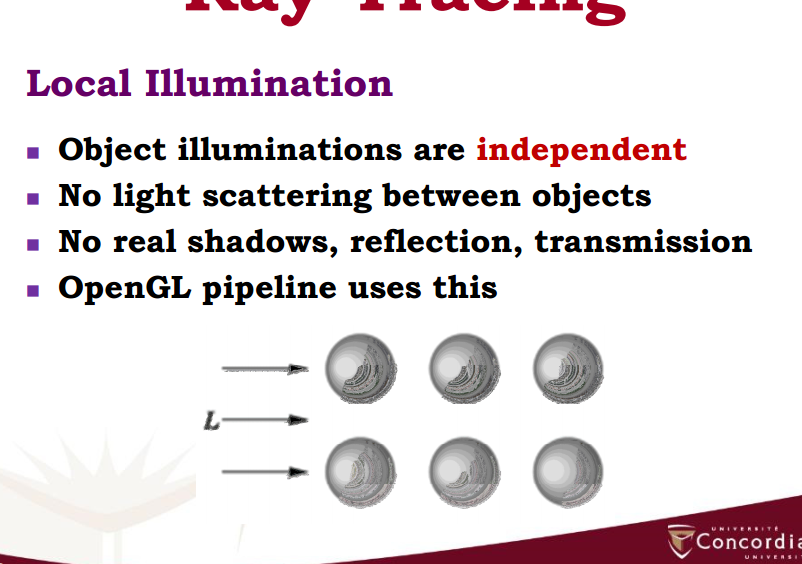
Local illumination

Object之间的照明是independent的

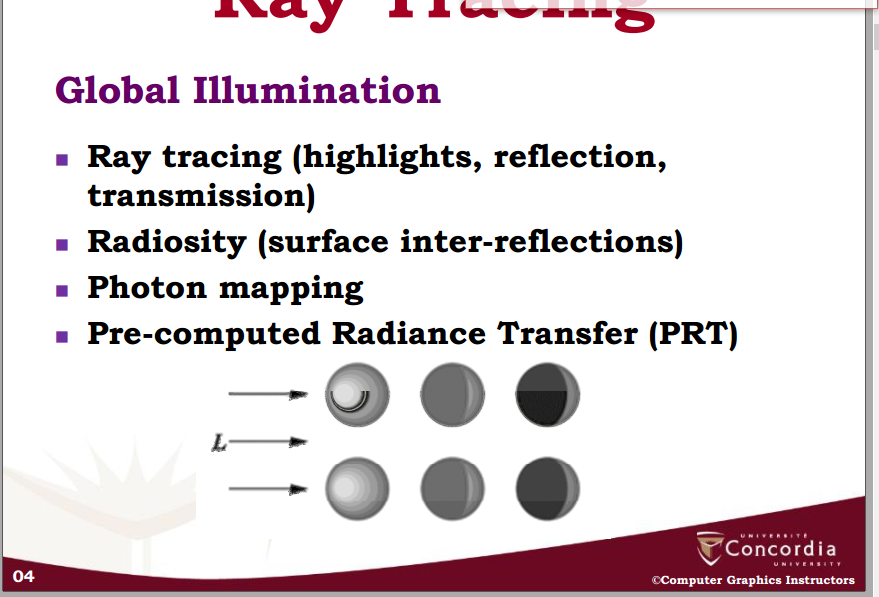
Object之间没有 light scattering 光散射

没有真实的影子，反射

Opengl通常使用local illumination



都是与LIGHT相关的，没有其他球的反射

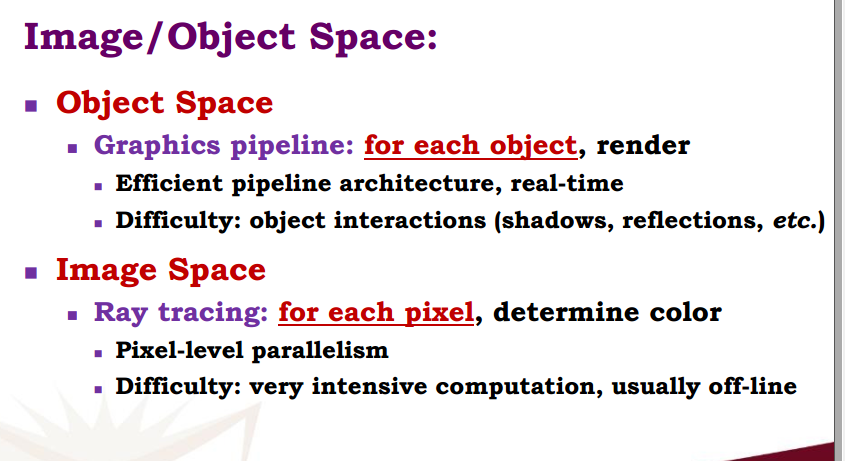
.global illumination:

会用到ray tracing

Radiosity (表面自身反射)

用photon mapping

使用PRT



Object space:

Graphics pipeline ,显卡pipeline， 对每一个object生效

有效的Pipeline结构，实时的

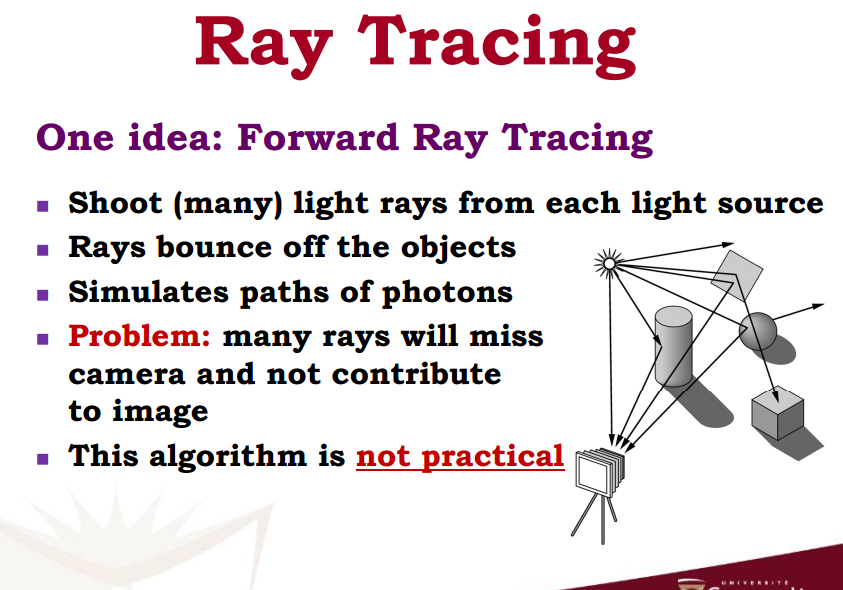
做不到object之间的交互（影子，反射）

Image Space

Ray tracing:针对每一个像素点，决定颜色

像素-level 平行 pixel-level parallelism

缺陷：大量密集计算，通常是off-line (离线状态)



Ray tracing

Forward ray tracing 正向光线追踪

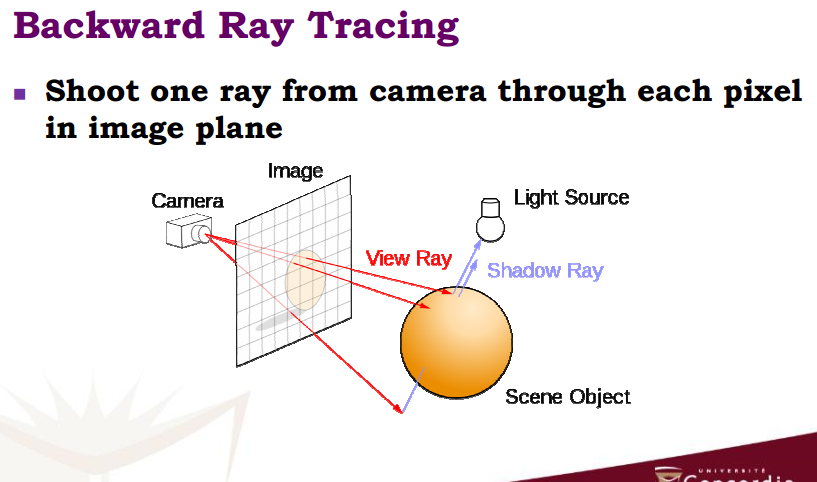
从每一个光源中都射出射线，这些光线在物体间弹跳，

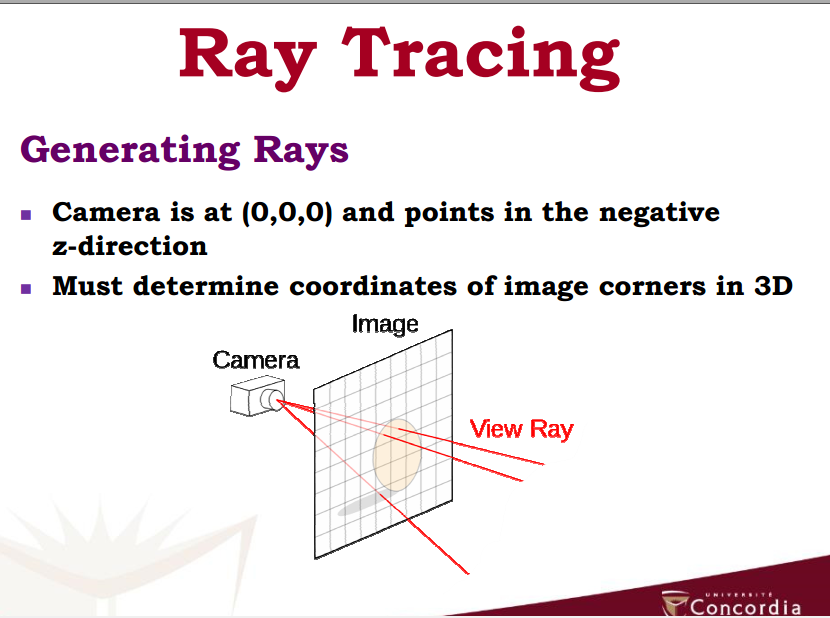
模拟光子的路径

问题：许多光线并不能射进camera,对生成图片没有效果，所以这个算法并不实际

Backward ray tracing

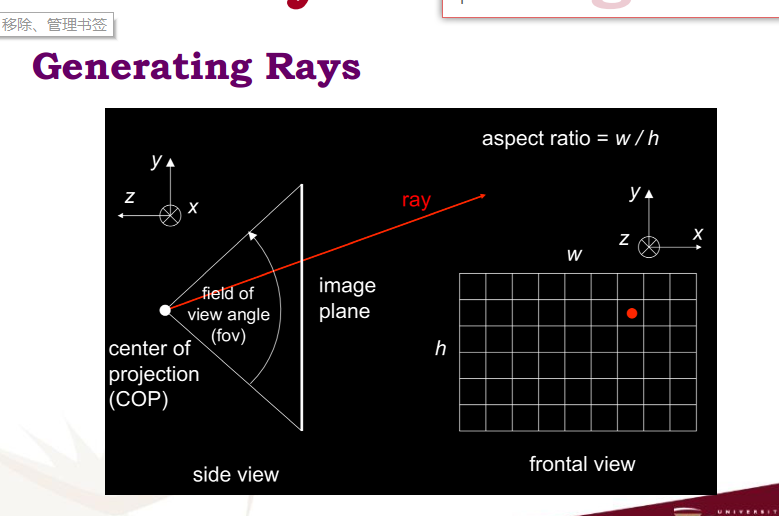
从camera 射出光线，通过一个image plane//生成图像的平面

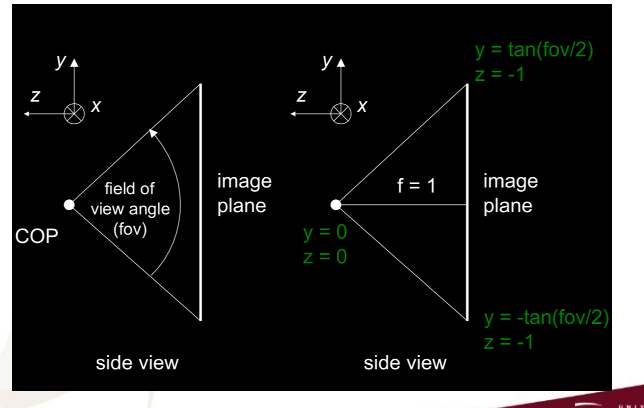


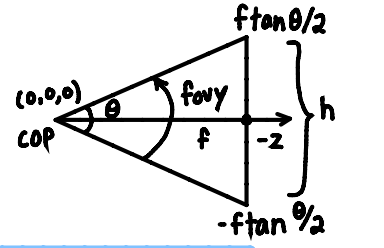


1.生成射线，camera位置在（0,0,0）,指向negative z

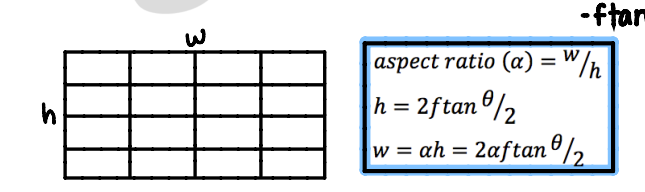
必须决定图像四个角的坐标







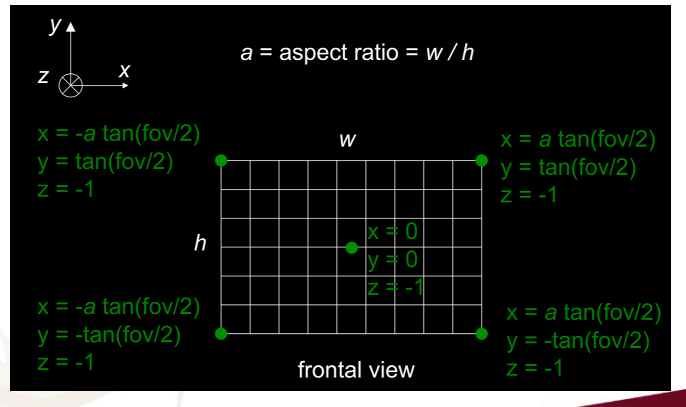
fov 是视角大小，000是一开始的camera位置，距离f=1，因为camera在000而且指向negative z，所以实际上我们这个平面就是z=-1, 上下y的范围就是

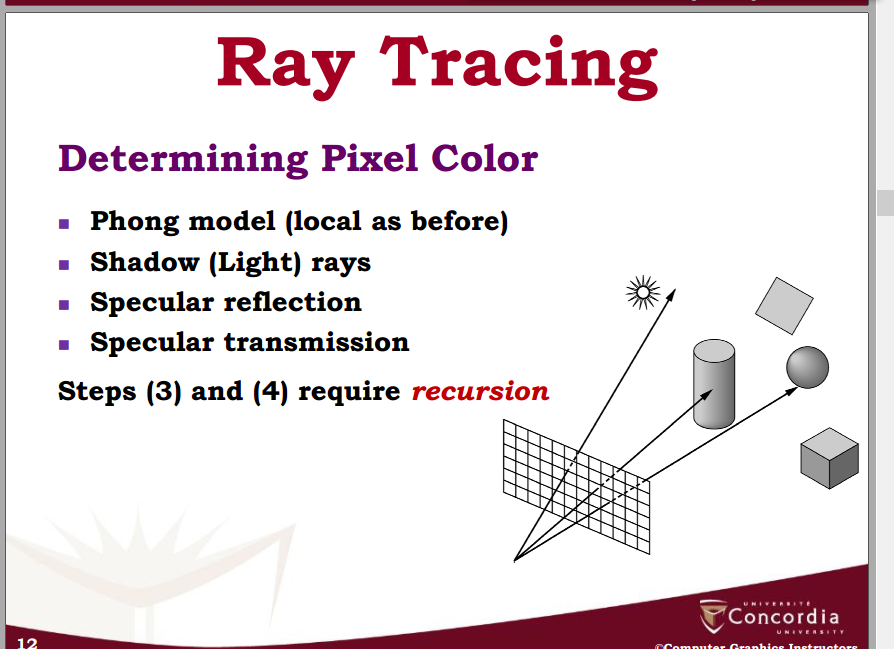


而aspect ratio 就是这个image的长✖高

因此我们可以知道知道四个尖角的端点，既x/y=a=w/h，因此w=ay=atanfov2

然后我们就能得到hw





决定像素格的颜色

1.用phone model（local）

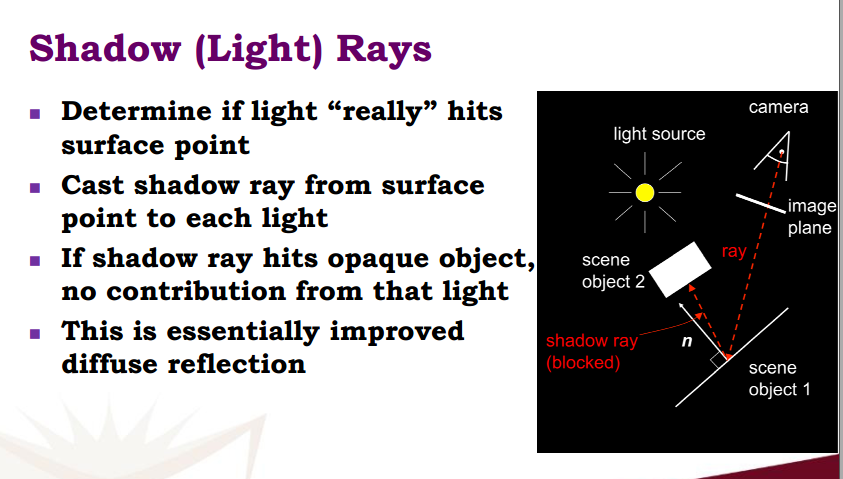
2.影子（光）的射线 shadow ray

3.Specular reflection 镜面散射

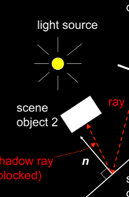
4..Specular transmission 镜面传输 三四步都需要recursion

Shadow (Light) Rays

这个ray只是单纯的camera通过image plane的射线一直碰到第一个scene object（背景这种），然后与光源相连

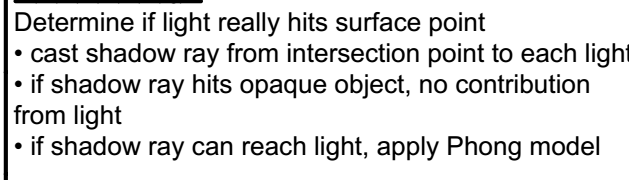


shadow ray的目的：决定光线是否真的中找到了物体表面的点上

1.将shadaow ray从交点intersection 与所有光源相连

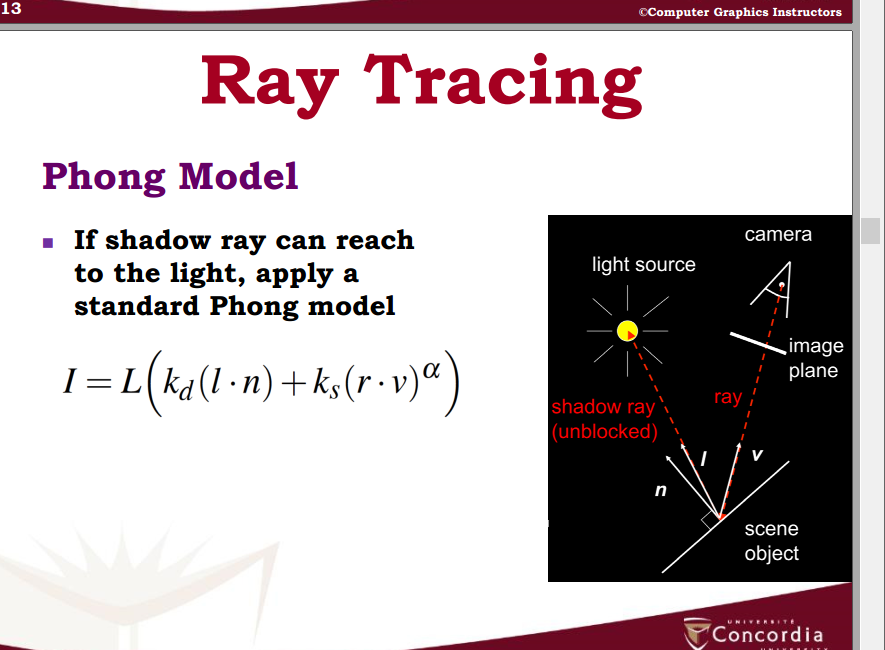
2.如果shadow ray遇到了不透明物体，那么光对他没有contribution

3,.如果shadow ray可以遇到light，先用phone model处理



从camera射出一条经过image plane的ray，并且最终指向光源，如果反弹一次后中间仍然有opaque object 不透明物体， 那么他就没有light效果

这增加了diffuse reflection， 扩散反射 //就是阴暗明亮，第一步ambient，第二步diffuse,第三步specular



Phong model

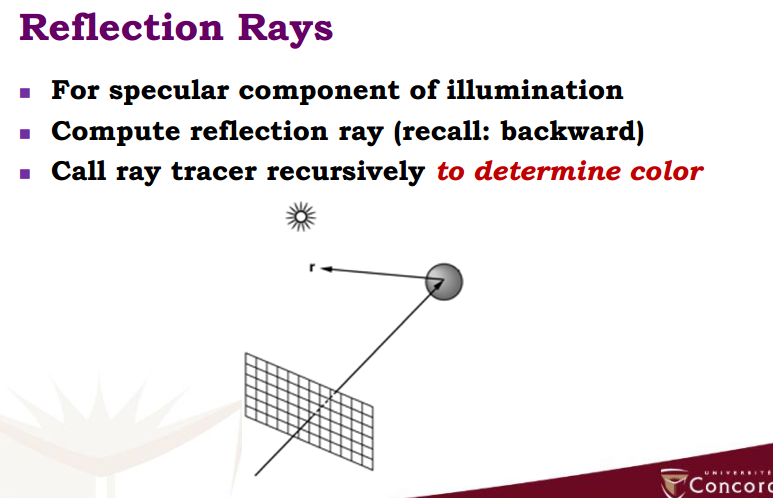
如果这个shadow ray可以正常触碰到光源，那么就是标准的phong model

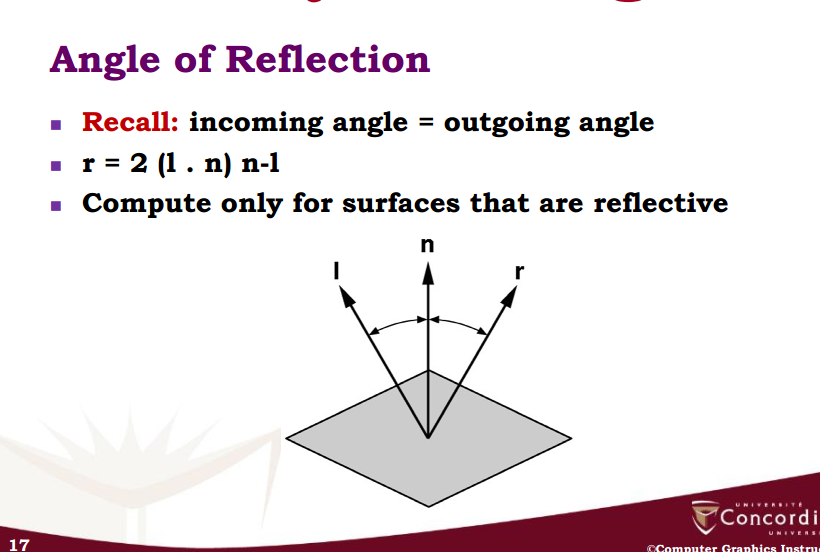
Reflection ray 反射光线

决定照明的specular(第三部分)

计算reflection ray

需要recursively重复call ray tracer来决定颜色





入射角等于反射角

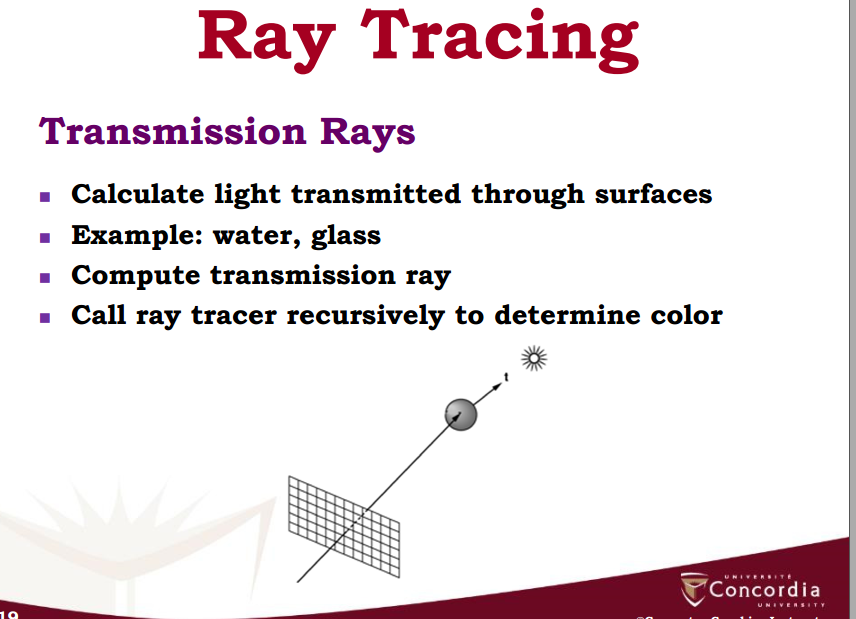


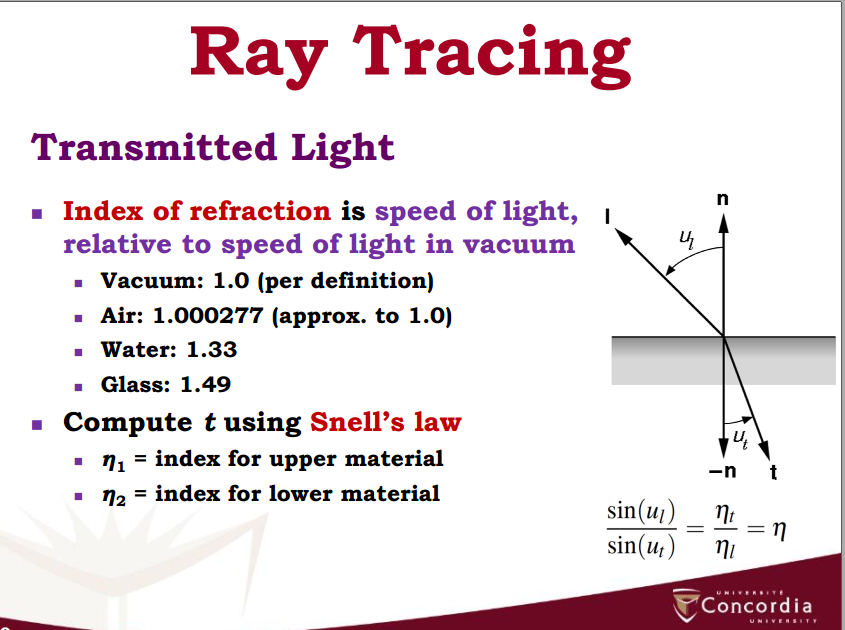
Transmission rays 传播射线 (遇到透明物体发生折射的光线)

计算透过表面的光线

例如 水，玻璃

也需要recursively call ray tracer

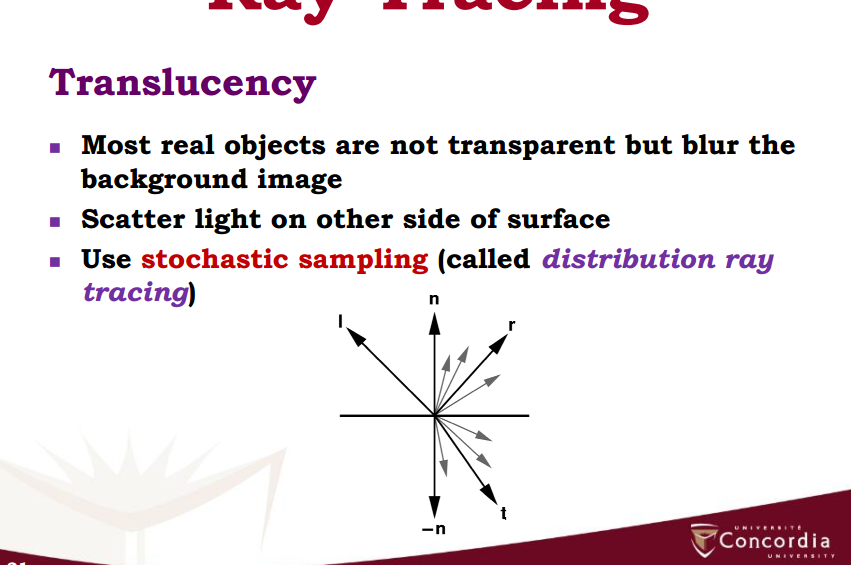




Index of refraction 折射率 光的速度

入射角度与出去角度的sin比值等于折射率的比值

也就是说通过入射角与两个折射率，能知道出去的角



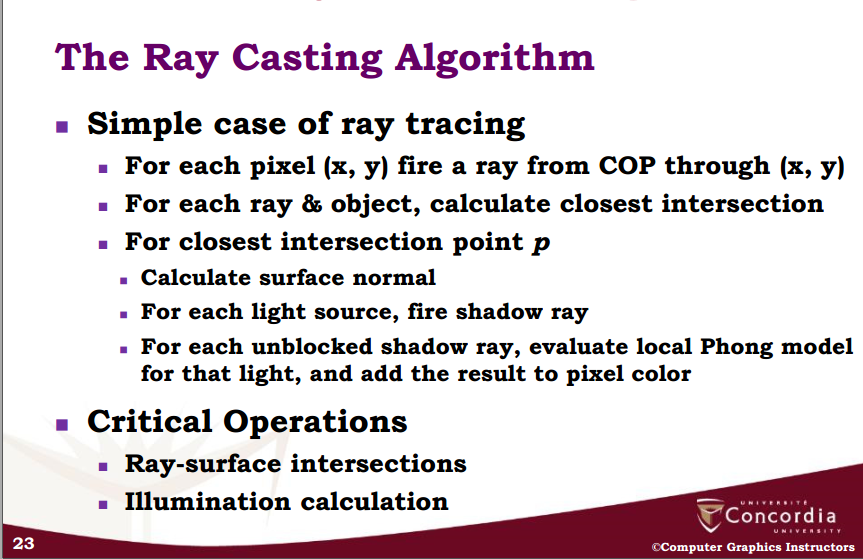
Translucency半透明

许多真实物体不是完全透明，而是浑浊

将光散射到scatter散射到表面的另一面

使用stochastic sampling随机抽样（叫做distribution ray tracing//分布式ray tracing）

Ray Casting Algorithm//ray casting算法



最简单的情况：

从camera的位置向image的每一个像素射出一条ray

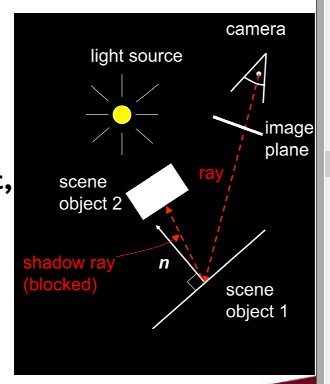
然后每一个射线与object，计算最近的交叉点（也就是camera遇到的第一个实体）

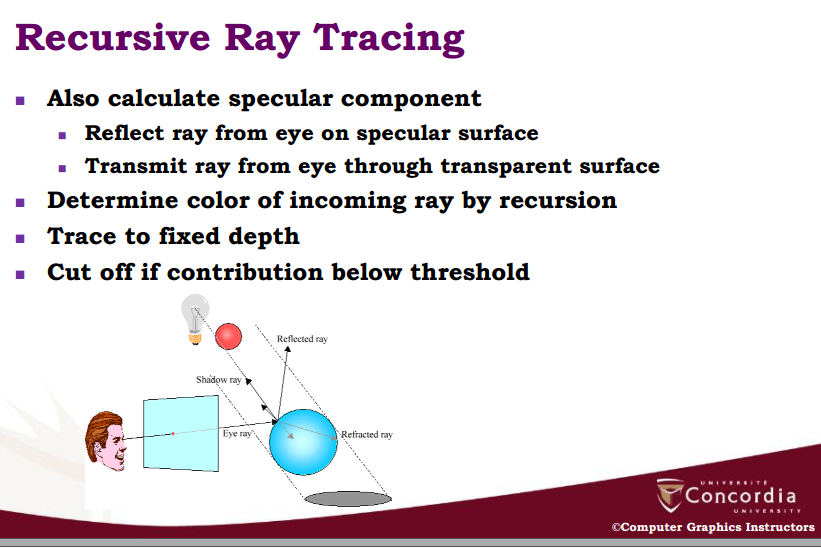
在这个最近交叉点P上

计算表面的normal垂直向量

对于每一个光源，与这个交点相连

如果中间没有再遇到第二个物体，就是local phone model，直接把这一部分像素作为phong model来处理，把结果加入image plane





Recursive ray tracing

也要关注specular的部分（第三步）

Eye ray的镜面反射 reflected ray

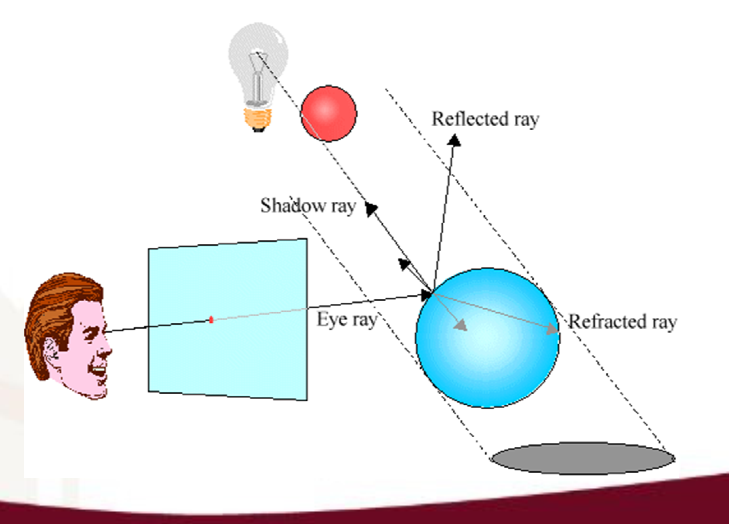
Eye ray 的折射，transparent surface

通过recursion决定最终实际射入眼睛的颜色

跟踪到固定深度

如果贡献低于临界值，就舍去

别管图，你就知道第三部是specular第四部是transparent，通过recursion就行了，如果低于临界值threshold就别管



Ray Tracing Assessment 评估

他是global illumination method

基于image

优势

更加真实的shadow, reflection, refraction

劣势‘

慢（一个一个像素画出来，而不是pipeline）

Aliasing混淆现象

物体间的diffuse reflections 漫反射 需要许多次弹跳