

真实感渲染略谈

中央电视台录制二部 连天红

渲染是三维动画至关重要的一步，渲染的效果直接影响到成品的质量。精致的模型、流畅的动作、逼真的材质、自然的灯光，这一切都需要经过最后的渲染才能栩栩如生地呈现在观众面前。而拙劣的渲染会使之前的所有努力都付之东流。

当前的动画渲染主要有两种发展方向，一种是追求真实的照片级图像质量的渲染（Photorealistic Rendering）；一种是追求特殊艺术效果的非真实渲染（NPR，Non-Photorealistic Rendering）。图 1-3 分别表现了卡通、彩色铅笔画、水墨画的渲染效果。



图 1



图 2

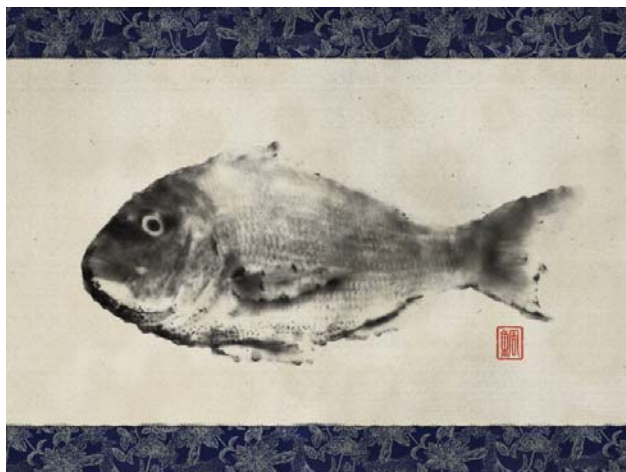


图 3

真实感渲染是目前大多数三维动画作品追求的效果，主要用于虚拟现实、影视特效、图形仿真、广告、电脑游戏等领域。真实性和实时性是它的研究要点。

所谓渲染，其实就是根据光学物理的有关定律，计算场景中所有物体可见表面上任一点投向观察者眼中的光亮度的大小和色彩。它可分为局部照明渲染和全局照明渲染。

局部照明 (Local illumination)

局部照明也称为直接照明 (direct illumination), 只计算光源发出的光直接照射在物体表面所产生的光照效果, 物体表面通常被假定为不透明, 且具有均匀的反射率。局部照明能表现由光源直接照射在漫射表面上形成的连续明暗色调、镜面上的高光以及由于物体相互遮挡而形成的阴影等。但它不考虑周围环境对当前物体表面的光照影响, 忽略了光在环境中各物体之间的传递, 因而在局部照明中, 直接光照不到的地方是黑暗的, 阴影的边缘也是硬的, 很难表现自然界复杂场景中的很多光照现象, 因而是片面的、不完全真实的。如图 4 所示。

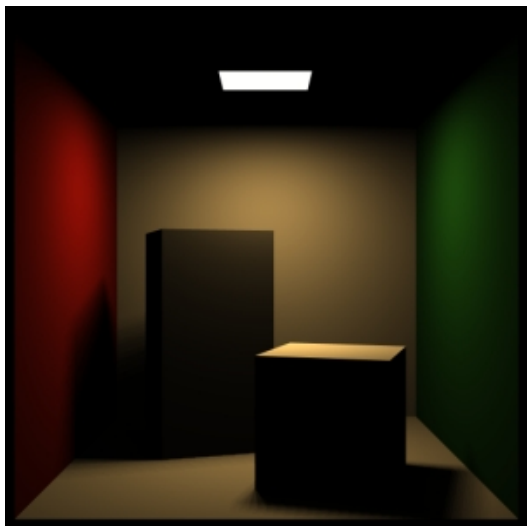


图 4

全局照明 (Global illumination)

如果想得到真实的照片级图像效果就必须采用全局照明的方法。全局照明除了考虑上述因素外，还要考虑场景中光线与物体之间，物体与物体之间相互影响而引起的所有间接照明 (Indirect Illumination) 情况。它可以模拟自然界中的各种光照现象，例如漫反射、镜面反射、折射、次表面光线散射 (Subsurface Light Scattering, 光线穿过半透明物体表面进入眼中)、颜色渗透 (Color bleeding, 带颜色的物体表面可以将颜色映到临近的物体表面) 和焦散 (Caustics, 光线经镜面反射或折射到一个漫射面而引起的聚焦现象) 等效果，还能够生成柔和的阴影。如图5所展示的是经过光线10次反弹后的全局照明效果，可以看到明显的颜色渗透现象（两个盒子各有一面被旁边的墙分别映上了红色和绿色）。

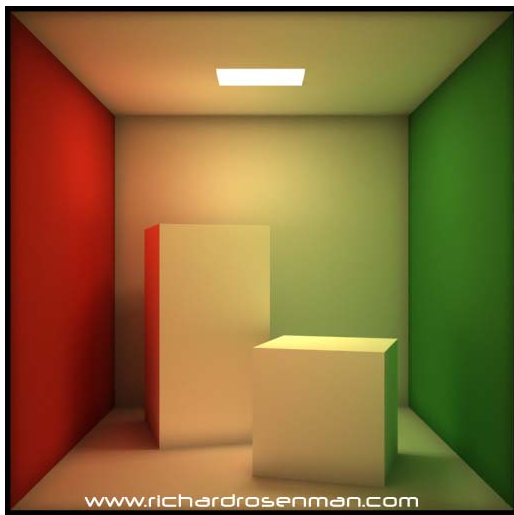


图5

光线跟踪 (Raytracing) 和光能辐射度 (radiosity) 算法是两种广为人知的实现全局照明的方法。这两种方法在执行过程中主要的区别是出发点不同：光线跟踪追随从观测者的视点出发返回到光源的光线，而光能辐射度则模拟从光源开始经过漫反射传播的光线效果。

众所周知，在自然界中，我们能够看到各种物体是由于光线照射到物体上，有部分光线经过反射或折射进入到我们眼中。其传播过程是光源 \rightarrow 物体 (\rightarrow 物体 $\cdots \rightarrow$ 物体) \rightarrow 人眼。假设光线都是从光源 L 出发，终止于眼睛 E ，而且只与两种物体表面发生相互作用：纯镜面用 S 表示，纯漫射面用 D 表示。则光线的传播路径可以用表达式 $L (D|S)^* E$ 来表示。 $*$ 表示光线经一个或多个镜面/漫射面后进入眼睛。如图 6-8 所示，展现了光线的各种传播路径：

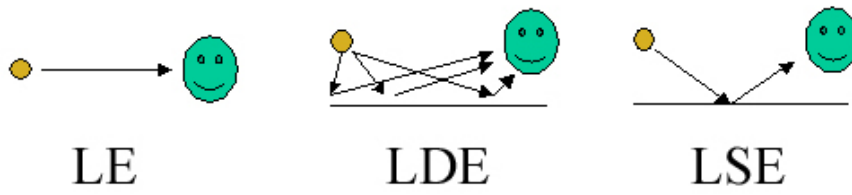


图 6

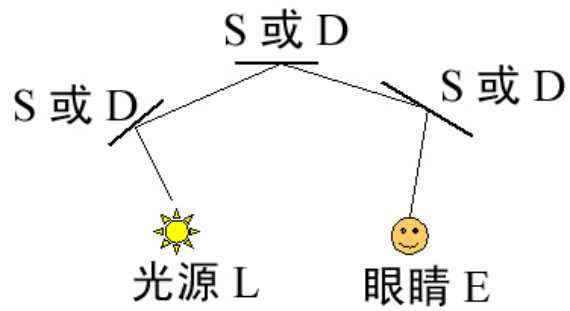


图 7

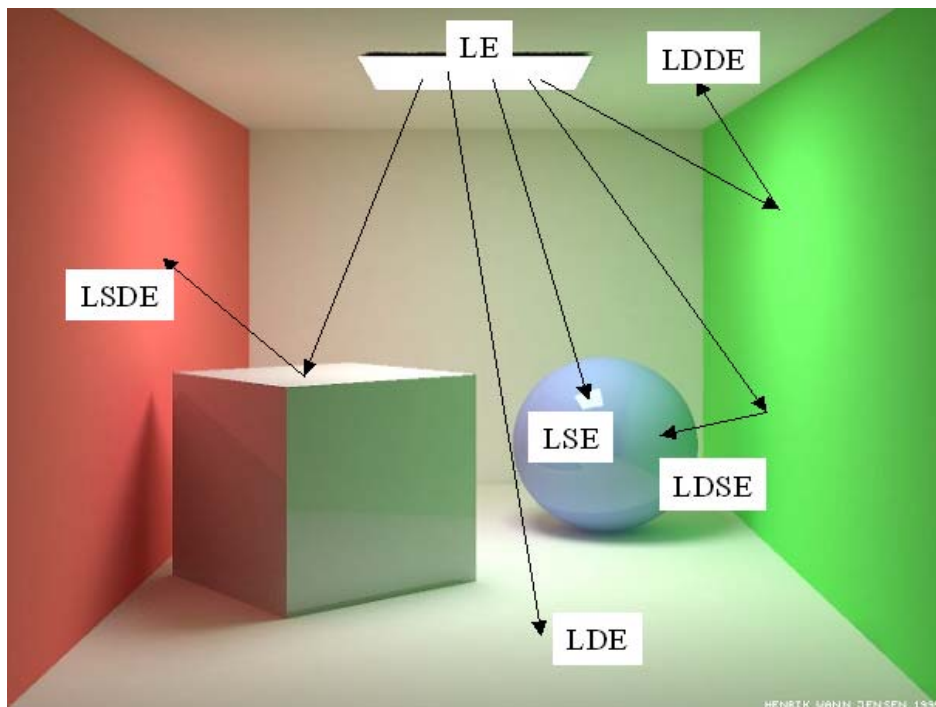


图 8

光线追踪:

光线在场景中物体之间的传播可以概括为四种方式:

- (1) 镜面→镜面;
- (2) 漫射面→镜面;

(3) 漫射面→漫射面；

(4) 镜面→漫射面。

光线跟踪算法可以求解理想状况下镜面→镜面、漫射面→镜面的效果。这里所说的理想情况是指它假设光线是一根没有大小和长度的直线；镜面是完全光滑的，与物体相交的光线遵循镜面反射和规则透射规律。

由于从光源发出的光线有无穷多条，而只有少数经由场景中物体表面之间的反射和透射后到达观察者的眼中，如果从光源开始跟踪，不仅做了很多无用功，而且计算量极其庞大。因此，标准光线跟踪算法采用逆向跟踪技术来完成整个绘制过程。如图 9 所示。

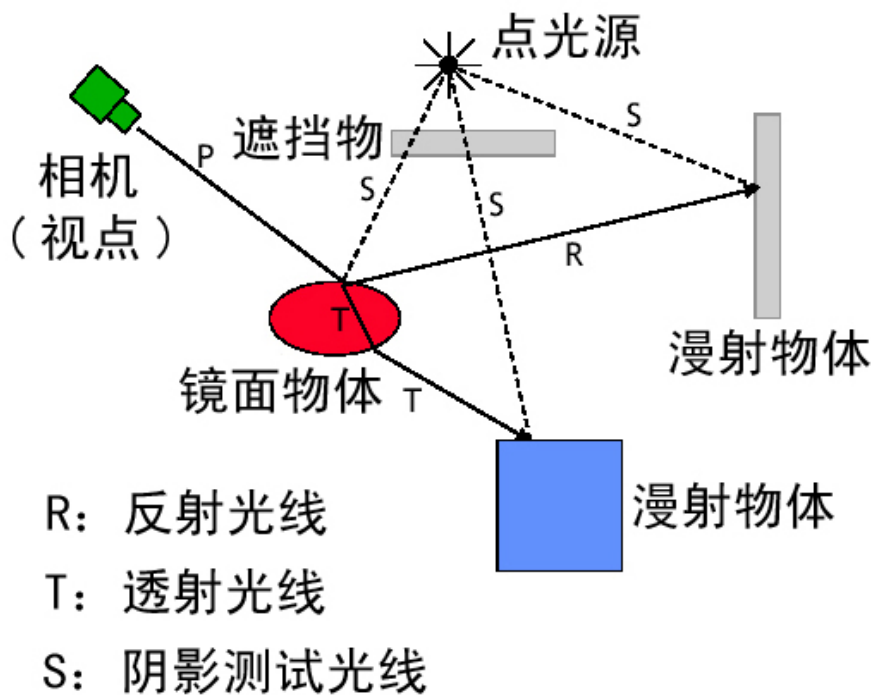


图 9

其基本工作流程如下：

从视点出发，通过图像平面上每一像素中心向场景发出一条光线。如图 10 所示：

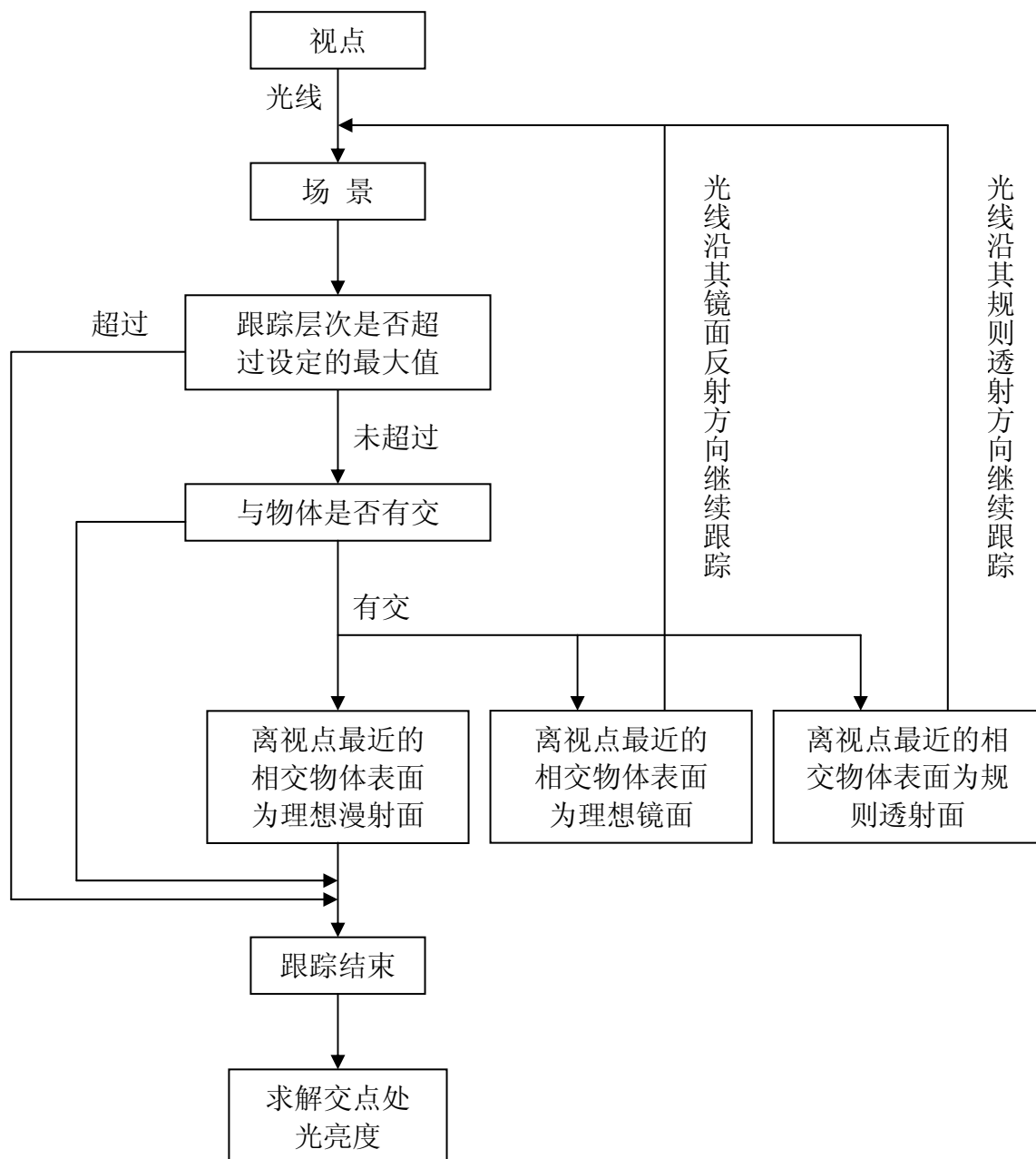


图 10

交点处光亮度由三部分组成：

- (1) 由光源直接照射而引起的光亮度；
- (2) 来自环境中其它物体的入射光在表面产生的镜面反射光亮度；
- (3) 来自环境中其它物体的入射光在表面产生的规则透射光亮度。

对于阴影的计算，算法首先确定当前交点是否位于该光源的阴影区域内，即从当前交点向该光源投射一条阴影测试光线，若测试光线在到达光源之前与场景中的不透明物体有交，则说明当前交点位于该光源的阴影内。此时当前交点由于没受到该光源的直接照射，其产生

的局部光亮度可取为零。若与测试光线相交的物体均为透明体，则从光源发射的光能在穿过这些物体时受到一定程度的衰减，从而使该光源在当前交点处产生的局部光亮度减弱。图11表现了标准光线跟踪算法渲染的反射、折射和阴影效果。

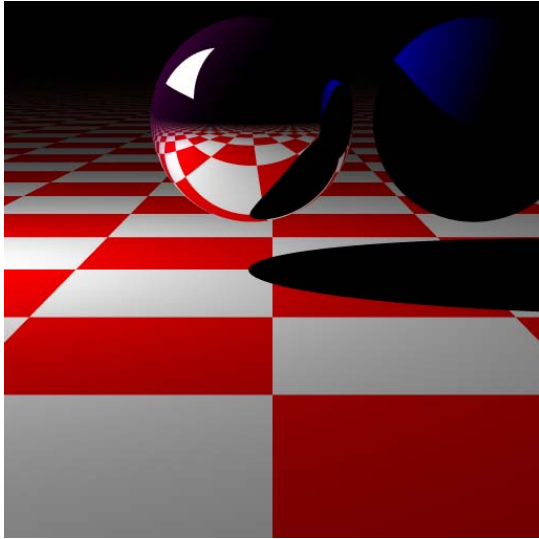


图11

对于普通PAL制 720×576 的视频分辨率，需要计算40多万个点的光亮度。

光线跟踪能够成功模拟物体间的理想镜面反射和规则透射及阴影效果，但也有很大缺陷：由于需跟踪每一条从视点发出的光线，因而涉及到大量的光线与景物的求交测试计算，计算量十分庞大；而它本质上是一个离散采样算法，各个屏幕像素的亮度是分别计算的，绘制的画面有时会出现严重的走样现象；对于物体间多重漫反射、颜色渗透、焦散和柔和阴影等现象无能为力。

为了解决这些缺陷，光线跟踪衍生出很多改进的算法，如用于提高跟踪效率的层次包围盒技术，三维DDA (Digital Differential Analyzer) 算法，空间八叉树剖分技术，二叉空间剖分 (BSP, Binary Space Partition) 技术等；用于克服图像走样的圆锥光线跟踪，像素细分技术，非均匀超级采样技术等。其中比较出名的是分布式光线跟踪 (Distribution Raytracing) 算法和双向光线跟踪 (Bi-directional Raytracing) 算法。

分布式光线跟踪算法在处理每一被跟踪光线时，在光线与物体表面的交点处不仅朝其镜面反射方向和规则透射方向发射光线，而且依据物体表面的光照性质朝其镜面反射方向和规则透射方向附近的立体角内亦发射采样光线。因此分布式光线跟踪算法实际上是一个使用蒙特卡罗积分采样的过程，有时也称其为蒙特卡罗光线跟踪 (Monte-Carlo Raytracing)。该算法有效地避免了传统光线跟踪算法由于点采样引起的图形走样现象，真实地模拟了景深、运动模糊和半影 (柔和阴影) 等效果，大大提高了光线跟踪算法所生成图形的真实性。图12、图13为采用分布式光线跟踪算法渲染的半影和运动模糊效果。

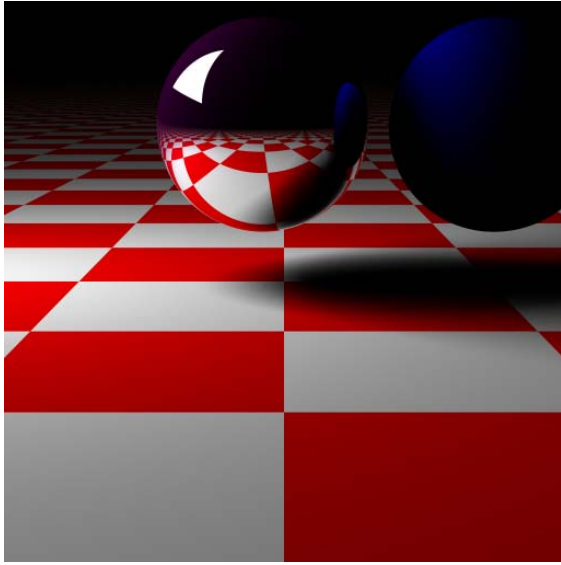


图12

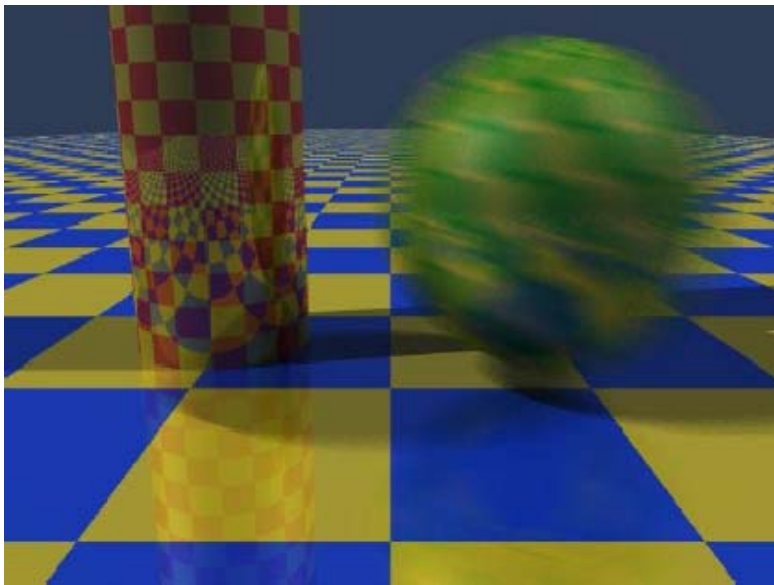


图13

但是采用传统的逆向光线跟踪算法始终无法模拟多重漫反射情况，即光线从镜面到漫射面或从漫射面到漫射面的传递。而且逆向光线跟踪与视点紧密相关，一旦改变视点位置，就要重新进行跟踪。采用双向光线跟踪技术可以模拟光线从镜面到漫射面的传递。所谓双向光线跟踪技术，即同时从光源和观察者出发，对光线进行正向和逆向跟踪。这里的逆向跟踪实际上就是传统的光线跟踪，而从光源出发的正向光线跟踪则用来计算来自环境的镜面反射光和规则透射光对漫射表面光亮度的贡献。显然，与逆向光线跟踪相反，正向光线跟踪是一个与视点和视线方向无关的过程。图14是采用标准光线跟踪算法渲染的图像，图15是采用双向光线跟踪算法渲染的图像，可以看到很明显的焦散现象。

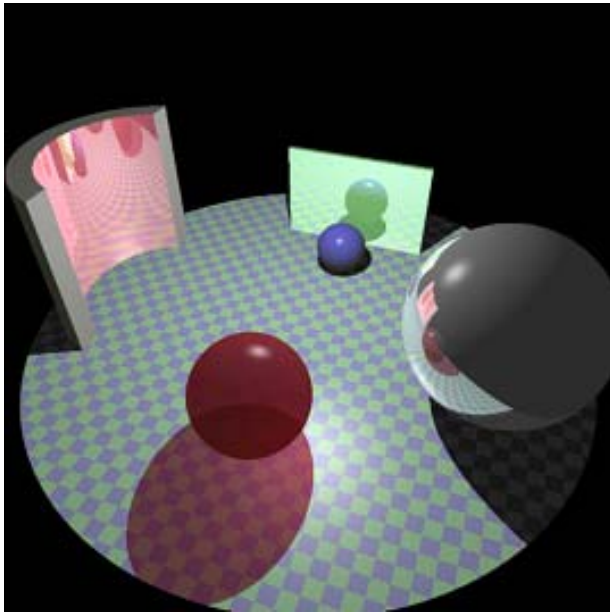


图14

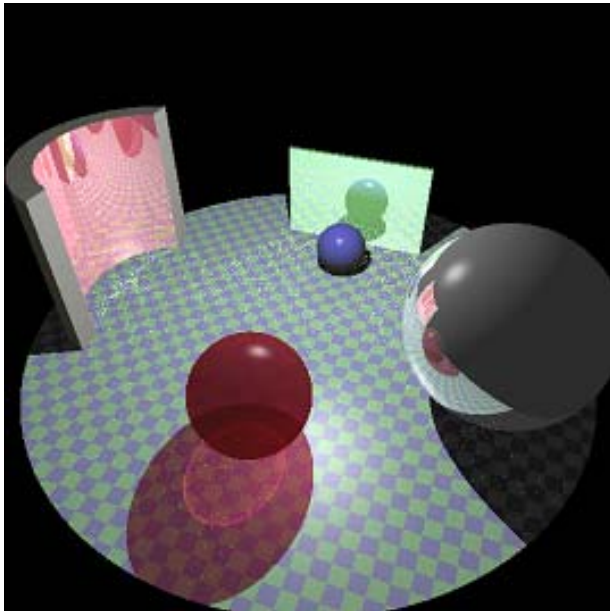


图15

光能辐射度：

光能辐射度方法基于物理学热辐射工程中的能量传递和守恒理论，即封闭环境中的能量经多重反射以后，最终会达到一种平衡状态。这种能量平衡状态可以用一系统方程来定量表达。由于方程非常复杂，无法精确求解物体表面各点处的辐射度值，可以对其适当简化。假设场景中的各物体表面被剖分成一系列互不重叠的小面片（patch），且各面片上的辐射度值和漫反射系数均为常数。如图 16 所示。



图 16

在理想状况下（1. 封闭环境；2. 漫反射是向四周均匀反射光线），各面片的辐射度方程为：

$$B_i = E_i + \rho_i \sum_{j=1}^N B_j F_{ij}, \quad i = 1, 2, \dots, N$$

B_i 、 B_j ：第 i 、第 j 个面片的辐射度

E_i ：第 i 个面片自身的能量（辐射度）

ρ_i ：第 i 个面片的漫反射系数

F_{ij} ：面片 i 到面片 j 的形状因子，表示面片 i 辐射到面片 j 的光能占面片 i 向四周辐射总光能的比例。由面片 i 、 j 的形状，它们的距离及方向决定

此方程的含义为每个面片向周围环境辐射的能量由它自身具有的辐射光能和它接受来自环境中其它面片向该点辐射光能后产生的反射光能组成。与以往光照模型和绘制算法不同，辐射度方法是一种整体求解技术。求得系统方程的解，就知道了每个面片的辐射度分布，根据辐射度与光亮度的关系，即可选择视点和视线方向绘制整个场景。由于辐射度的求解与视点位置无关，当视点和视线方向改变时，不用重新计算各面片上的辐射度值，只需重新绘制场景即可。因此，这种方法特别适合虚拟现实中的互动式漫游。图 17 是采用辐射度方法渲染的场景：



图 17

辐射度方法的工作流程如图 18 所示：

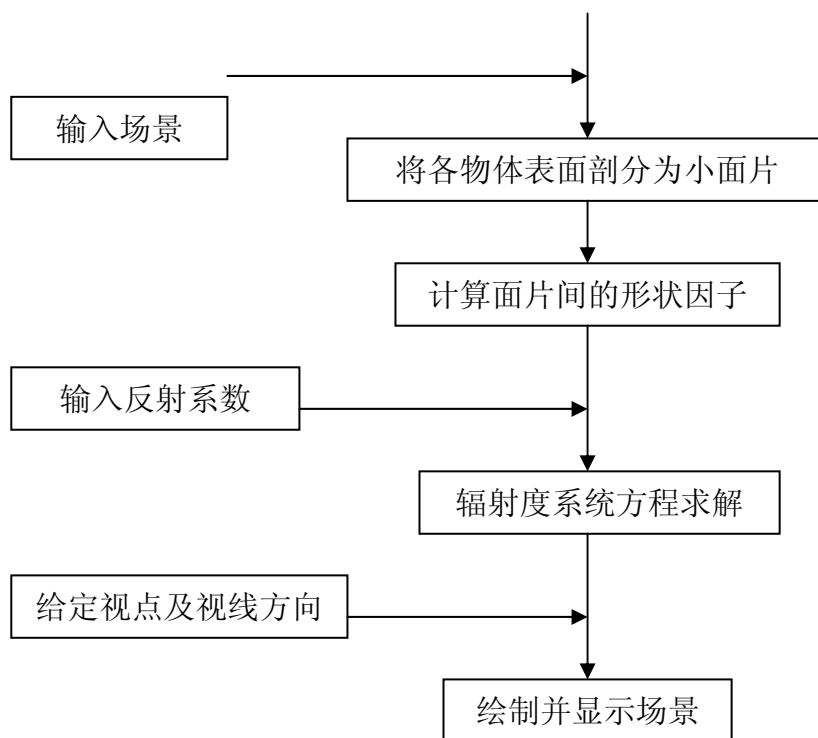


图18

从图中可以看出，辐射度方法关键要解决3个问题：物体表面的剖分；形状因子的计算；辐射度方程的求解。其中形状因子的计算量占到整个计算量的90%左右。

剖分是将物体表面分割成一系列简单形状面片，如长方形和三角形，每一面片均假设为

具有均匀辐射度。剖分所生成面片的数量和形状不仅严重影响辐射度方法的效率，而且也会影响面片间形状因子的计算精度。应尽量避免将物体表面分割成狭长的面片形状。在采样点(即面片)个数一定的情况下，一个好的剖分算法应使这些采样点(面片)均匀地分布在各物体表面上。

对于形状因子的计算，可采用半立方体算法、半球面分割技术、单平面投影算法、蒙特卡罗积分方法等。

求解辐射度方程的算法主要有迭代法和逐步求精法。与线性方程组的直接求解法不同，迭代法首先由用户给出线性方程组解的一个初始估计，然后在此基础上逐步进行迭代，使得每一步得到的近似解均比前一步的近似解更好地逼近线性方程组的解。若迭代步骤是收敛的，经过多次迭代后的近似解与线性方程组的真解之差趋于0时，迭代结束。使用迭代法只有等到迭代完毕时才可根据给定的视点来进行场景绘制，其中没有任何求解信息的反馈，给交互设计带来了很大不便。基于辐射源传播的逐步求精算法是对一种迭代算法Southwell算法进行了改进，使得在每一迭代过程中，可立即显示基于当前近似解的场景画面。逐步求精算法每次在场景中选取具有最大待辐射的辐射量的面片，并将它所积聚的光能向环境中辐射出去，从而使得各表面的待辐射能量得以更新。

近年来采用函数逼近的有限元辐射度方法引起了人们的广泛关注。它的核心是将一复杂的函数用一系列简单基函数的线性组合来近似表达，所需内存远远小于传统辐射度算法，为复杂场景的模拟提供了可能。

标准辐射度方法只考虑了从漫射面到漫射面的光能传递形式，而标准光线跟踪算法则考虑了镜面到镜面及漫射面到镜面的两种光能传递形式。两步算法(two-pass)结合了两种方法的优点，第一步采用拓广的辐射度方法计算出环境中正确的漫射和镜面光能分布，第二步采用光线跟踪算法来绘制场景。但是对于复杂场景计算量很大。

光子映射 (Photon Mapping)

光子映射 (Photon Mapping) 是当前模拟全局照明最快的算法之一，是由 Henrik Wann Jensen 在 1994 年提出并发展起来的。光子映射技术与以往的光照绘制方法不同，它是从光的物理属性出发，研究光子在场景中的传输，使得对于光照的模拟真正达到了从自然属性的角度来考虑。这种方法能够更有效的模拟一些特殊的光学现象和效果，例如，焦散、颜色渗透，半影、烟雾，次表面光线散射，夜空，运动模糊等。如图 19-23 所示：



图 19

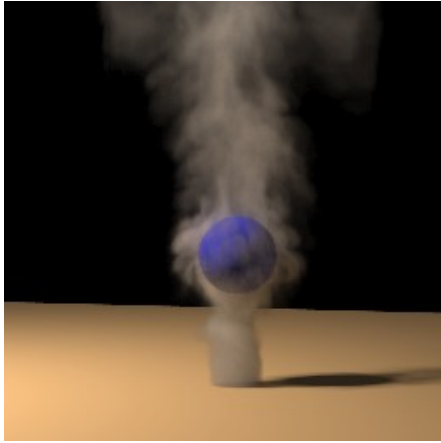


图 20



图 21



图 22

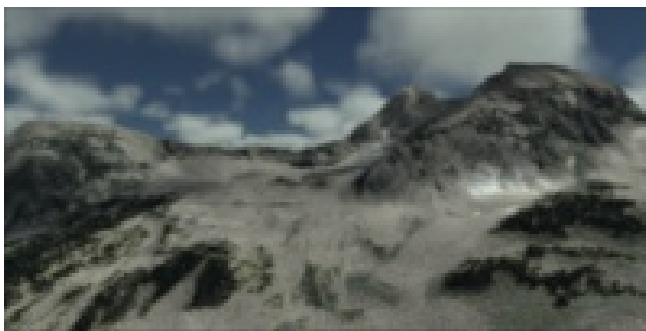


图 23

光子映射是光线跟踪算法的延伸，它结合了双向光线跟踪和蒙特卡罗方法的优势，主要通过两个步骤实现：

1、 构建光子图：

通过从光源向场景发射大量光子（能量包）来构建光子图。使用类似于路径追踪（Light Tracing）的方法跟踪穿过场景的每个光子。光子与物体表面碰撞后会产生三种情况：被反射、透射或吸收。使用俄罗斯轮盘赌的方法来决定是哪种情况发生。被反射光子运动的新方向由物体表面的双向反射分布函数（BRDF）来计算出。当光子与漫反射表面碰撞时被记录下来，碰撞点的位置、光子的能量、入射的方向等都被存储在光子图中——这是一种被称为平衡kd树的简洁有效的数据结构，这种存储结构与物体的几何属性无关，使用它可以提高渲染效率并减少内存的占用。

建立的光子图分为两种：焦散光子图（caustics photon map）和全局光子图（global photon map）。焦散光子图只存储与产生焦散效果有关的光子，当射向镜面物体的光子经镜面反射后与漫射表面碰撞时被记录下来，这时得到的就是焦散光子图。焦散的渲染是通过基于焦散光子图的辐射度估算来实现的，这个过程可以直接看到，因此需要高密度的光子。

全局光子图通过向所有物体发射光子来建立，用于对场景中光通量的粗略估计。它不能

被直接看到，因此不需要像焦散光子图那样精确。全局光子图中的光子可分为直接照明光子、间接照明光子和阴影光子。如图24所示：通过追踪从光源开始穿越整个场景的光线来产生阴影光子。在第一个交点处存储一个普通的光子，在下个交点处存储阴影光子。这些阴影光子用于渲染时减少阴影光线的数目，优化阴影的渲染。

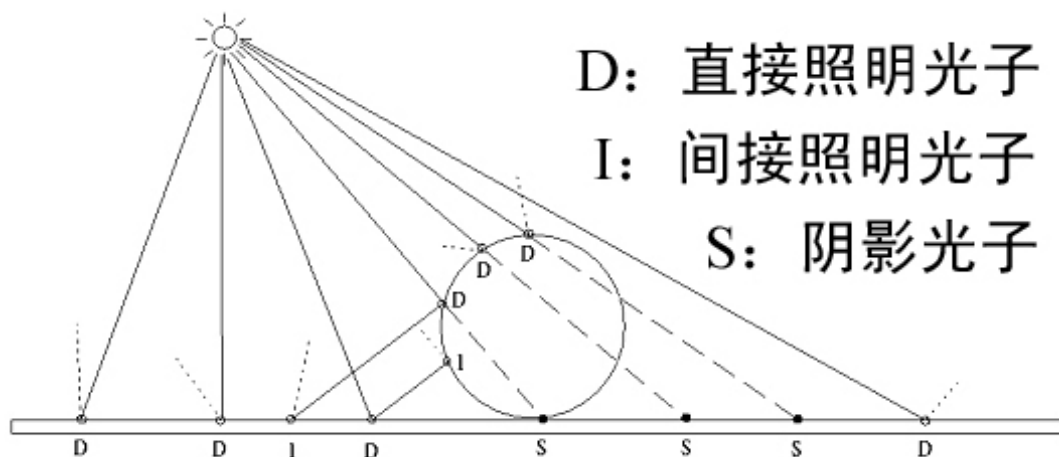


图24

有两个单独的光子图可以加快渲染速度，减少内存需求并提高渲染精确度。与焦散有关的光子具有高密度低能量，普通的光子低密度高能量。焦散光子图只包含与焦散有关的光子，因此焦散效果的渲染速度较快。而全局光子图拥有较少的光子数，且不包含焦散光子，因此这些光子的能量级别相似，定位速度很快。这样做显著提高了辐射度估算的精确度。

2、渲染：

一旦建立了光子图，有几种方法可以看到它。光子图是与视点无关的，因此从场景中的任何视点都可以使用光子图。能够看到光子图的一个非常快的方法是计算场景中每个顶点的发光值然后通过大部分三维硬件都支持的顶点照明从任何给定的视点渲染场景。另一个方法是直接看到光子图。方法很简单，对每个像素追踪一条穿过场景的光线，计算与表面碰撞点的辐射度估算。最后的图像采用蒙特卡罗光线跟踪来渲染，这种方法是通过很多采样点的估算进行平均来计算像素辐射度。每个采样都是追踪一条从视点出发穿过像素射向场景的光线。每条光线返回的辐射度，相当于光线从与之相交的第一个物体表面交点处射出时的辐射度。这个辐射度由四部分组成：直接照明、镜面反射、焦散、柔和间接照明。

辐射度估算可采用精确估算和近似估算。当物体表面能被眼睛直接看到或有少许的镜面反射时使用精确估算。如果光线原点与物体交点之间的距离小于一个阈值时也使用精确计算；如果从视点出发的光线与物体表面相交后被漫反射或者光线很弱对像素辐射度的贡献非常小

时则使用近似估算。

1. 直接照明：

计算光源直接照明产生的辐射度。通常是采用向所有光源发射阴影光线进行可见性测试来计算的。根据需要可采用精确估算或近似估算。

我们观察到多数场景中有大块面积或是被完全照亮或是在阴影中。采用精确估算时可以使用光子图中的信息来识别这些面积而避免使用阴影光线。在两种情况下需要使用阴影光线：全局光子图中距离最近的光子混合了直接照明光子和阴影光子或已定位的照明和阴影光子数目太少。近似估算只是简单地从全局光子图中获得辐射度估算（没有使用阴影光线或光源估算）。

2. 镜面反射：

计算镜面或高光表面反射的辐射度。采用标准蒙特卡罗光线跟踪来计算。通过基于BRDF的重点采样，只使用有限数目的采样光线就能完成大部分情况的计算。

3. 焦散：

计算漫射面或轻微光泽表面的焦散。使用焦散光子图的信息进行估算。不能通过蒙特卡罗采样来计算焦散，因为大部分情况下这几乎是不可能实现的。基于焦散光子图的辐射度估算可以直接看到，因此焦散光子图上的光子数目必须要多。

4. 柔和的间接照明：

计算离开光源后至少被漫反射一次的入射光的辐射度。这些光被物体表面漫反射因此最后的照明非常柔和。近似估算是基于全局光子图的辐射度估算。精确估算使用重点采样来计算间接照明，将光子图的信息与BRDF结合以产生最优化的采样方向。

还有一种使光子图直接可见的方法，即经过处理的光子图可以作为物体表面的纹理贴图（也称为灯光贴图）在渲染时使用。

上述几种都是实现全局照明的基本方法。现在很多算法都是在这个基础上发展起来的。三种方法各有特色：光线追踪比较适合有大量光照的室外场景、角色动画和在空旷的场景中渲染物体；光能辐射度较适合室内外建筑效果图的渲染；而光子映射经常用于焦散、次表面光线散射、烟雾等渲染。大多数流行的渲染器都集成了多种算法，使用哪一个要根据具体的场景来选择。