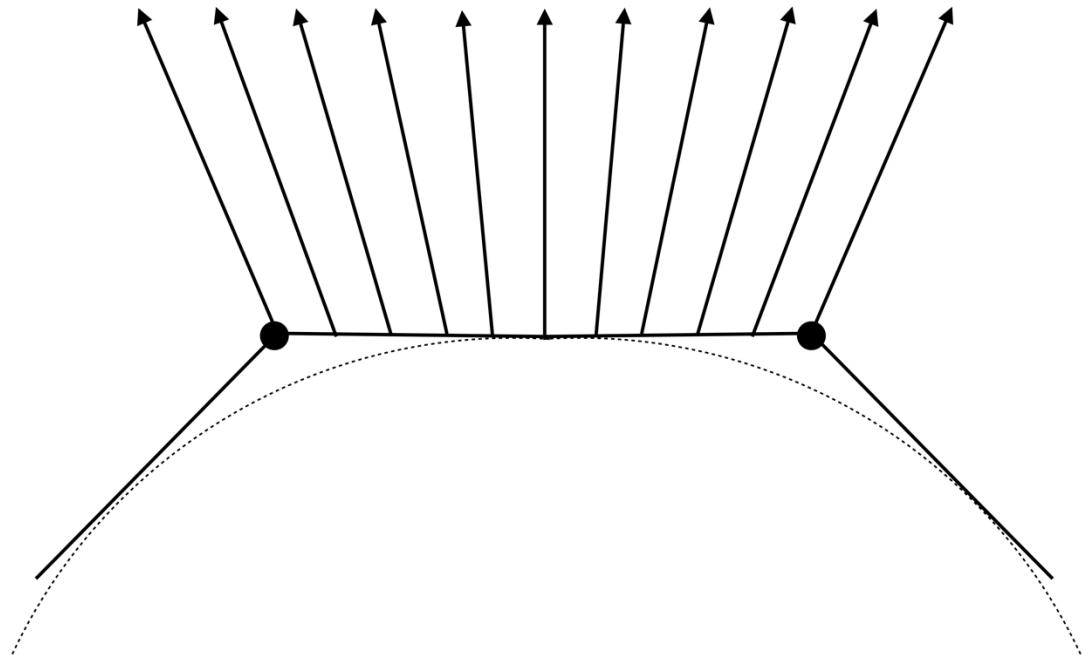
- 一个简单的方案：平均周围面法线

$$N_v = \frac{\sum_i N_i}{\|\sum_i N_i\|}$$



计算三角形内部点的法向量

□利用重心坐标插值三角形顶点的法向量



$$\begin{aligned}color(x) &= color(x_i)\phi_i + \\& color(x_j)\phi_j + color(x_k)\phi_k\end{aligned}$$

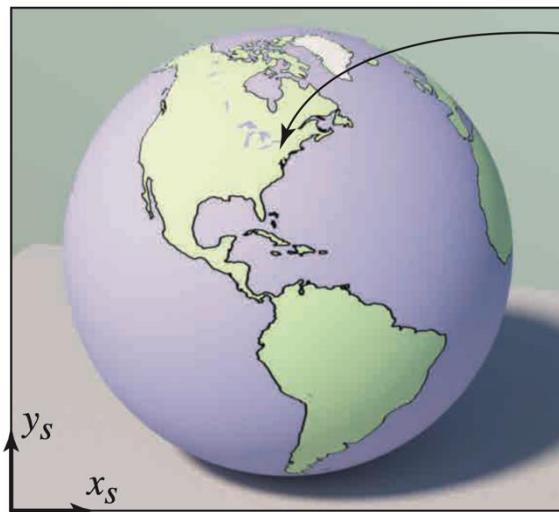
$$\begin{aligned}normal(x) &= normal(x_i)\phi_i \\& + normal(x_j)\phi_j \\& + normal(x_k)\phi_k\end{aligned}$$

□需要对插值后的法向量进行归一化

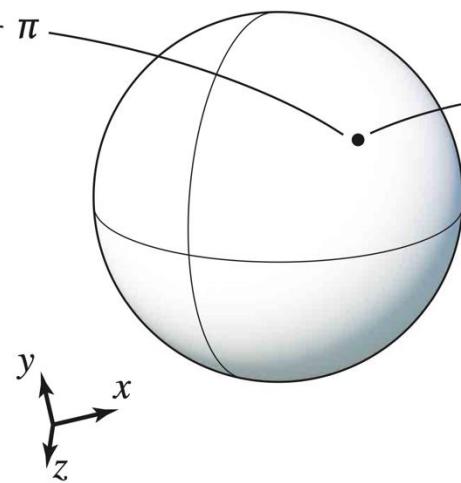
将着色结果绘制到纹理中

纹理映射 Texture mapping

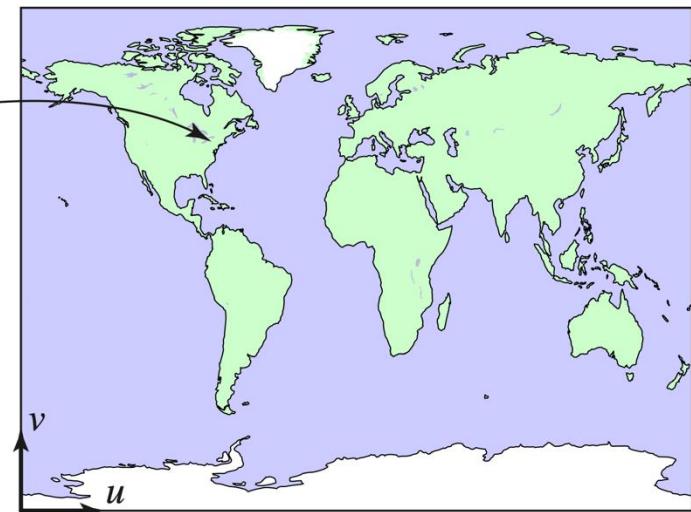
每个 3D 曲面点对应 2D 图像 (纹理) 中的一个点



Screen space



World space



Texture space

环境光贴图 (Environment Map)

- 口将环境编码成纹理，并映射到物体的表面，常用于模拟镜面反射 (如金属和玻璃)
- 口不需要实时计算复杂的光线跟踪便能模拟复杂的光照效果
- 口不能很好地处理动态光照，以及复杂的多次反射和折射



Light from the environment



Rendering with the environment

如果构造环境光的纹理？

一种做法是球形环境贴图 (Spherical Environment Map)

- 先将环境映射到一个虚拟的球面上
- 然后根据视线方向和表面法线在球面上选择对应的纹理映射到物体 (比如汽车、金属、镜子) 当中



球形环境光贴图



Eucalyptus Grove Light Probe
©1999 Paul Debevec
<http://www.debevec.org/Probes>

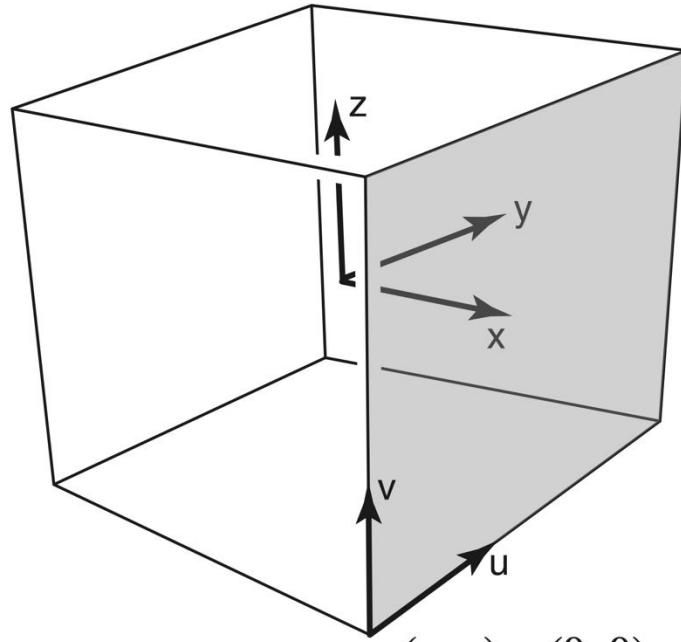
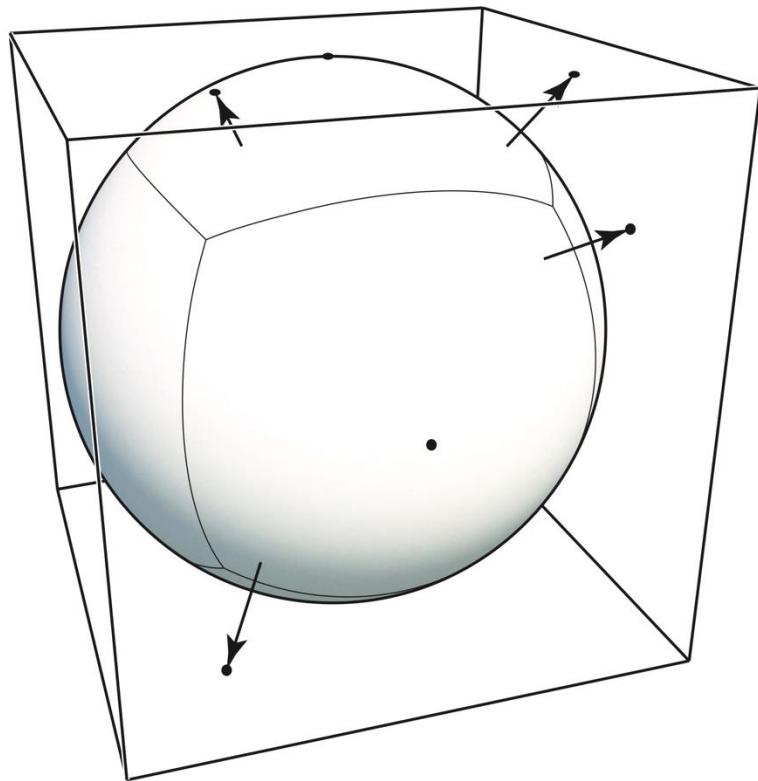
球形环境光贴图展开

容易出现变形 (特别是上面和下面的部分)

- 比如地球的北极和南极比地图上呈现的要大
- 不能准确地反映出环境光



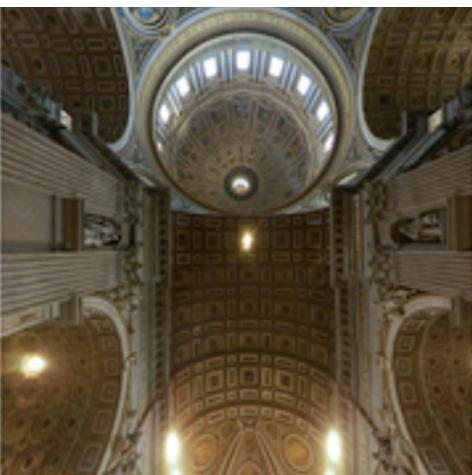
立方体环境贴图



right face has
 $x > |y|$ and
 $x > |z|$

$(u, v) = (0, 0)$
 $x = -y = -z$

- 将环境映射到一个立方体的六个面上，而不是一个球体
- 提供更均匀的分辨率，可以更准确地表示环境的细节

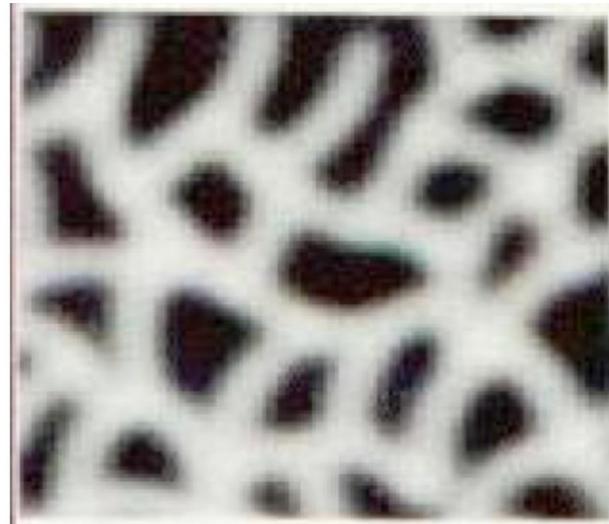


更少的变形!

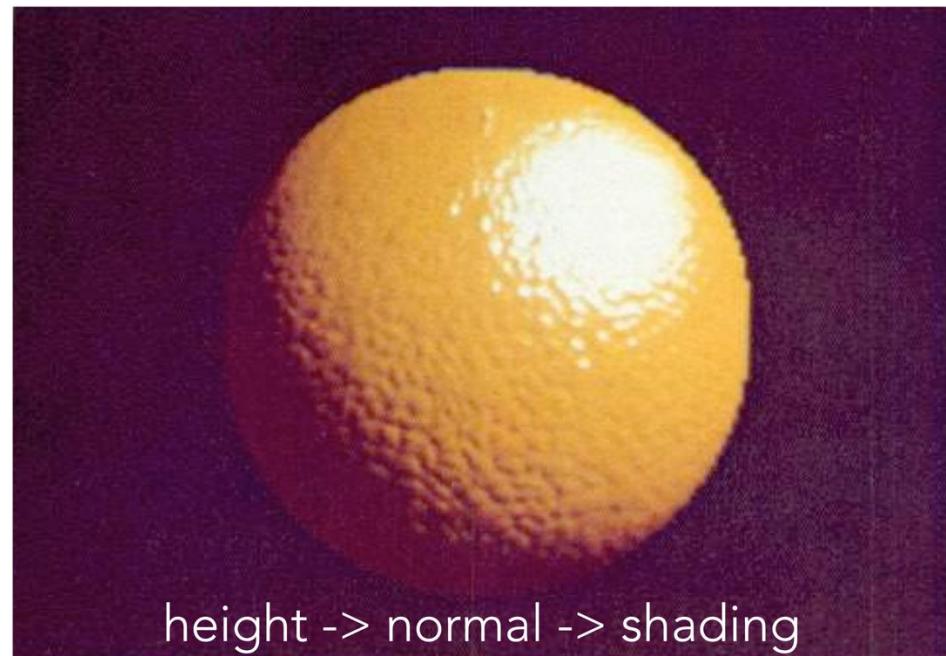
凹凸贴图 Bump mapping

利用纹理映射来改变着色效果

- 纹理可以存储任何信息，比如相对基础模型的高度
- 生成假的几何表面（不需要存储复杂的几何信息）



Relative height to the
underlying surface

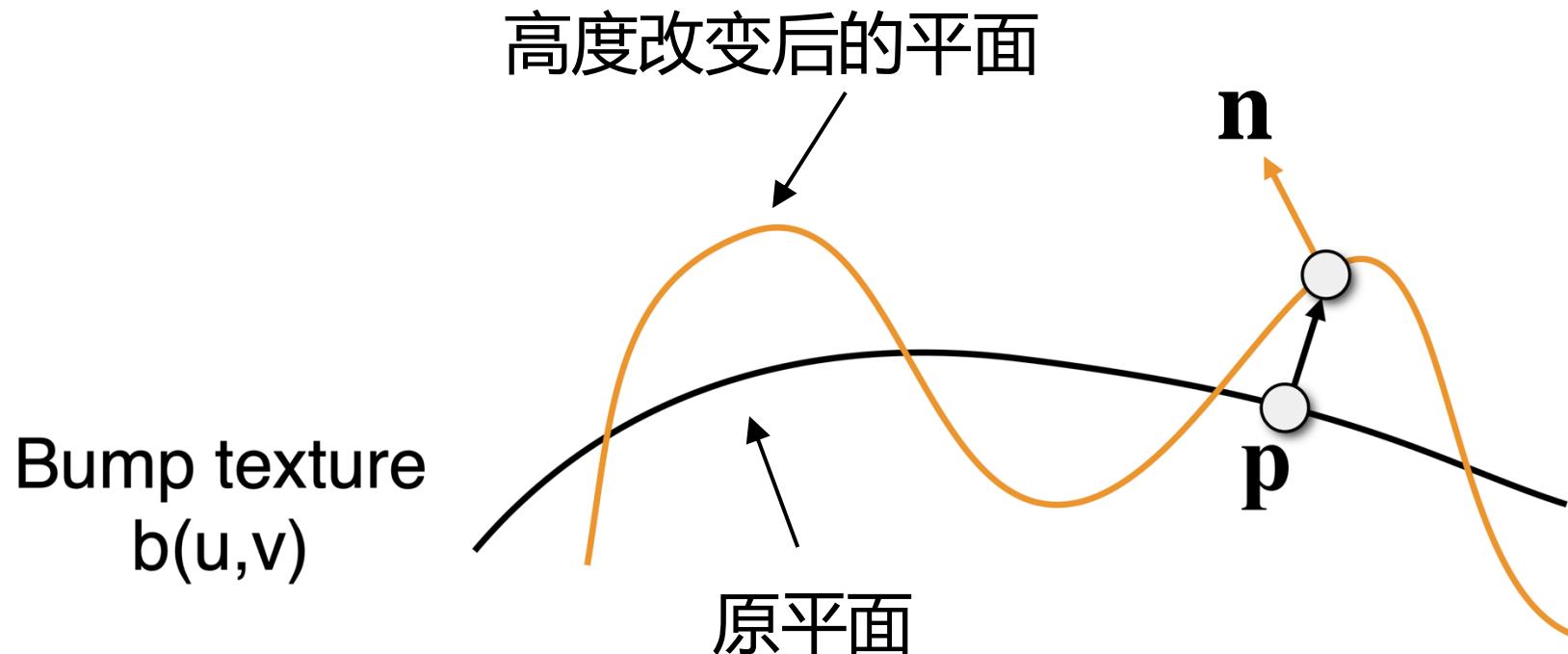


height -> normal -> shading

凹凸贴图 Bump mapping

口添加曲面细节而不添加更多三角形

- 改变每个像素相对于原平面的高度
- 扰动每个像素的曲面法线 (仅用于着色计算)



How about shadows?

阴影贴图

- 如何在光栅化中制造阴影效果?
- 阴影贴图 (Shadow mapping)!



古墓丽影 2018 中的阴影效果

阴影贴图

口是一种基于图像空间的算法

- 即在生成阴影的过程中不需要考虑场景中的几何信息
- 必须处理走样问题

口算法的关键思想

- 不在阴影中的点能同时被光源和相机看到

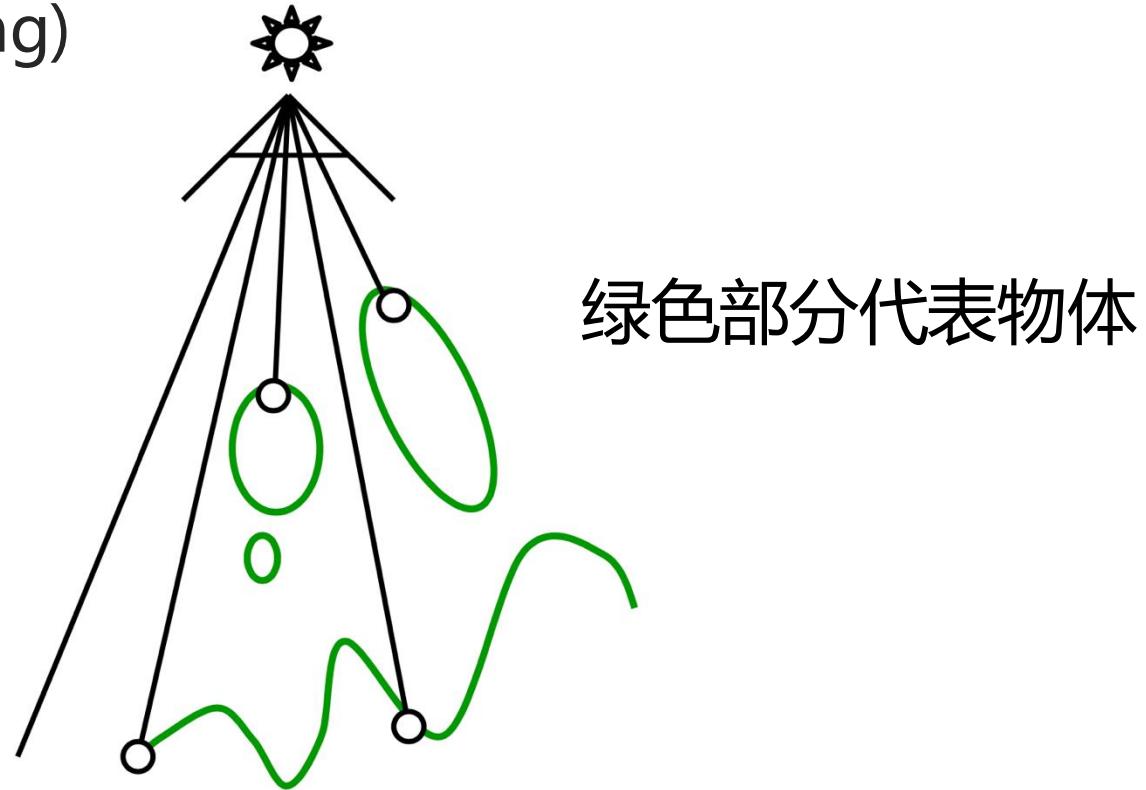
口经典的阴影贴图算法仅能处理点光源

- 产生硬阴影，通常具有很清晰的阴影边界

阶段一：阴影图生成阶段

从光源的视角渲染场景，创建一个深度图

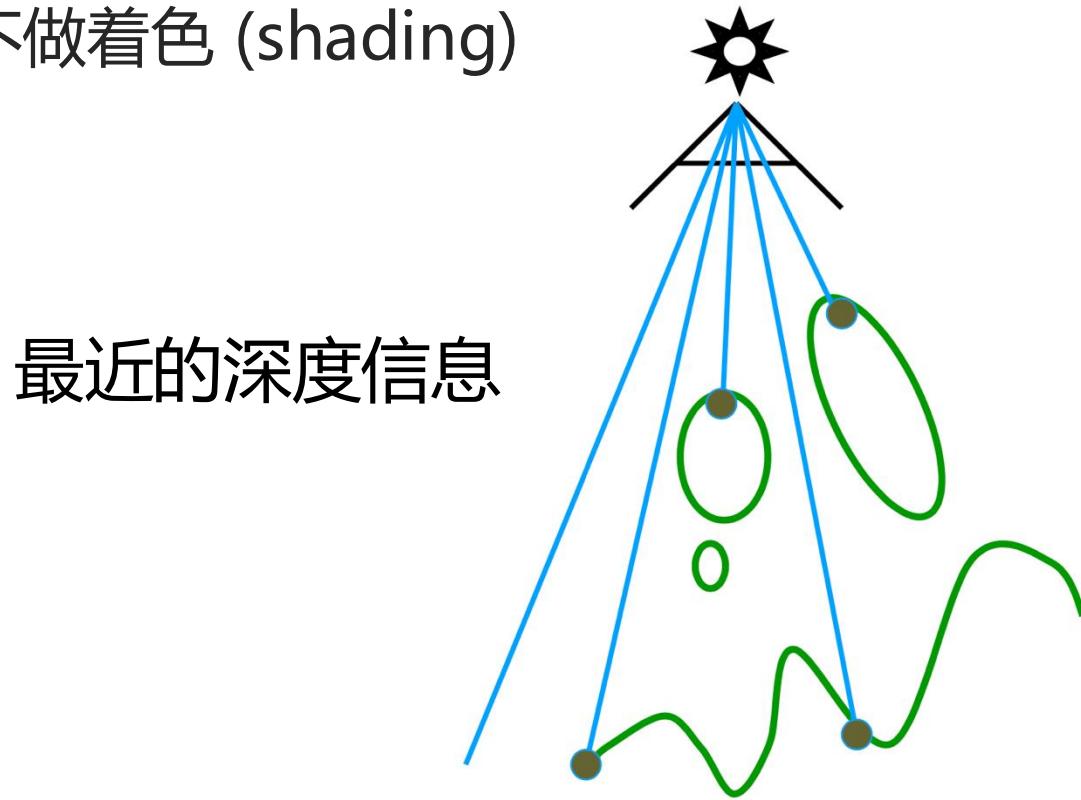
- 深度图中的每个像素存储了对应场景中的点到光源的距离，即 Z-buffer 算法
- 不做着色 (shading)



阶段一：阴影图生成阶段

从光源的视角渲染场景，创建一个深度图

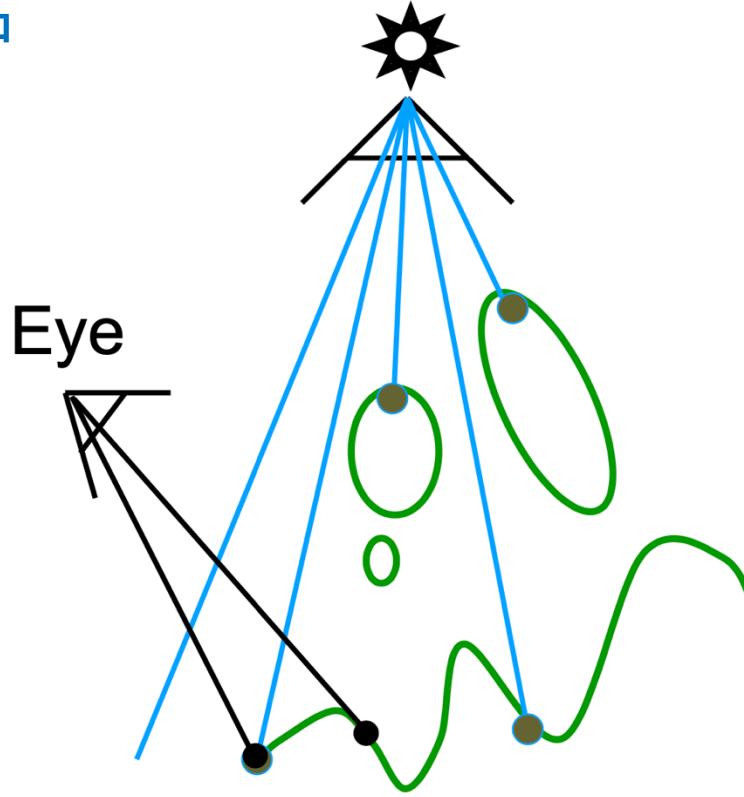
- 深度图中的每个像素存储了对应场景中的点到光源的距离，即 Z-buffer 算法
- 不做着色 (shading)



阶段二：阴影测试阶段

从相机的视角渲染场景

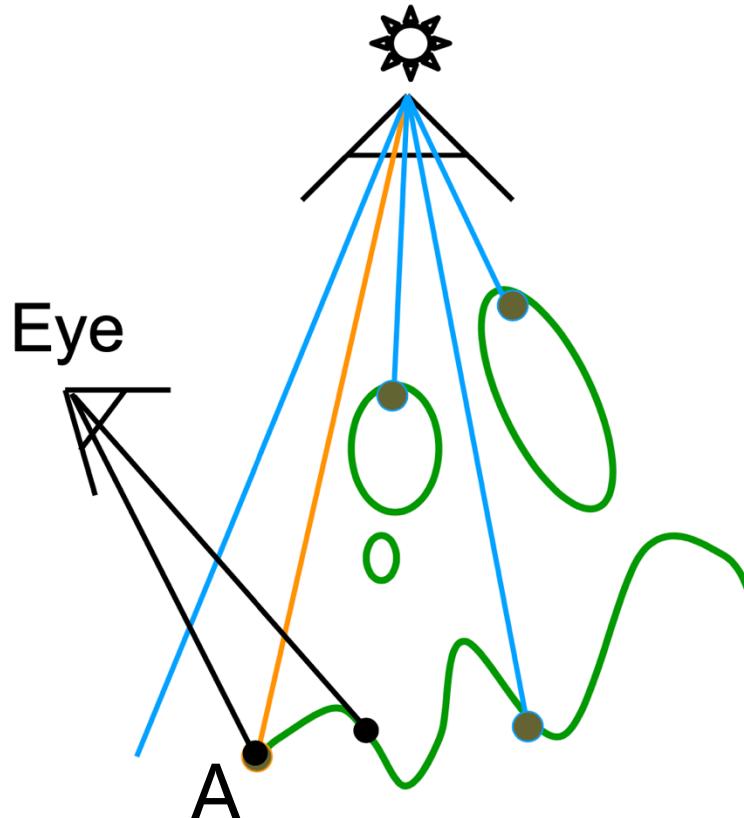
- 将相机看到的每个点投影到光源处 (计算该点到光源的距离)
- 如果这个点的深度值等于深度图中的对应值，那么这个点就不在阴影中



阶段二：阴影测试阶段

从相机的视角渲染场景

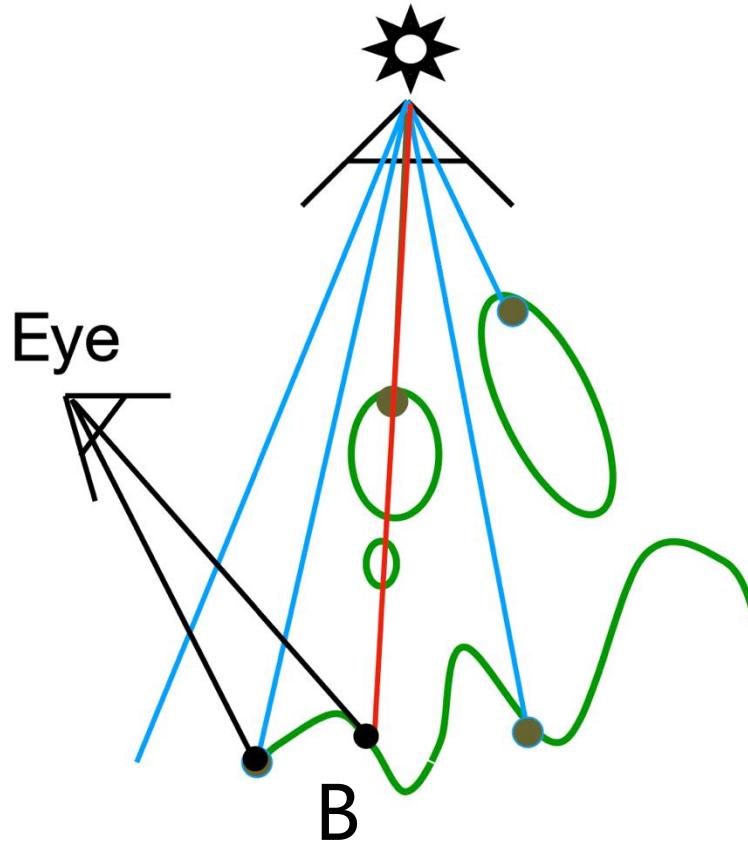
- A 点投影的深度与深度图中记录的对应值相等
- 因此不在阴影中



阶段二：阴影测试阶段

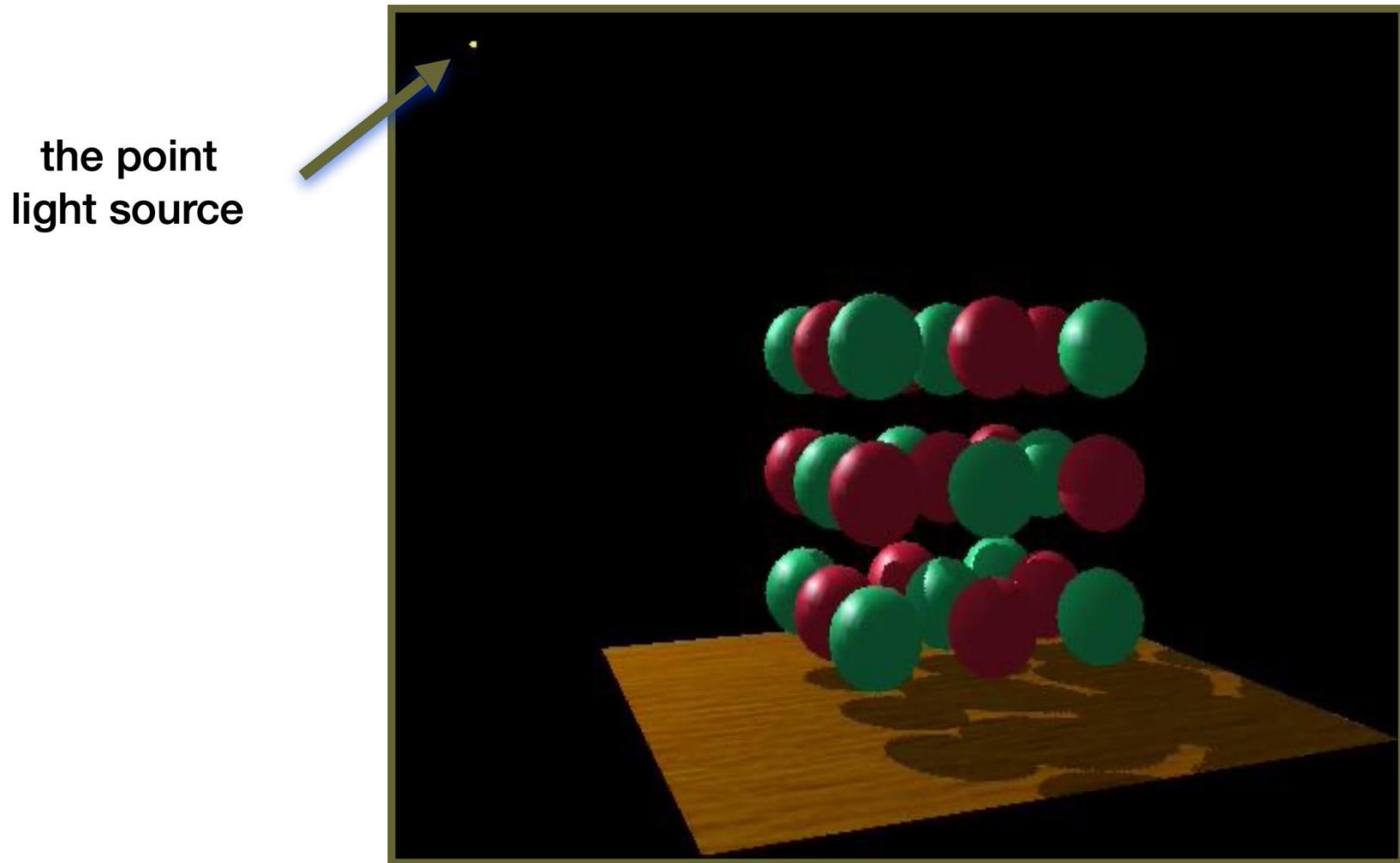
□ 从相机的视角渲染场景

- B 点投影的深度与深度图中记录的对应值不相等
- 因此在阴影中



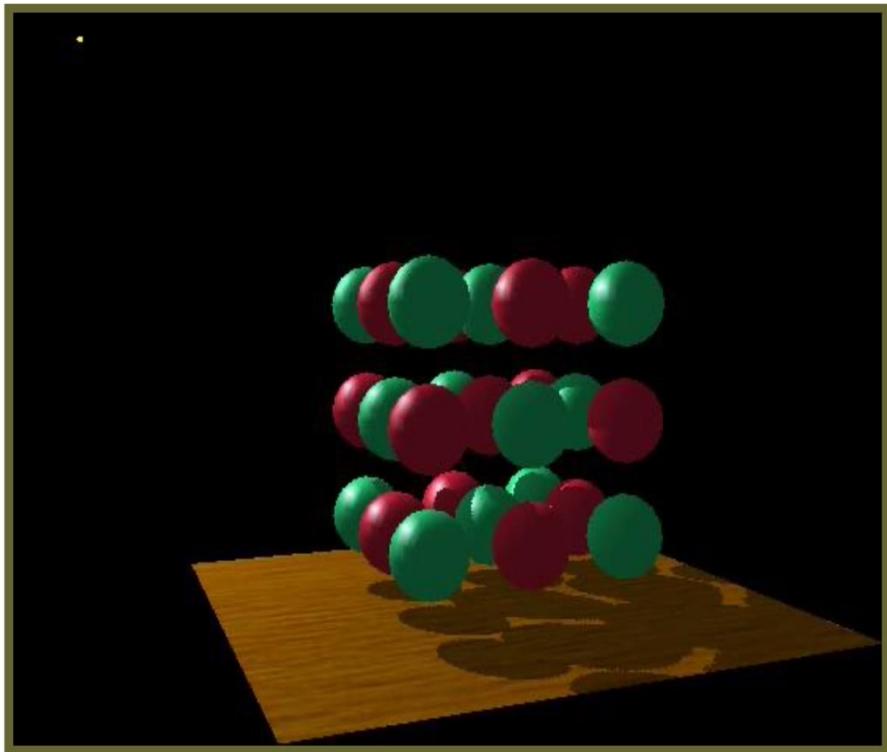
可视化阴影贴图

□一个包含阴影的复杂场景

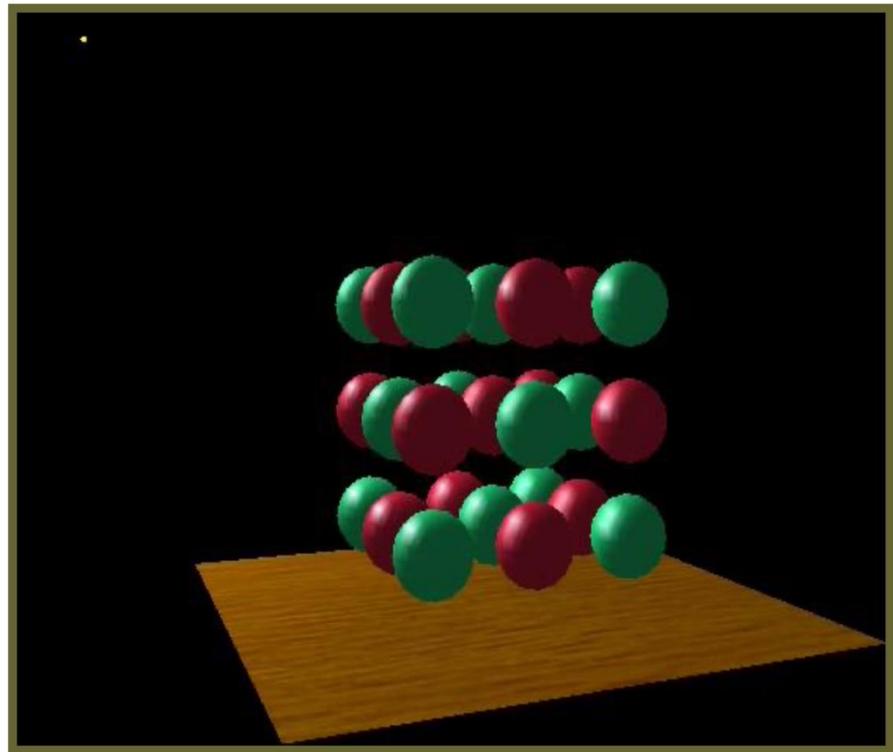


可视化阴影贴图

口有阴影和无阴影的比较



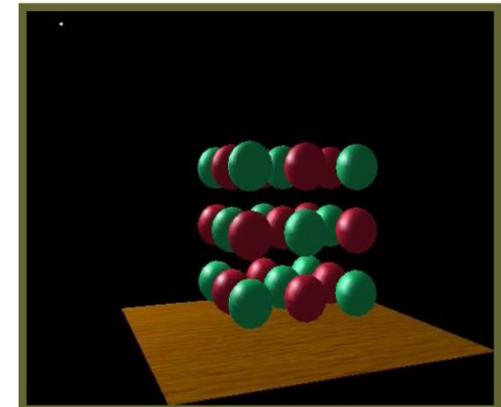
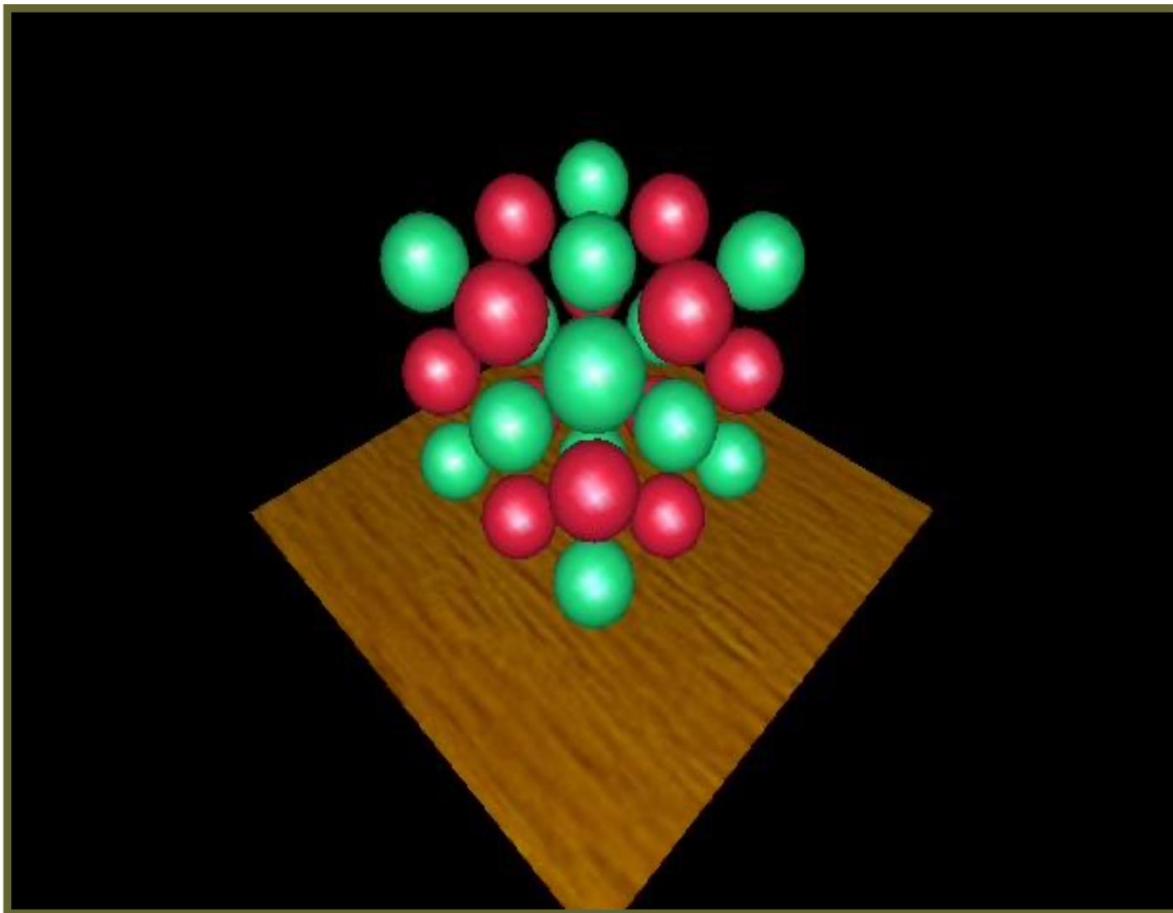
with shadows



without shadows

可视化阴影贴图

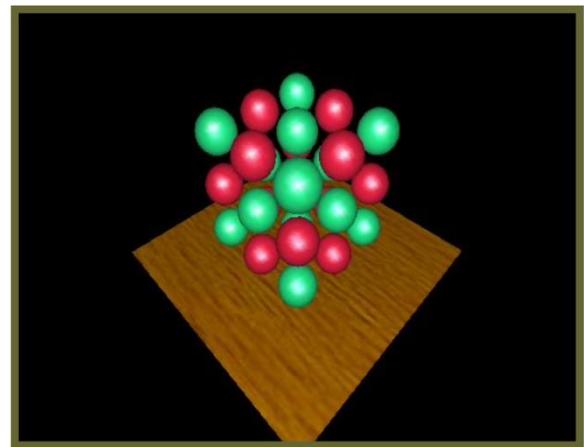
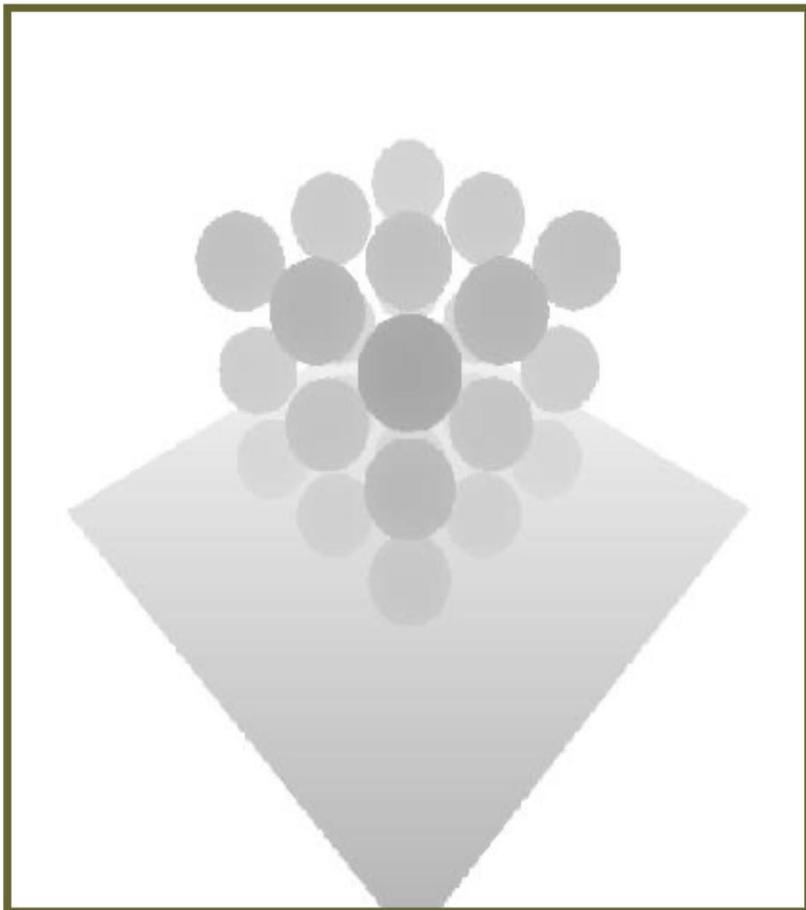
□ 从光源处看到的场景



FYI: from the
eye's point-of-view
again

可视化阴影贴图

口从光源处记录的深度图

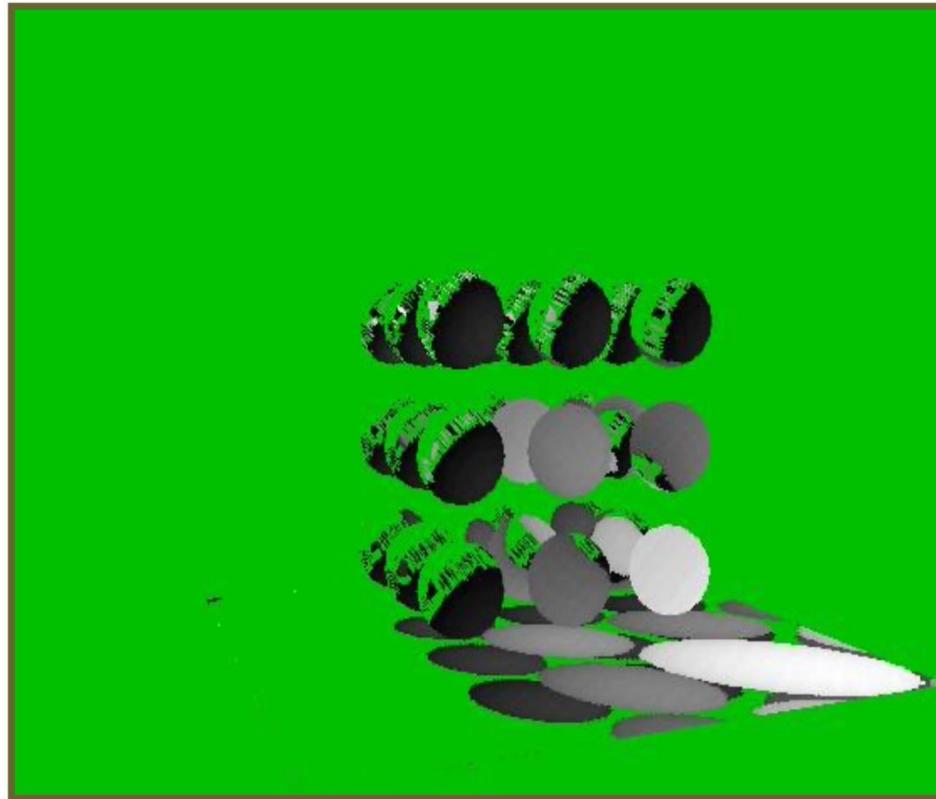


FYI: from the
light's point-of-view
again

可视化阴影贴图

口对比投影的深度与深度图中记录的深度

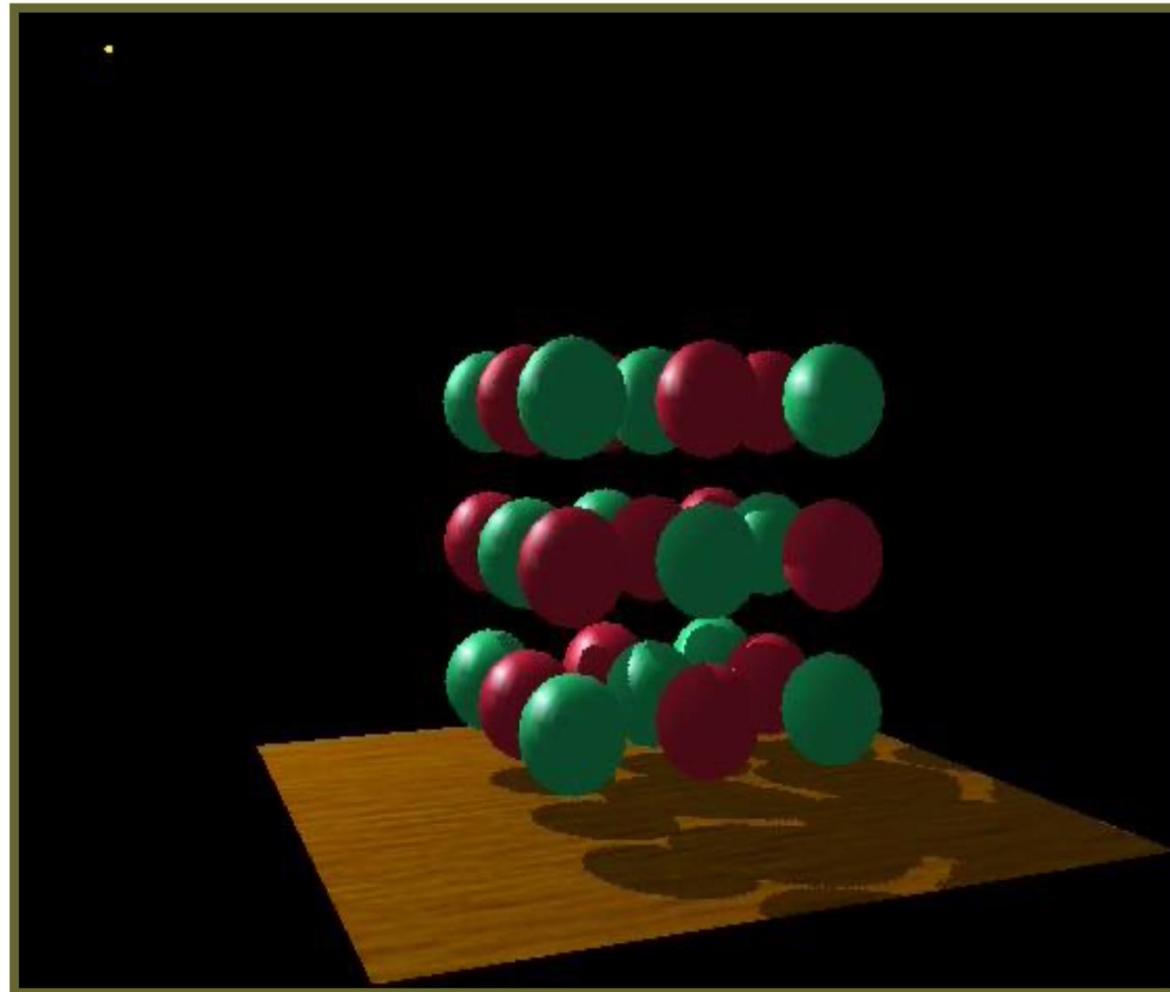
绿色表示二者的值大致相等



非绿色则是阴影所在的区域

可视化阴影贴图

口阴影贴图的结果



阴影贴图的应用

- 早期动画（玩具总动员等）
- **每一个 3D 视频游戏的基本阴影技术！**



Zelda: Breath of the Wild

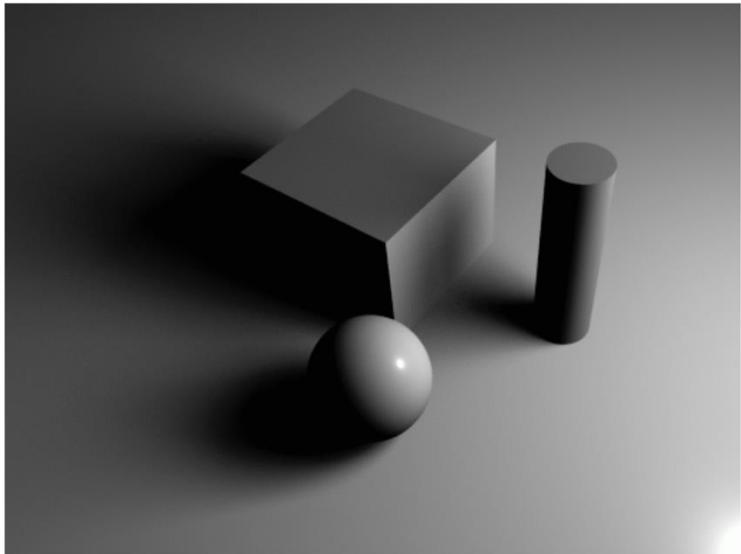
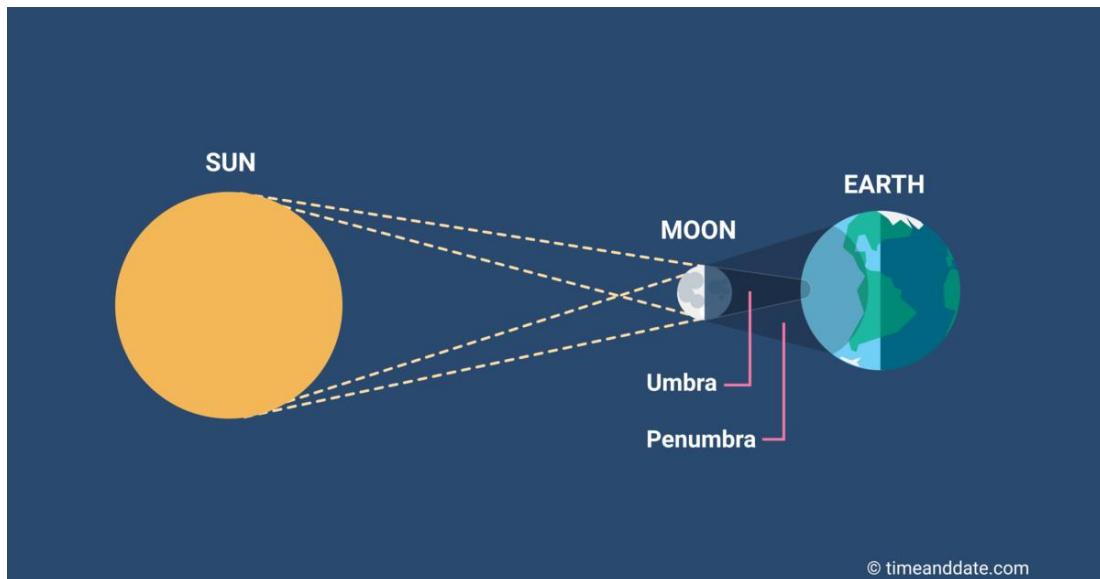
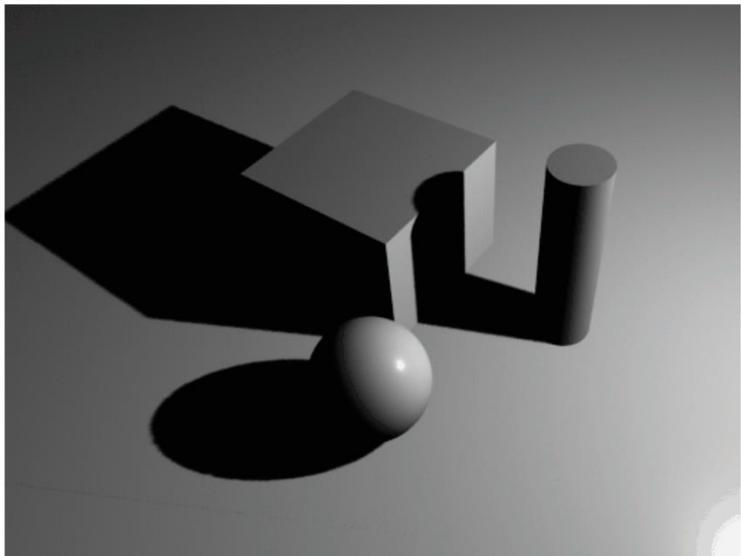


Super Mario Odyssey

阴影贴图的问题

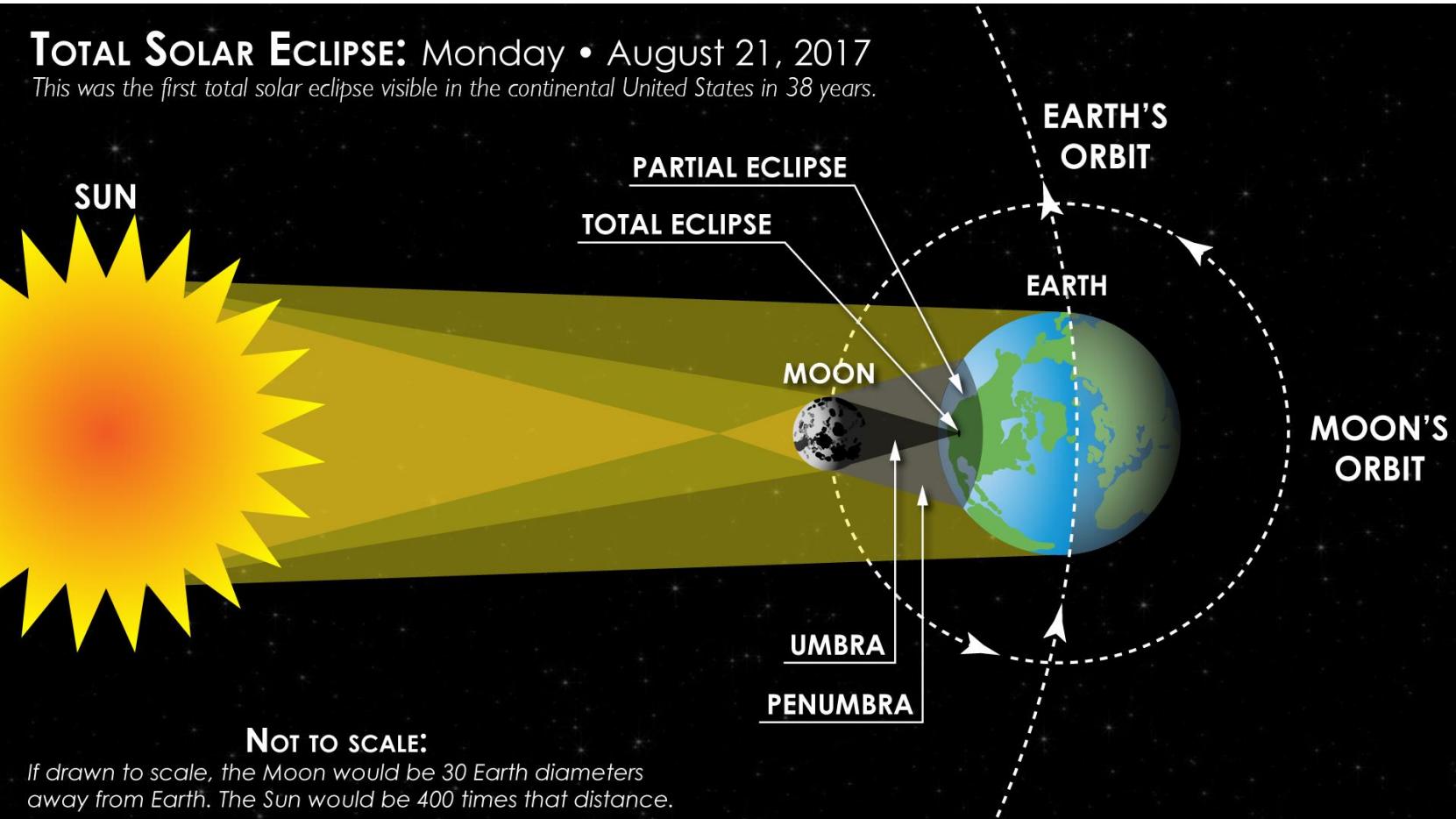
- 仅处理点光源，生成硬阴影
- 质量取决于阴影贴图分辨率
 - 基于图像的技术的一般问题
- 涉及浮点深度值的相等比较
 - 受限于计算机表示浮点数的精度问题

硬阴影和软阴影



- 完全在阴影中的区域称为
本影 (Umbra)
- 部分在阴影中的区域称为
半影 (Penumbra)

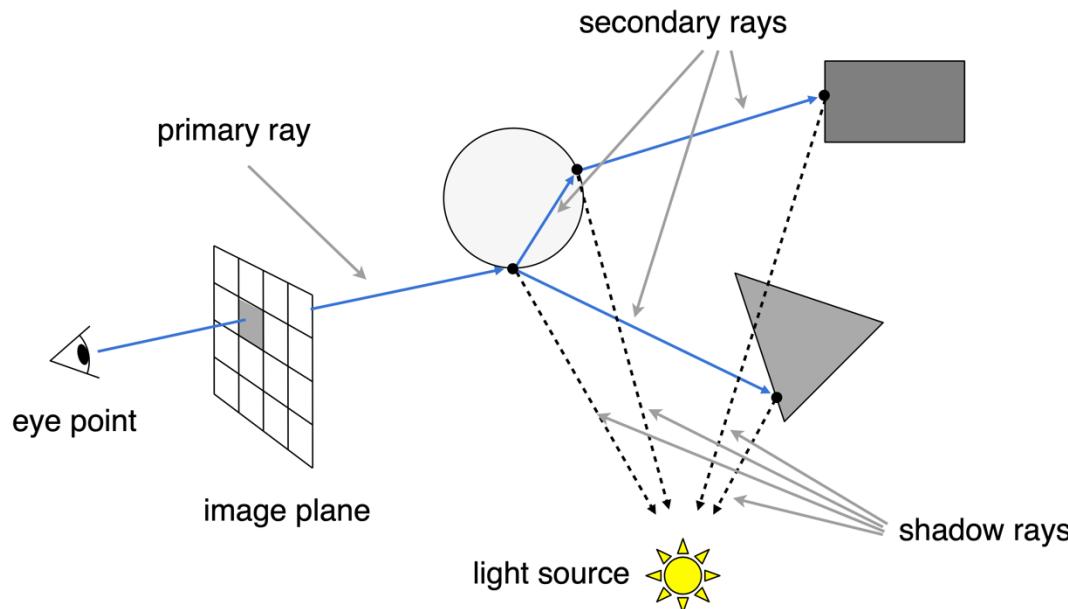
日食



- 点光源不可能出现软阴影
- 如果有软阴影，一定是因为光源有一定大小

下节课：光线追踪

- 口 阴影贴图只是一种近似的阴影生成技术，无法处理光线的反射和散射，也无法生成软阴影或半透明物体的阴影
- 口 光线追踪计算光线路径，生成更真实的阴影、反射和折射等效果
- 口 然而，光线追踪的计算量大，在实时渲染（如电子游戏）中，阴影贴图仍然是一种常用的阴影生成技术





中山大學 软件工程学院
SUN YAT-SEN UNIVERSITY SCHOOL OF SOFTWARE ENGINEERING

谢谢

陈壮彬
软件工程学院
chenzhb36@mail.sysu.edu.cn