

Shadow Mapping

draw shadows without ray tracing

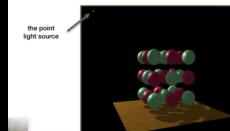
- An Image-space Algorithm
 - no knowledge of scene's geometry during shadow computation
 - must deal with aliasing artifacts
- Key idea:
 - the points NOT in shadow must be seen both **by the light** and **by the camera**

得到 a shadowmap 只处理单光源

visualizing shadow mapping

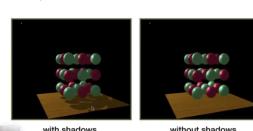
Visualizing Shadow Mapping

- A fairly complex scene with shadows



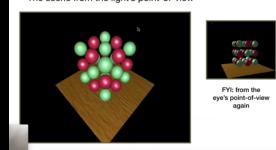
Visualizing Shadow Mapping

- Compare with and without shadows



Visualizing Shadow Mapping

- The scene from the light's point-of-view

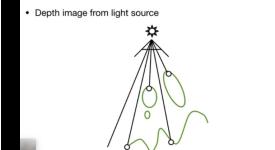


Visualizing Shadow Mapping

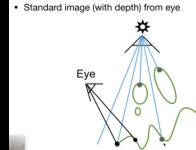
- Depth buffer from the light's point-of-view



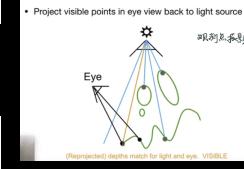
Pass 1: Render from Light



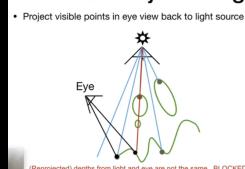
Pass 2A: Render from Eye



Pass 2B: Project to light



Pass 2B: Project to light

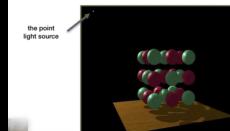


得到 a shadowmap 只处理单光源

visualizing shadow mapping

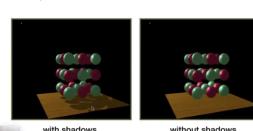
Visualizing Shadow Mapping

- A fairly complex scene with shadows



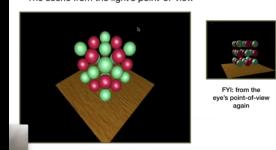
Visualizing Shadow Mapping

- Compare with and without shadows



Visualizing Shadow Mapping

- The scene from the light's point-of-view



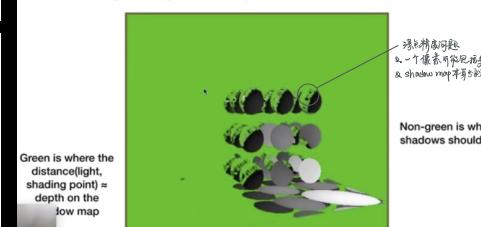
Visualizing Shadow Mapping

- Depth buffer from the light's point-of-view



Visualizing Shadow Mapping

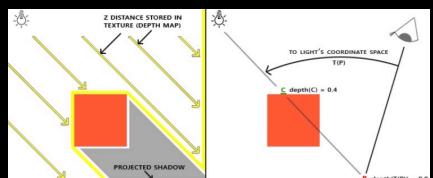
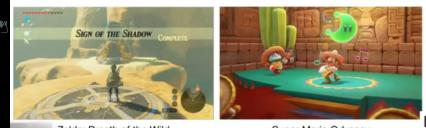
- Comparing $\text{Dist}(\text{light}, \text{shading point})$ with shadow map



Shadow Mapping

- Well known rendering technique

- Basic shadowing technique for early animations (Toy Story, etc.) and in EVERY 3D video game

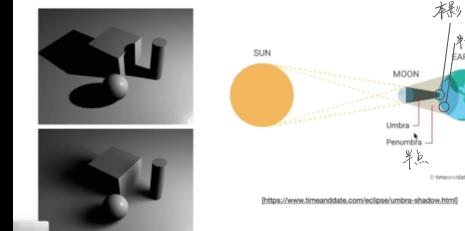


Problems

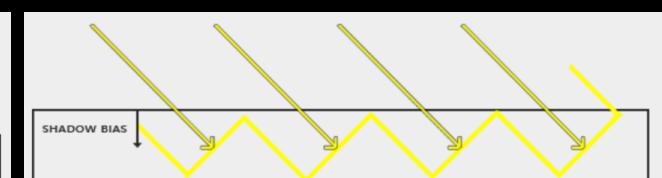
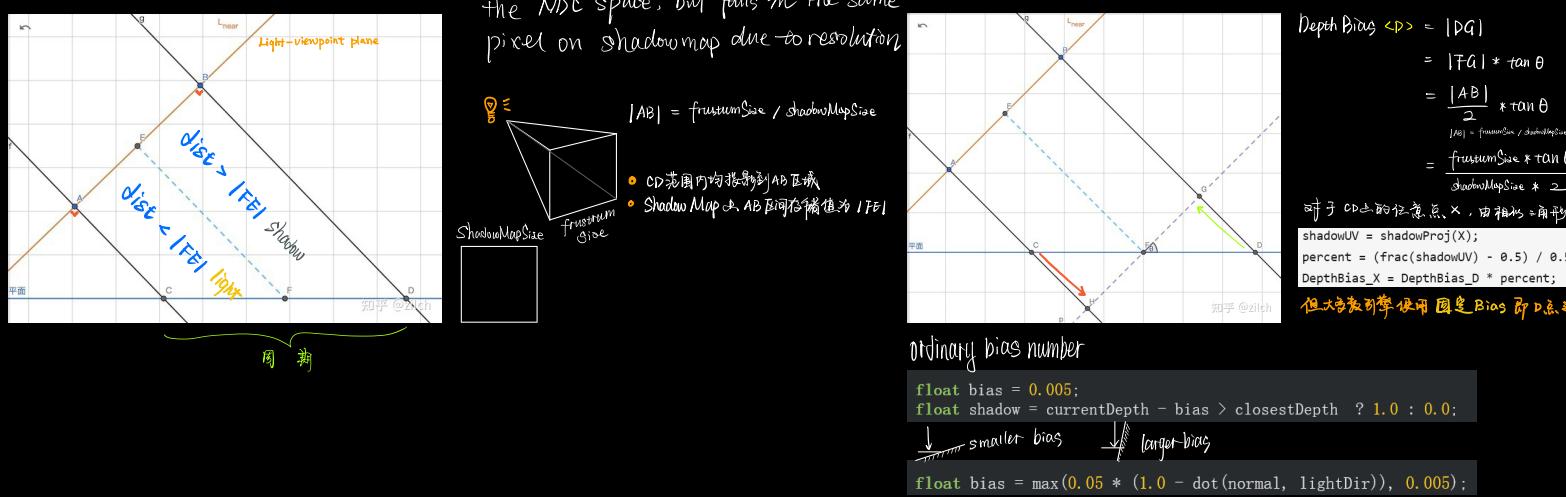
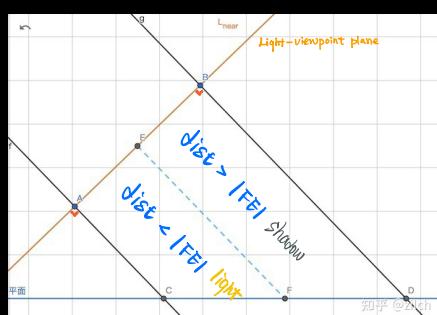
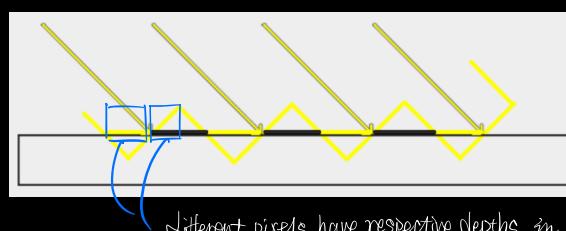
Problems with shadow maps

- Hard shadows (point lights only)
- Quality depends on shadow map resolution (general problem with image-based techniques)
- Involves equality comparison of floating point depth values means issues of scale, bias, tolerance

- Hard shadows vs. soft shadows



Shadow acne Bias



`float bias = 0.005;`
`float shadow = currentDepth - bias > closestDepth ? 1.0 : 0.0;`

`↓ smaller bias ↓ larger bias`

`float bias = max(0.05 * (1.0 - dot(normal, lightDir)), 0.005);`

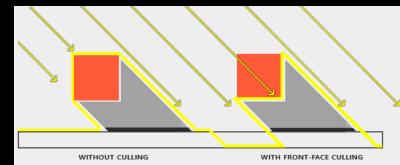
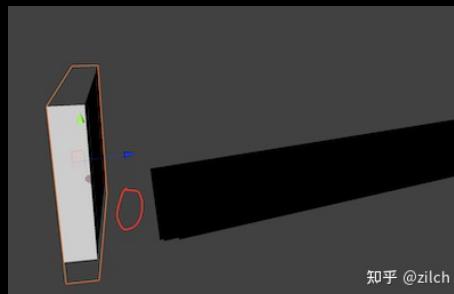
圆是 Bias 问题

漏光问题



统一前移 IDGI 导致上溢溢出，遮挡物之前

Bias 越向无穷大问题



front-face culling could help · BUT only works for closing model

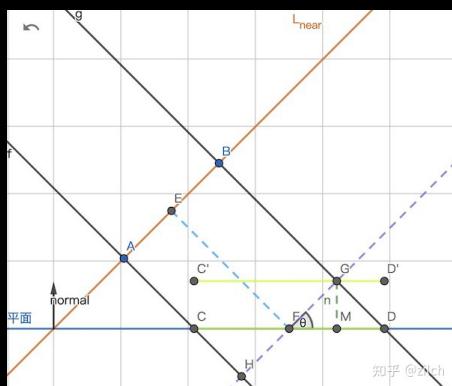
知乎 @zilch

当入射光与平面夹角趋于 0°

$$\frac{\text{frustumSize} \times \tan \theta}{\text{shadowMapSize} \times 2} = |IDGI| \rightarrow \text{无穷大}$$

Solution?

Normal Bias



朝光面 法线方向偏移

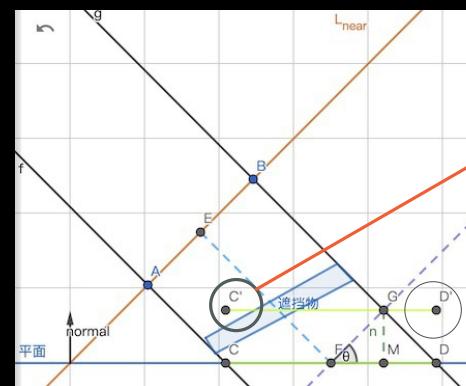
将 CP 朝法线方向平移至 MI

C'C' 移深模糊于 1 像素，点亮
D'D' 移至最近隔壁像素，亦点亮

$$\begin{aligned} \text{NormalBias}(X) &= |GM| \\ &= |GP| * \sin(\theta) \\ &= \frac{\text{frustumSize} * \sin(\theta)}{\text{shadowMapSize} * 2} \end{aligned}$$

当 θ 趋于 0°, bias → $\frac{\text{frustumSize}}{\text{shadowMapSize} * 2}$

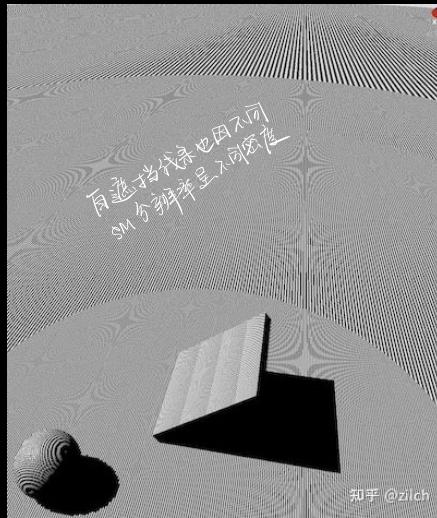
漏光问题



漏光

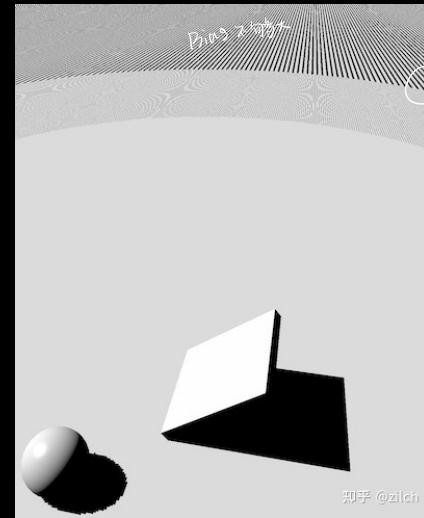
可能漏光

Cascaded Shadow Map 级联阴影贴图



由最近远距离不同
ShadowMap 分辨率

计算某系的 Bias 时需知 cascade index
级数，由此获得对应 shadowMap 分辨率
并得知 Bias



若仍用统一 Bias

调大则
近处漏光

6. 常见的计算优化

在实际的实现中，为了避免三角函数运算，通常可以使用 $1 - \text{dot}(\text{lightDir}, \text{normal})$ 来近似代替 \tan 和 \sin ，然后暴露出两个系数 C_{depth} 和 C_{normal} ，给予使用者进行调整。

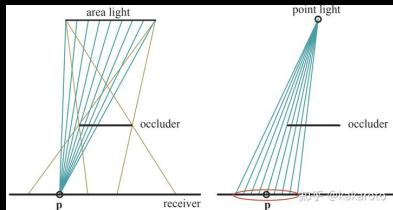
于是 bias 的公式就变成如下形式：

$$\begin{aligned} A &= (1 + \text{ceil}(R)) * \frac{\text{frustumSize}}{\text{shadowMapSize} * 2} \\ B &= 1 - \text{dot}(\text{lightDir}, \text{normal}) \\ \text{DepthBias} &= C_{depth} * A * B \\ \text{NormalBias} &= C_{normal} * A * B \end{aligned}$$

其中 A 可以在 CPU 端完成计算，以上的实现在大多数情况下都可以取得不错的效果。

常见计算优化

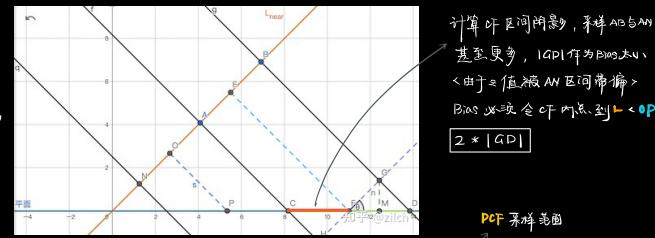
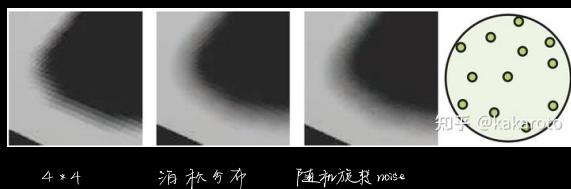
Percentage - closer filtering PCF



normal sampling
soft shadow

PCF approximation

- 并非基于物理，仅依赖接收表面
吉遮挡抑制距离不影响结果
- 采样方式影响结果



易知，该推导可以扩展到任意采样半径R。即：

$$DepthBias = (1 + \text{ceil}(R)) * \frac{\text{frustumSize} * \tan(\theta)}{\text{shadowMapSize} * 2}$$

• ceil为向上取整函数

同样的，对于Normal Bias我们也可以推出其满足这个比例系数，即：

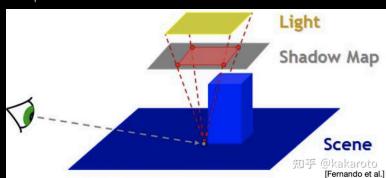
$$NormalBias = (1 + \text{ceil}(R)) * \frac{\text{frustumSize} * \sin(\theta)}{\text{shadowMapSize} * 2}$$

Percentage - closer soft shadows PCSS



在 PCF 基础上，根据与光源及着色点的距离，利用三角形原理，动态计算 PCF 采样范围。

① 计算 Δ_{receiver} 着色点与平均遮挡物距离



从点向光源方向投影
并在 shadowMap 上划范围
△ 该范围的深度值将被用
于采样并计算平均遮挡距离



② 计算 w_{penumbra} 采样范围

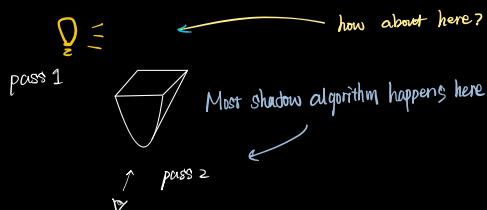
3.3 后续的工作*

关于PCSS的一个缺陷是遮挡物采样区域很大，使用旋转泊松采样图案可以降低采样不足导致的纹理，而对于运动物体的泊松采样存在不稳定性，Jimenez提出一个介于抖动和随机之间的模型来改善结果。

Sikachev提出了一个快速方法称为contact hardening shadows (CHS)，这种方法解决了PCSS的另一个问题，即软阴影部分依赖于shadow map分辨率。CHS方法是首次生成shadow map的mipmaps，然后根据用户定义的世界空间尺寸，使用适当尺寸的mip等级。当得到PCF采样范围后，在高分辨率的mip等级处查询软阴影，在低分辨率mip等级处查询软阴影。

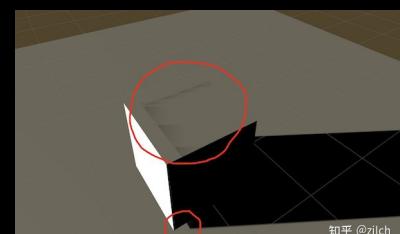
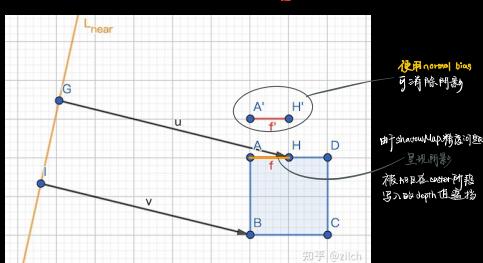
基于CHS的算法和应用有很多，并产生层级结构min/max shadow map概念，即每一阶会存储两张mipmaps，其中一张存储了这个区域内最大的深度值(HIZ)，另一张存储了最小的深度值。若看色点z轴深度大于其HIZ的值，那么必然大于这个区域所有的深度值，则看色点就处于阴影中而不必再去采样，提高了计算效率。

Shadow Caster Vertex Based Bias SCVBB used in Unity URP



在 shadow caster 阶段，会遮蔽抑制 [基于顶点 反向偏移]

- ✓ 相比于基于像素的 shadow receiver 阶段计算
更加廉价
- ✗ 精度：Bias值与光线及法线夹角相关，只考虑顶点不够
- ✗ Bias mismatch. Caster阶段的 normal bias ≠ Receiver阶段



知乎 @zilch