计算机图形学

童 伟 华　管理科研楼1205室
E-mail: tongwh@ustc.edu.cn

中国科学技术大学　数学科学学院
http://math.ustc.edu.cn/

第二节 辐射度方法

介绍

- 光线跟踪算法适合于处理场景中的镜面反射表面
  - 真实场景往往含有漫反射表面
- 绘制方程描述一般的绘制方程
  - 计算量大
  - 双向反射函数维数很高，不易于建模
- 辐射度方法
  - 在假定场景中的物体均为理想漫反射表面的情况下，求解绘制方程
  - 与视点无关，只需求解一次
  - 辐射度方法可与其他绘制算法配合工作，譬如光线跟踪算法等

术语

- 能（energy）～光（入射光，透射光）
  - 必定守恒
- 能通量（flux）＝光通量（luminous flux）＝能量（power）＝单位时间的能
  - 用流明（lumens）表示
  - 与波长有关，因此需要把光通量效率曲线对波谱进行积分
- 能密度(Φ)＝单位面积上的能通量

术语

- 密度～亮度
  - 亮度是可感知的

  =单位立体角单位面积上的通量

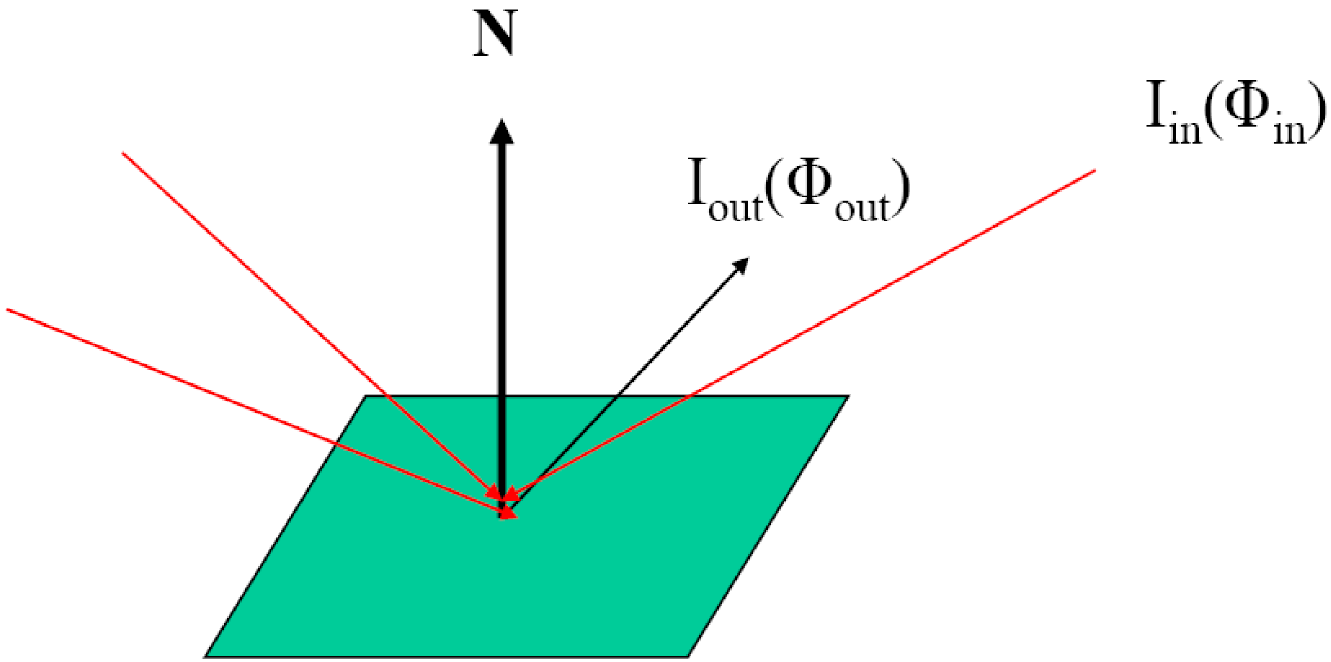
  =单位立体角单位投影面积上的能

  - 用candela（亮度单位， The candela is the luminous intensity, in a given direction, of a source that emits monochromatic radiation of frequency 540×10^{12} hertz and that has a radiant intensity in that direction of $^1/_{683}$ watt per steradian）进行度量

  $\Phi = \int\int 1\, dA\, d\omega$

光照方程（Kajiya）

- 考虑表面上的一点



光照方程

- 出射光有两种来源
  - 发射光
  - 反射的入射光
- 必须对所有的入射光进行积分
  - 在半球上进行积分
- 必须考虑入射光的衰减因素

光照方程

$$I_{out}(\Phi_{out}) = E(\Phi_{out}) + \int_{2\pi} R_{bd}(\Phi_{out}, \Phi_{in}) I_{in}(\Phi_{in}) \cos\theta \, \mathrm{d}\omega$$

发射光

法向与Φ_{in}的夹角

双向反射系数

光照方程

- 光照方程是能量平衡的状态
  - 进入的能量 = 发出的能量
- 在半球上进行积分
- Fredholm积分方程
  - 一般没有解析解
- 对于R_{bd}的各种近似可以给出各种标准的光照模型
- 为了考虑对象的遮挡关系，应当在右边前面加上遮挡项

另外一种表示

- 考虑在p点来自于p'点的光

$$i(\mathbf{p}, \mathbf{p}') = \upsilon(\mathbf{p}, \mathbf{p}')(\varepsilon(\mathbf{p},\mathbf{p}') + \int \rho(\mathbf{p}, \mathbf{p}', \mathbf{p}'')i(\mathbf{p}', \mathbf{p}'')d\mathbf{p}''$$

遮挡因子 = 0 或 $1/d^2$

从p'到p的发射项

在p'对来自于所有p''的光朝向p点的反射

辐射度

- 考虑把对象剖分为平坦的曲面片（这可能对应于模型中的多边形）
- 假设每个曲面片为完美的漫反射界面
- 辐射度＝通量＝单位面积单位时间内离开曲面片的能量

记号

n个曲面片的编号为1到n

b_i = 曲面片i的辐射度

a_i = 曲面片i的面积

离开曲面片i的总亮度 = $b_i\, a_i$

$e_i\, a_i$ = 从曲面片i反射的亮度

ρ_i = 曲面片i的反射率

f_{ij} = 形状因子 = 从曲面片j离开的能量到达曲面片i的比率

辐射度方程

能量平衡

$$b_i a_i = e_i a_i + \rho_i \sum f_{ji} b_j a_j$$

互反律

$$f_{ij} a_i = f_{ji} a_j$$

辐射度方程

$$b_i = e_i + \rho_i \sum f_{ij} b_j$$

矩阵形式

$$\mathbf{b} = [b_i]$$

$$\mathbf{e} = [e_i]$$

$$\mathbf{R} = [r_{ij}] \quad r_{ij} = \rho_i \text{ if } i \neq j \quad r_{ii} = 0$$

$$\mathbf{F} = [f_{ij}]$$

矩阵形式

- 方程为

$$b = e - RFb$$

形式上的解为

$$b = [I - RF]^{-1} e$$

没有实用性，因为n通常非常大

- 另外一种方法应用到F为稀疏矩阵
- 稍后考虑形状因子的确定

求解辐射度方程

- 对于稀疏矩阵，如果采用迭代方法求解，那么通常每次迭代只需要O(n)的计算量

Jacobi方法

$$b_i^{k+1} = e_i - \sum_{j=1}^{n} \rho_i f_{ij} b_j^k, \quad i = 1, 2, \ldots, n$$

Gauss-Seidel方法：需要进行中间过程的更新

$$b_i^{k+1} = e_i - \sum_{j=1}^{i-1} \rho_i f_{ij} b_j^{k+1} - \sum_{j=i+1}^{n} \rho_i f_{ij} b_j^{k+1}, \quad i = 1, 2, \ldots, n$$

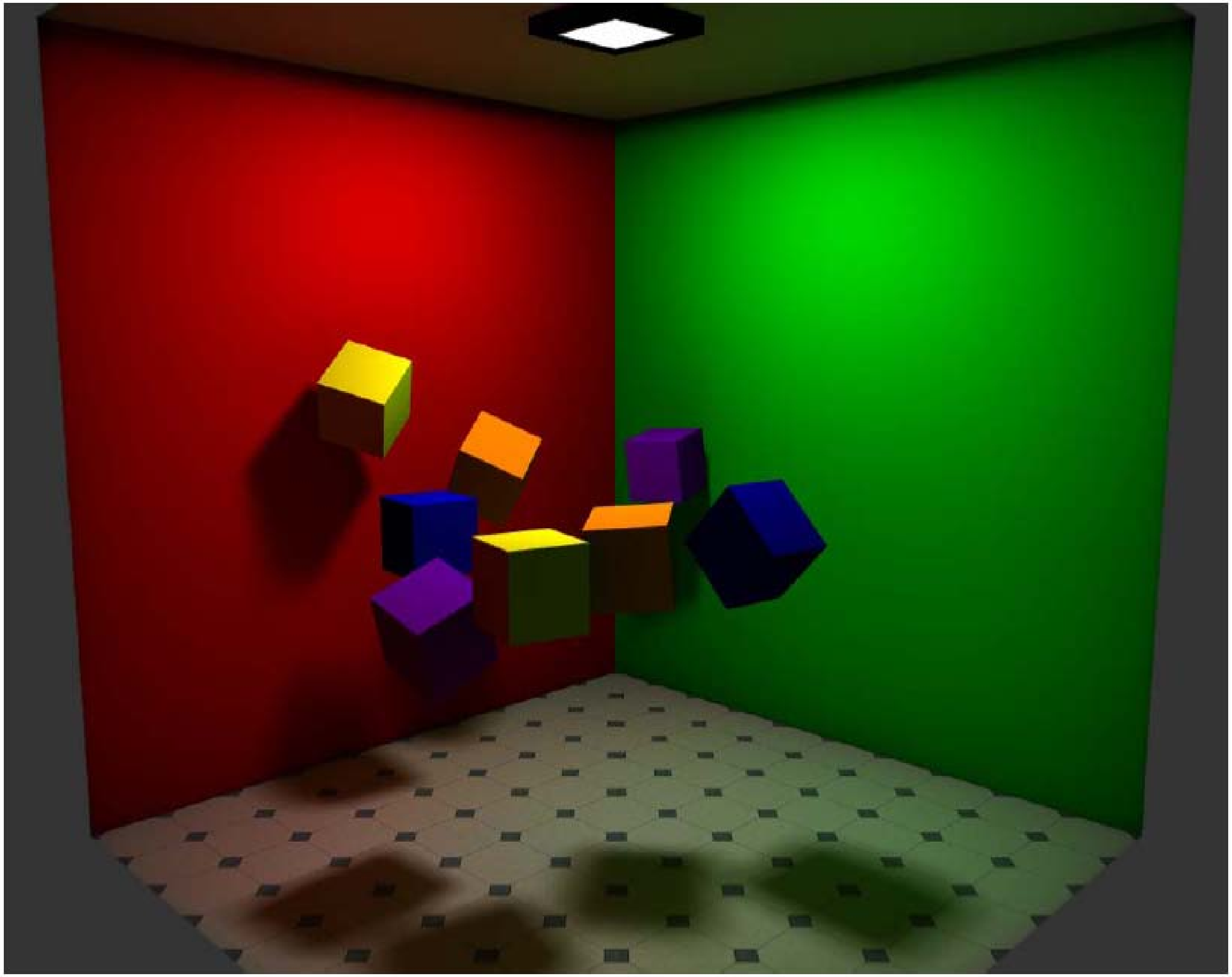
级数近似

$$1/(1-x) = 1 + x + x^2 + \ldots\ldots$$

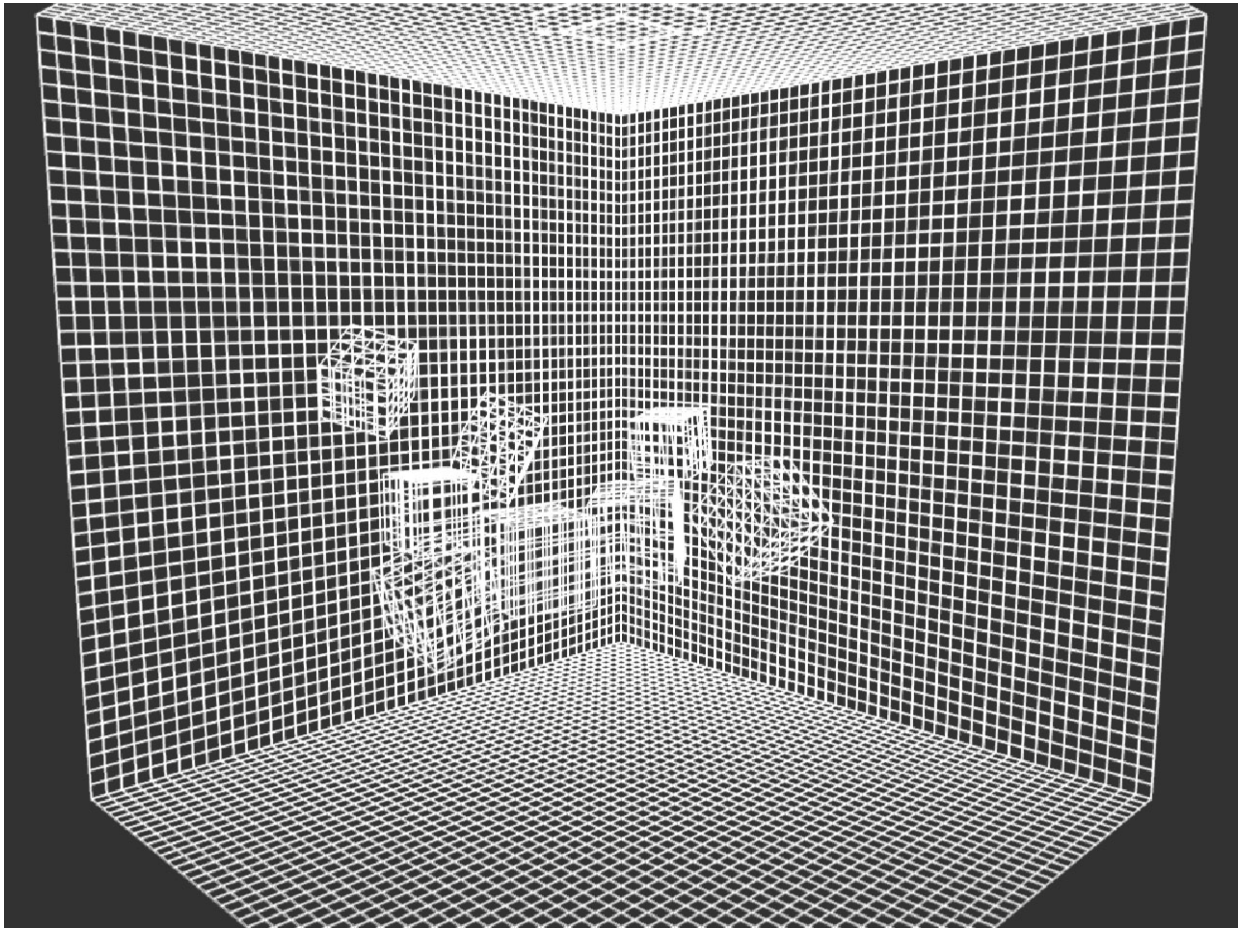
$$[\mathbf{I}-\mathbf{RF}]^{-1} = \mathbf{I} + \mathbf{RF} + (\mathbf{RF})^2 + \ldots$$

$$\mathbf{b} = [\mathbf{I}-\mathbf{RF}]^{-1}\mathbf{e} = \mathbf{e} + \mathbf{RFe} + (\mathbf{RF})^2\mathbf{e} + \ldots$$

输出的场景

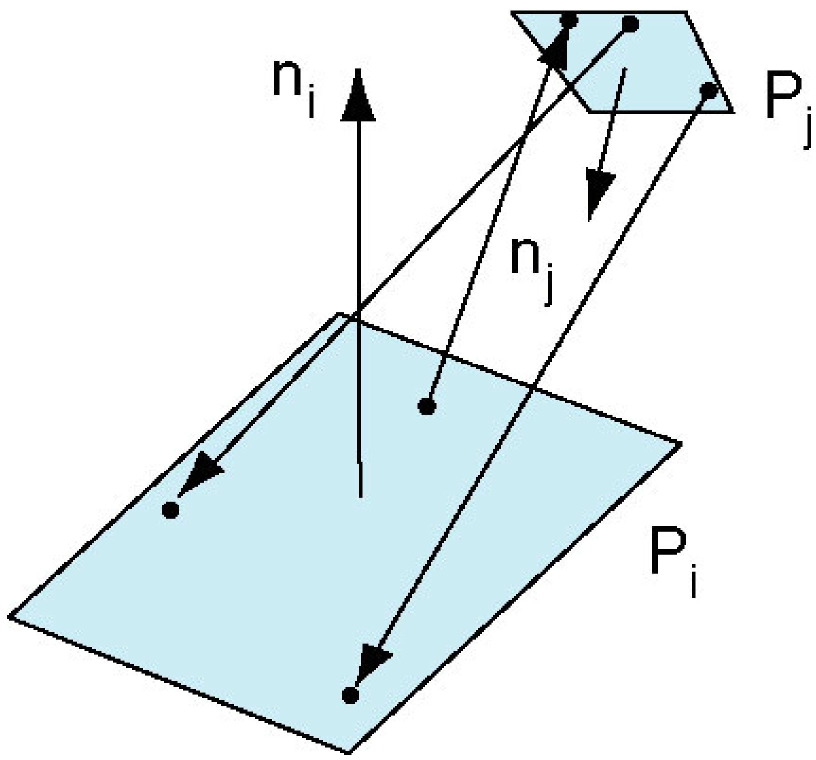


面片

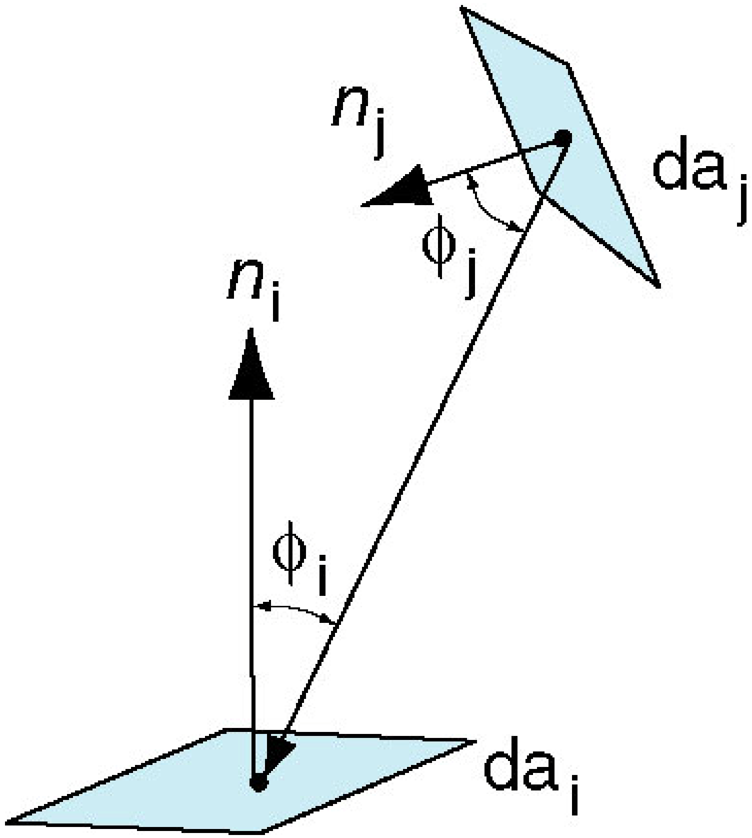


形状因子的计算

- 考虑两个平坦的曲面片



foreshortening



积分表示

$$f_{ij} = (1/a_i) \int_{ai}\int_{ai} (o_{ij} \cos\theta_i \cos\theta_j / \pi r^2)da_i\, da_j$$

遮挡

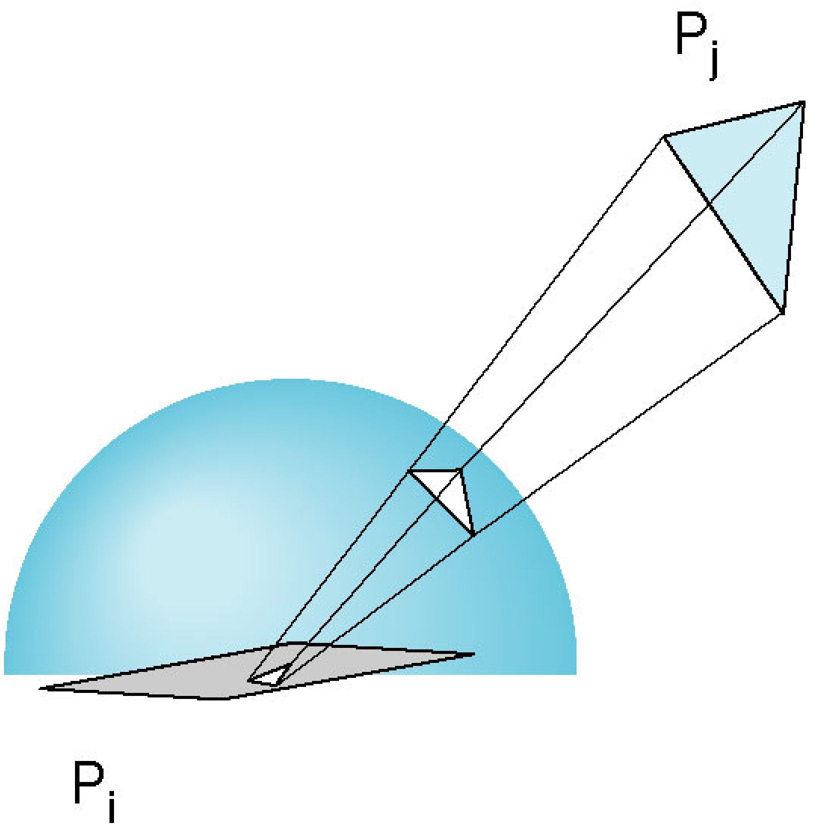
曲面片j的foreshortening

曲面片i的foreshortening

求解积分方程

- 只有极少的情形中积分可以有封闭的解
  - 遮挡使得解更复杂
- 实用的方法就是采用数值解法
- 可以采用类似于纹理映射中的两步方法
  - 半球
  - 半立方体

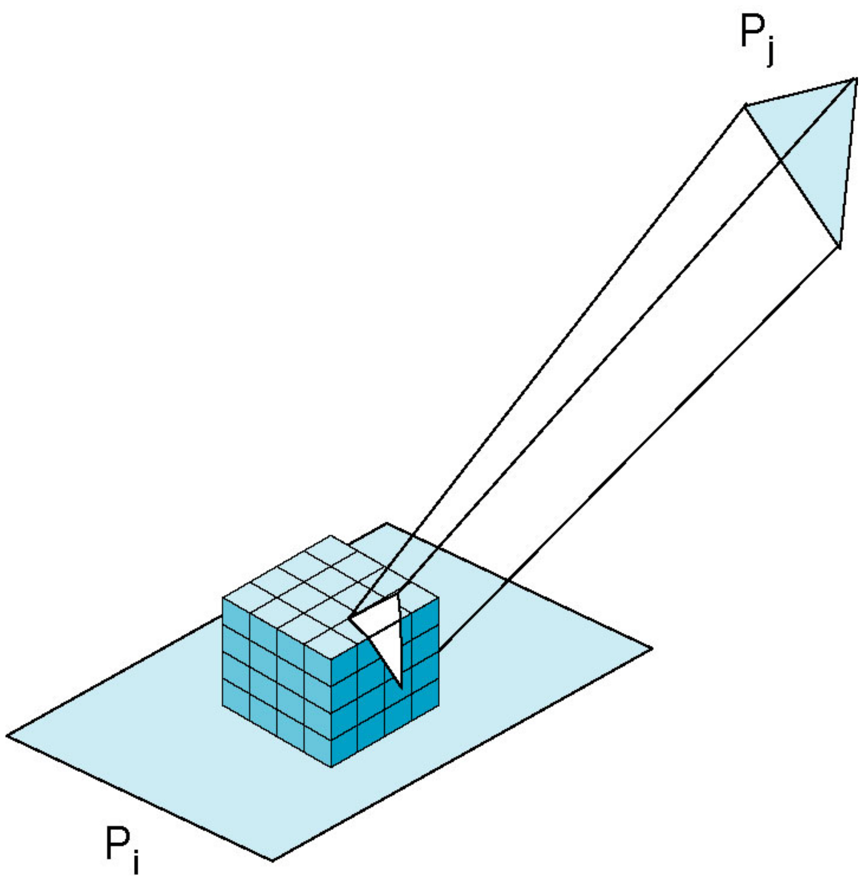
半球



半立方体

- 相对于半球，半立方体更容易应用
- 把每个侧面剖分为像素
- 计算每个像素的delta形状因子，然后把这些因子加在一起即可
- 为了得到delta形状因子，只需要从每个像素发射一条光线就可以得到

半立方体



求解积分方程

- 另一种近似解法
  - 充分利用图形系统快速计算简单场景的能力
  - 把点光源放在面片 P_j 上，然后使用现有的绘制器（譬如OpenGL）绘制场景，可以得到形状因子的近似值（类似某种意义上地测量）

辐射度方法的实现

- 辐射度绘制方法的主要步骤：
  - 将场景剖分为由许多面片组成的网格，计算其形状因子（计算最为密集的一步，可采取coarse-to-fine的策略）
  - 求解辐射度方程
  - 可将辐射度方程求解出来的值作为面片的漫反射系数，利用其他绘制算法绘制场景（譬如光线跟踪、OpenGL等）

Thanks for your attention!