

密级：_____

浙江大学

博士学位论文



论文题目： 可变材质的交互级全局光照明算法的研究

作者姓名 孙鑫

指导教师 石教英 教授

合作导师 周 昆 教授

学科(专业) 计算机科学与技术

所在学院 计算机科学与技术

提交日期 二零零八年 四月

A Dissertation Submitted to Zhejiang
University for the Degree of
Doctor of Philosophy



TITLE: Research on Interactive
Global Illumination Rendering
with Dynamic Materials

Author: Sun Xin

Supervisor: Prof. Jiaoying Shi

Co-Supervisor: Prof. Kun Zhou

Subject: Computer Science and Technology

College: Computer Science and Technology

Submitted Date: April 2008

摘 要

本文研究了可变材质的交互级全局光照照明绘制算法。可变材质指用户可以通过交互改变绘制场景中物体的材质。固定材质的交互级全局光照照明绘制可以采用预计算辐射传输(PRT)的方法来实现。然而,可变材质的交互级全局光照照明绘制却是近年来刚刚涉及的课题。在本文中提出的可变材质交互级全局光照照明绘制算法,能在用户改变物体材质的同时,交互级绘制全局光照照明效果。

本文提出的绘制算法针对两种绘制场景:第一种,物体几何不变,采用双向反射分布函数(BRDFs)来表示物体的反射材质,不包含散射、折射和透射等其它材质类型。第二种,物体几何可以被用户改变,并且只采用折射和散射这两种材质。

针对第一种绘制场景,我们算法中的主要创新点,是根据不同的传播光路将辐射传输进行分离。用于固定材质的PRT方法,根据绘制效果和光照的线性关系实现了动态光照下的绘制效果的实时更新。但是绘制效果和材质之间不存在这样的线性关系,所以PRT不能支持可变材质的实时交互级全局光照照明绘制。虽然最终到达视点的总的辐射亮度和材质不存在线性关系,但是每一条传播路径的辐射传输和经过的反射物体的材质的乘积成正比。所以,采用我们的方法将辐射传输根据不同的传播光路分离为多个部分,就能够将这个非线性的问题转化为一系列线性问题的加和。我们需要对每一部分的辐射传输分别进行预计算,而不是像PRT中将所有光路的辐射传输一起进行预计算,从而就第一次实现了基于像素的可变材质的实时全局光照照明绘制。我们的算法不但支持空间静态可变材质的实时全局光照照明绘制,而且支持空间动态可变材质的交互级全局光照照明绘制。也就是说,用户不但可以对一个物体的材质作整体修改,而且还可以对一个物体不同部分的材质作不同的修改。

我们还提出了基于顶点的空间静态可变材质的交互级全局光照照明绘制算法。该算法第一次允许用户同时改变材质、环境光照和视点,并且在环境光照不变的时候达到实时的绘制性能。算法主要包含三个创新点:光路切分,预计算传输张量,针对BRDF的镜面项分离和张量分解。通过光路切分,我们有效减小了辐射传输的数据量。预计算传输张量用于辐射传输的表示、预计算、存储和绘制。针对BRDF的镜面项分离和张量分解实现了在动态局部入射辐射亮度、可变材质、动态视点条件下的物体表面的快速着色。

针对第二种绘制场景，我们最主要的贡献是提出了一个完整的绘制算法流水线，第一次实现了可变折射和散射材质的交互级全局光照明绘制。目前大多数的相关的绘制算法都是离线算法，而实时和交互级的绘制算法，都不能兼顾折射和散射这两种材质所涉及的所有绘制效果。我们的绘制算法流水线的输入是场景中的物体几何、材质、光照条件和视点等参数，所以，用户可以在绘制过程中任意地交互修改这些参数。整个绘制算法流水线运行在图形处理单元(GPU)上，并且能够达到交互级的绘制性能。在我们的绘制算法流水线中，有两个环节非常重要，也是我们算法中最主要的两个创新点：几何物体的快速体素化和自适应的非线性光子贴图算法。前者能够允许用户交互改变物体的几何；后者是达到交互级的绘制性能的保证。

本文中的绘制算法有很广泛的应用前景，比如游戏、辅助设计等。其中针对可变折射和散射材质的绘制算法流水线，还是几何变形、交互级几何建模或者进行流体的物理仿真时的交互级可视化算法。

关键词：可变材质，实时绘制，全局光照明，双向反射分布函数，预计算辐射传输，预计算传输张量，非线性光子贴图

Abstract

This thesis is about interactive global illumination rendering with dynamic materials. The users can change the materials by interaction if dynamic materials are used. Global illumination rendering with fixed materials can be solved by precomputed radiance transfer(PRT). Researchers turn to global illumination rendering with dynamic materials these years. The algorithms proposed in this thesis allow the users to change the materials while global illumination results are rendered with interactive performance.

The algorithms in this thesis are about two kinds of rendering scenes. In the first, the geometry remains unchanged, we use the bidirectional reflectance distribution functions(BRDFs) to represent the dynamic reflection materials, and the scattering, reflection and transparency are not included. In the second, the geometry can be changed by users, the objects are with refractive and scattering materials.

For the first kind of rendering scenes, the main contribution of our algorithm is radiance transfer separation for different light paths. The algorithm of PRT used for fixed materials renders with dynamic illuminations in real-time according to the linear relationship between rendering results and illuminations. But the rendering results do not linearly depend on the materials, so PRT can not interactively render with dynamic materials. Even the total radiance to the viewpoint is not linearly dependent on the materials, the radiance transferring along a given light path is linearly dependent on the product of all the reflective materials on the way. So the radiance transfer separation converts the non-linear problem to be the sum of a number of linear problems. We precompute the radiance transfers of different light paths independently, which is different from PRT whose precomputation is for the whole radiance transfer. This is the first algorithm of pixel based real-time global illumination rendering with dynamic materials. Our algorithm not only render global illumination results in real-time with spatial static dynamic materials, which allows the users to change the material of a whole object, but also interactively render global illumination results with spatial variant dynamic materials, which allows the users to

apply different changes to the materials of different parts of an object.

We also propose a algorithm of vertex-based interactive global illumination rendering with spatial static dynamic materials. This is the first algorithm allowing the users to change the material, environment lighting and viewpoint simultaneously, and the real-time performance can be achieved when the environment lighting is fixed. The algorithm includes three contributions: light path separation, precomputed transfer tensors, mirror separation and tensor decomposition of BRDFs. The light path separation reduces the size of the data of the radiance transfer. The precomputed transfer tensor is used to represent, precompute, store and render the radiance transfer. The mirror separation and tensor decomposition of BRDFs are used to achieve fast shading on object surface under dynamic local incident radiance, dynamic material and dynamic viewpoint.

For the second kind of rendering scenes, our main contribution is to propose a complete rendering pipeline, which is the first solution of interactive global illumination rendering with dynamic reflective and scattering materials. The inputs of the pipeline are geometry, materials, illumination and viewpoint, so the users can change all of these parameters while global illumination rendering in real-time. The whole pipeline is implemented on graphics processing units(GPUs) to achieve interactive performance. There are two very important stages of the pipeline, which are also our contributions: fast voxelization makes the rendering which dynamic geometry possible, the adaptive non-linear photon-mapping is critical to the interactive performance of our pipeline.

The algorithms in this thesis have many applications, such as games and computer aided design. Our rendering pipeline for refractive and scattering materials can be also used as real-time visualization algorithm for mesh deformation, interactive modeling and fluid physical simulation.

Keywords: dynamic material, real-time rendering, global illumination, bidirectional reflectance distribution function, precomputed radiance transfer, precomputed transfer tensor, non-linear photon-mapping

目录

摘 要	i
Abstract	iii
第1章 引言	1
1.1 反射材质场景的绘制	1
1.2 折射和散射材质场景的绘制	4
1.3 本文概览	5
第2章 基本理论	7
2.1 辐射测量学和光度学(Radiometry & Photometry)	7
2.2 双向表面散射反射分布函数和双向反射分布函数(Bidirectional Surface Scattering Reflectance Distribution Functions, BSSRDFs & Bidirectional Reflectance Distribution Functions, BRDFs)	8
2.3 绘制方程(Rendering Equation)	11
2.4 预计算辐射传输(Precomputed Radiance Transfer, PRT)	12
2.5 张量(Tensor)	12
2.6 斯涅尔定律和程函方程(Snell's law & Eikonal Equation)	13
第3章 相关工作和研究现状	15
3.1 全局光照明	15
3.2 图像空间的重光照	16
3.3 预计算辐射传输	17
3.4 双向反射分布函数	18
3.5 材质的编辑和浏览	19
3.6 张量的应用	20
3.7 反射材质场景的可变材质的实时绘制	21
3.8 折射和散射材质场景的可变材质的实时绘制	21
第4章 基于像素的可变材质的实时全局光照明绘制	23
4.1 空间静态可变材质的实时全局光照明绘制	24
4.1.1 算法概述	26
4.1.2 双向反射分布函数的线性表述和降维	29
4.1.3 预计算	31
4.1.4 绘制	33
4.1.5 实验结果	34
4.1.6 结论	36

4.2	空间动态可变材质的交互级全局光照明绘制	37
4.2.1	算法概述	39
4.2.2	预计算及数据压缩	42
4.2.3	材质编辑和绘制	43
4.2.4	实验结果	44
4.2.5	结论	47
第5章	基于顶点的空间静态可变材质的交互级全局光照明绘制	49
5.1	算法概述	50
5.2	基于预计算传输张量的重光照	51
5.2.1	辐射传输的入射辐射亮度	52
5.2.2	BRDF	54
5.2.3	重光照	56
5.3	算法实现	58
5.3.1	BRDF的镜面项分离	58
5.3.2	BRDF的编辑和浏览	60
5.3.3	实现的参数和运算代价的分析	62
5.4	实验结果	66
5.5	结论	69
第6章	可变透明和散射材质的交互级全局光照明绘制	73
6.1	算法概述	74
6.2	绘制流水线	75
6.3	绘制流水线的各个步骤	78
6.3.1	体素化	78
6.3.2	八叉树的建立	79
6.3.3	光子的生成	82
6.3.4	自适应的非线性光子贴图	82
6.3.5	逐像素的绘制	84
6.4	算法实现	85
6.4.1	辐射亮度的存储	85
6.4.2	GPU实现中CUDA和OpenGL的选用	86
6.4.3	八叉树的改进	87
6.5	实验结果和应用	88
6.5.1	重光照	90
6.5.2	可变材质	90
6.6	结论	93

第7章 总结与展望	95
7.1 全文工作的总结	95
7.2 未来工作的展望	97
参考文献	99
攻读博士学位期间的研究成果	109

图目录

图 2-1	辐射亮度	8
图 2-2	双向反射分布函数	10
图 2-3	斯涅尔定律	14
图 2-4	光的传播路径和波振面垂直, 此图引自文献 ^[85]	14
图 4-1	不同的绘制技术得到的结果。(a)点光源直接光照, 不考虑物体的遮挡; (b)点光源直接光照, 考虑物体的遮挡; (c)面光源直接光照, 考虑物体的遮挡; (d)面光源直接光照和间接光照, 考虑物体的遮挡和反射。	25
图 4-2	光的传播路径。 P_A : 光经过一次反射到达视点。 P_B, P_C : 光经过两次反射到达视点。	26
图 4-3	Bunny和白色墙壁(包括地面和天花板)的BRDF是可编辑的。Bunny $K_d = 0.2$ $K_s = 0.4$, 白色墙壁 $K_d = 0.3$ $K_s = 0.3$ 。从(a)到(d): Bunny $n_s = 5$, 白色墙壁 $n_s = 5$, L^2 范数的误差为0.0150%; 从(e)到(h): Bunny $n_s = 180$, 白色墙壁 $n_s = 180$, L^2 范数的误差为0.0021%。(a)(e): 直接光照的效果; (b)(f): 间接光照的效果; (c)(g): 最终绘制效果; (d)(h): 参考图像。	35
图 4-4	圆环和平面的BRDF是可编辑的。其中圆环是黄色的, 不包括蓝色通道的颜色分量。圆环 $K_d = 0.2$ $K_s = 0.2$, 平面 $K_d = 0.2$ $K_s = 0.2$ 。从(a)向(d): 我们的结果; 从(e)向(h): 相应的参考图像。(a)(e): 圆环 $n_s = 5$, 平面 $n_s = 5$, L^2 范数的误差0.2698%; (b)(f): 圆环 $n_s = 5$, 平面 $n_s = 180$, L^2 范数的误差0.3114%; (c)(g): 圆环 $n_s = 180$, 平面 $n_s = 5$, L^2 范数的误差0.1624%; (d)(h): 圆环 $n_s = 180$, 平面 $n_s = 180$, L^2 范数的误差0.2640%。	36
图 4-5	Gargoyle和飞机的材质是可编辑的。四周墙壁和飞机都贴上了纹理。	37
图 4-6	空间静态和空间动态材质的全局照明绘制。(a)(c)空间静态材质; (b)(d)空间动态材质。	38
图 4-7	光的传播路径。红色: 经过(R_1)的直接光照。绿色: 经过(R_1, R_2)的间接光照。蓝色: 经过(R_2, R_2)的间接光照。	40
图 4-8	Cubes。(d)是(c)的参考图像, L^2 范数的误差为 $7.518e - 004$ 。	45
图 4-9	Bunny。(d)是(c)的参考图像, L^2 范数的误差为 $2.8787e - 005$ 。 ..	46
图 4-10	Tube。(d)是(c)的参考图像, L^2 范数的误差为 $3.4280e - 004$ 。	46

图 5-1	不同BRDF的维纳斯雕像的绘制效果。右下角是对应BRDF的球模型的参考图像。在周围的墙上可以看到随着维纳斯的反射材质的改变而产生的明显的间接光照效果的不同。这个绘制场景有39.7K个顶点，预计算时间为12.3小时，内存占用389 Mbytes。如果视点、材质和光照都被改变，绘制性能为2.62 fps；如果只有视点和材质被改变，绘制性能为10.94 fps；如果只有视点被改变，绘制性能为57.09 fps。	49
图 5-2	包含间接光照效果的重光照	52
图 5-3	间接光照的光路包含了一个BRDF基函数序列，采用PTT表示。图中黄色的光线代表经过一次反射的间接入射光照，其中 $H(r, p)$ 代表物体 r 对应的BRDF的基函数 p 的系数。图中蓝色的光线代表经过两次反射的间接入射光照	53
图 5-4	镜面项分离。从左向右：初始BRDF；分离后的剩余部分； $0^\circ, 7^\circ, 14^\circ, 21^\circ$ 的镜面项部分。	59
图 5-5	不同的 n_h 和 n_{ω_i} 得到的张量分解的误差	60
图 5-6	不同项数的镜面项分离得到的BRDF的张量分解的重构结果。从左到右：参考图像；没有镜面项分离；镜面项项数为1 (0°)，2 ($0^\circ, 7^\circ$)，3 ($0^\circ, 7^\circ, 14^\circ$)，4 ($0^\circ, 7^\circ, 14^\circ, 21^\circ$)。	61
图 5-7	采集得到的BRDF用Cook-Torrance模型进行拟合	61
图 5-8	不同光照分辨率下的经过一次反射间接入射光照的绘制效果。从左到右， n_L 分别是 $6 \times 4 \times 4$ ， $6 \times 8 \times 8$ ，and $6 \times 16 \times 16$ 。我们可以判断 $6 \times 4 \times 4$ 的分辨率太低，因为桌面上的焦散十分暗淡。包括直接光照和两次反射的间接光照的绘制效果参看图5-13	63
图 5-9	采用不同的光照的压缩维度及采用不同的特征向量的数量时所得到的聚簇张量近似的误差	64
图 5-10	不同的CTA压缩率的绘制效果(采用的特征向量的数量/光照压缩后的维度)。第一行：8/64，16/64，32/16，32/32。第二行：32/64，32/96，64/64，参考图像	65
图 5-11	PTT的绘制结果和参考图像的比较。左：光线跟踪的绘制结果；中：我们方法的直接光照的绘制结果；右：我们方法包含间接光照的绘制结果。	66
图 5-12	室内场景的间接光照的绘制效果。第一行：直接光照的绘制效果；包含了两次反射的间接光照的绘制效果。第二行：地板和沙发的BRDF经过修改编辑后的绘制效果。	67
图 5-13	不同的Blinn-Phong模型的BRDF的飞机模型的绘制结果，其中BRDF的漫反射和高光部分的加权系数不变。第一行，增强了飞机模型底座BRDF的高光效果。第二行，增强了飞机的BRDF的高光效果。	68
图 5-14	编辑BRDF产生的间接光照效果的变化。图中手表采用了不同的金属材质。	69

图 5-15	国际象棋。上：直接光照的绘制效果；下：采用PTT实现的包含间接光照的绘制效果。.....	70
图 6-1	我们的交互级绘制算法得到的绘制结果	73
图 6-2	我们的绘制算法流水线.....	76
图 6-3	(a) 一个物体的三角形面片的表示。(b) 将这个物体体素化为体数据后的结果。.....	79
图 6-4	八叉树的建树算法，为了便于描述，采用二维的形式。我们首先建立金字塔结构记录最小和最大的折射率。然后自顶向下分析，折射率变化范围不大于阈值 $\varepsilon = 0.05$ 的，就将当前的层数作为所对应范围内每个体素的层数。.....	81
图 6-5	使用八叉树的自适应光子传播。(a)光子的位置可以确定所在八叉树的叶节点在八叉树中的层数。(b)传播步长 Δs_{octree} 确保光子直接传播到达当前叶结点的边缘。.....	83
图 6-6	不同算法得到的绘制结果。(a) (b)基于程函方程的绘制方法；(c) (d)我们的绘制方法；(e) (f)光线跟踪得到的参考图像。...	89
图 6-7	光照改变后产生的面焦散和体焦散的变化的绘制结果.....	91
图 6-8	体喷涂的绘制结果：(a)改变物体内部的消散系数；(b)改变物体外部的散射系数。.....	91
图 6-9	用户交互改变透明物体的几何形状后产生的面焦散和体焦散的变化。.....	92
图 6-10	使用基于手绘草图的建模方法得到的两个物体的绘制结果	93

表目录

表 2-1	辐射测量学和光度学中对应的基本术语及其度量	9
表 4-1	预计算的数据	33
表 4-2	所示的基于像素的空间静态可变材质的实验结果的预计算的性能	34
表 4-3	所示的基于像素的空间动态可变材质的实验结果的预计算的性能	45
表 5-1	实验结果的性能数据。在BRDF空间的表示中，我们用‘BP’表示Blinn-Phong，用‘CT’表示Cook-Torrance。‘V’表示只有视点改变时的绘制性能；‘V&B’表示只有视点和BRDF改变时的绘制性能；‘V,B&L’表示视点、BRDF和光照都改变时的绘制性能。所有时间结果中直接光照的分辨率都是 $64 \times 64 \times 6$ 。	71
表 6-1	绘制性能。每个实验结果的分辨率为 640×480 ，使用的显卡为 Nvidia GeForce 8800 Ultra，体数据的分辨率为 $128 \times 128 \times 128$ ，生成光子的阴影贴图的分辨率为 1024×1024 。	90

第1章 引言

真实感图形绘制广泛运用于电子娱乐、计算机辅助设计和计算机图形电影等领域。其主要目标，是在虚拟场景中逼真地模拟和重现现实世界中的各种光影效果，生成各种照片级真实感的图像或者图像序列。

真实感图形绘制的基础，在于遵循并模拟对场景中光的传播的物理描述。由于光的传播的复杂度非常高，因此在绘制时往往带来巨大的计算负荷，并导致运算时间上的高昂代价。但是，电子娱乐、辅助设计等领域有很高的交互性上的要求，需要绘制效果能够实时响应绘制场景的改变，从而形成了真实感图形实时绘制技术的应用背景和需求。

目前大多数的实时全局光照明算法只能支持动态光源，也就是说，用户在改变场景光照的同时，能够实时得到全局光照的绘制结果。比如Sloan^[84]提出的预计算辐射传输(PRT)的绘制框架，实现了动态环境光照、动态视点、静态场景和静态材质下的实时全局光照明绘制。最近三年，可变材质成为一个重要的研究课题，即用户通过交互改变场景中物体的材质，并实时得到绘制图象。本文主要研究了两种绘制场景中的可变材质的实时全局光照明绘制算法，一种绘制场景中物体采用反射材质，另一种绘制场景中物体采用折射和散射材质。

1.1 反射材质场景的绘制

本文中首先致力于这样的场景：物体都采用三角形网格表示，光在传播过程中只在物体表面反射，不考虑空气中的散射和物体中的折射。这类场景的材质可以使用双向反射分布函数来表示。针对这类绘制场景，Kajiya^[47]提出的绘制方程是一种正确、简明而且实用的物理描述，被广泛应用于相关的绘制算法中。

如果绘制场景采用的光源，并不是少数点光源、方向光源等简单光源，而是环境光源、面光源等复杂光源，以现有的硬件性能，还无法实现光源改变时的实时绘制。如果还需要考虑全局光照的绘制效果，也就是考虑到光在传播过程中受到绘制场景的作用而发生的改变，绘制的计算复杂度就更高，所以大量传统的全局光照明绘制算法，比如光线跟踪、辐射度和光子贴图算法，都是离线算法，用户对绘制场景的几何、光照、材质、视点做的任何改变，都需要完全重新绘制，无法达到实时或者交互级的绘制性能。

由于绘制场景的作用，光在传播过程中强度和方向都将经过复杂的改变，而这种复杂改变前后两个不同的光辐射之间的对应关系，被称为辐射传输。显然，辐射传输不但可以用于表示光辐射的前后变化，而且可以用于表示光传播的物理过程。任何实时全局光照明绘制算法，本质上都是分析辐射传输的特性，利用其中的不变的因素加速绘制过程。

辐射传输和从光源入射的辐射强度成线性关系，即到达视点的辐射亮度和入射的光照呈线性关系，辐射传输的这条性质就可以用于加速动态光照的全局光照明绘制。基于这条规律Dorsey^[21, 22]和Dobashi^[20]实现了快速的重光照。Sloan^[84]也利用这一点，提出了PRT的绘制框架：将基光源的绘制效果预先计算并储存，在绘制时只需要将当前光源投影到基光源上，用投影的系数线性组合预计算的结果，就能以很高的性能得到动态光源和动态视点的绘制结果。将大量模拟光线传播的运算预先计算并储存，用预计算的时间和存储空间换取运行时的绘制时间的方法，成为后续大量实时绘制算法的必要工具和手段。

物体的材质，同样对绘制效果有非常重要的作用。尤其在全局光照明的绘制中，由于考虑了光在传播过程中的多次反射，因此能带给我们更为丰富的视觉反馈，而且还可以形成渗色、焦散等绘制效果。但是，光在场景中的传播过程中可能在物体表面经过多次反射，每次反射的效果都受到反射点材质的作用。因此辐射传输和物体材质之间不存在线性关系，PRT的方法也就不能直接应用于可变材质的实时绘制中。同时，非线性问题自然带来了更高的数据空间和计算复杂度，所以这种非线性关系正是研究问题的困难所在。如何利用有限的计算资源和存储资源克服非线性对PRT的限制，充分利用预计算过程降低绘制过程的计算复杂度，从而以合理的时间和空间代价实现可变材质的实时绘制，成为解决问题的重点所在。

PRT的框架无法支持可变材质，其根本原因在于PRT所描述的辐射传输只是光照的函数，而不是材质的函数。换句话说，预计算的辐射传输和材质是互相耦合的，因此无法相应材质变化而组合得到相应的辐射传输。因此，解决这个问题的思路，在于将辐射传输表示为材质的函数，并且基于此进行预计算和绘制。

本文对该问题的解决策略基于以下的观察：虽然辐射传输和场景中物体的材质成非线性的关系，但是光在传播过程中每次在物体表面发生反射时，出射的辐射亮度相对于入射的辐射亮度发生的改变和反射点的物体材质成正比；也就是说，虽然全局的辐射传输和物体的材质成非线性关系，但是局部的单次反射的辐射传输和相应的物体材质成线性关系。我们由此提出的基于像素绘制的算法：将一个场景中的

辐射传输分为许多不同的部分，每一部分的辐射传输所经过的反射的次数以及相应的材质都是确定不变的。经过这种处理，某一部分的辐射传输和其对应的每一次反射的材质都成线性关系，那么用户对场景中的任何一个材质的改变都能体现在每一部分的辐射传输中。也就是说每一部分的辐射传输都是所经过的反射材质乘积的线性函数，而所有这些线性函数的加和就是整个辐射传输关于物体材质的函数。由此，我们实现了动态材质的全局光照的实时绘制。

空间静态可变材质指场景中一个物体上的材质处处相同，用户对它的交互改变也是整体的，即改变之后这个物体的材质还是处处相同的。空间动态可变材质指场景中一个物体上的材质可以处处不同，用户在交互过程中可以对它的不同部分做不同的改变，改变之后这个物体的材质还是可以处处不同。对于空间静态可变材质的绘制场景，辐射传输分离而得到的各个部分数量有限。对于空间动态可变材质的绘制场景，我们将每一个空间动态可变材质的物体的表面分为大量的子区域，并将每个子区域都当作是空间静态可变材质区域来进行近似。这样分离得到的辐射传输的数量将大大增加，计算的时间和空间复杂度也会远远超过现有的硬件条件的限制。我们对此采取了相应的优化和加速算法，从而将基于像素的空间静态可变材质的实时全局光照绘制算法，进一步扩展到了基于像素的空间动态可变材质的交互级全局光照绘制算法。

基于像素的可变材质的实时全局光照算法，最大的局限性在于用户无法改变光照和视点。在许多应用领域，比如室内设计，用户需要同时修改场景的光照、材质以及视点。首先确定光照和视点，然后编辑选择物体材质的做法，会给我们的应用带来很大的困难。我们提出的基于顶点的全局光照绘制算法，不但可以实现空间静态可变材质的实时绘制，还能实现动态视点的实时绘制和动态环境光照的交互级绘制。动态视点和动态环境光照带来了更高的时间和空间的计算复杂度，所以我们需要采用合理有效的手段减小辐射传输的数据空间。为此我们采用以下的方法：我们将辐射传输分为两步，从光源出射的辐射到每个顶点入射的辐射，称之为入射辐射传输，以及从每个顶点入射的辐射到该顶点出射到视点的辐射，称之为顶点着色，显而易见，这两部分的数据空间的加和，要远小于整个辐射传输过程的数据空间，然后我们对这两部分辐射传输分别进行预计算和绘制；对于入射辐射传输，我们采用和基于像素绘制算法类似的方法，将辐射传输进行分类，所不同的是分类不但要考虑到反射的材质，还需要考虑到从光源入射的光照的不同，此外，还需要对不同反射次数的入射辐射传输采用不同的压缩策略进行压缩；对于顶点着色，我们

对材质空间采用镜面项分离和张量分解的方法进行表示和压缩,从而支持动态局部入射光照、动态材质、动态视点下的快速着色。

1.2 折射和散射材质场景的绘制

折射和散射是两种非常重要的光照效果,能给我们带来丰富的视觉享受。折射材质场景的绘制和上文所述反射材质场景的绘制相比有明显的特点:首先,折射材质场景的绘制结果非常的锐利和高频,一般的近似和优化算法带来的误差在视觉效果上非常敏感;其次,光传播到物体表面时绝大部分的辐射强度都发生折射或镜面反射,因此一般不需要使用双向反射分布函数表示物体表面的材质,在绘制时也就不需要在物体表面的光的入射点进行半球面上入射的辐射亮度的积分。现有的折射材质场景的实时绘制算法除了需要利用这些特点外,还需要基于其它一些先验的假设,比如:光在传播过程中经过的折射次数是固定不变或者是预先可以确定的;一个物体内部的折射率处处相同,因此光路由于折射产生的变化只在物体表面产生。许多现有的算法^[10, 67, 106],都是基于这样的简化;而许多镜面反射材质场景的实时绘制算法^[72, 75]也采用同样的假设;其实对于这种约定的场景,折射和反射的绘制算法都很类似^[40, 99]。还有一系列图像空间的折射材质的算法^[103-105],还无法处理参与介质的散射,这样就无法绘制体焦散的效果。现有的大量参与介质算法都是离线算法^[46],主要是因为光在参与介质中强度和方向处处可能发生变化,而不局限于物体的表面,因此计算的复杂度非常高。现有的折射场景的实时绘制算法只能处理简单的均匀的参与介质的散射效果^[50]。Ihrke^[43]的算法虽然能够产生折射和复杂的参与介质的散射效果,但是在改变光照或者材质的时候,无法达到交互级或者实时的绘制性能,同时,也没有提供直接从一个物体生成绘制所需要的体数据的方法。

我们在本文中提出的算法在性能、绘制效果以及交互性上相比以前的算法有很大提高。我们的算法,能够绘制场景中复杂光源的折射和反射效果,以及简单光源的阴影、折射、反射、吸收、各向异性的单次散射、面焦散、体焦散等效果,并能在改变视点时达到实时的绘制性能,在改变光照或材质时达到交互级的绘制性能。我们不需要预先确定折射的次数,折射率、吸收率、散射率可以处处不同。此外,我们的方法可以允许用户有很大程度的交互上的自由,比如改变材质的参数,或者物体的几何等。我们的算法包含一个完整的绘制算法的流水线,其中包括以下步骤:将三角形面片表示的几何物体体素化成体数据表示的场景的材质分布,为这个体数据建立一个用八叉树表示的加速结构,采用我们为这种八叉树定制的非线性光

子贴图算法计算光子在体数据中传播的曲线路径，最后根据视点位置逐像素绘制得到最终图像。对场景中物体作刚体或非刚体变换时都能得到体数据中相应折射率的改变，用户也可以通过材质刷直接喷涂修改体数据的材质属性。由于整个算法流水线都是在图形处理单元上实现的，因此所有不需要使用图形处理单元的用于生成几何体或者体数据的交互级算法都能作为我们算法的输入，比如交互级建模的算法，几何变形的算法，流体物理仿真的算法等。也就是说，我们的算法可以成为许多仿真和动画序列的全局光照明效果的实时可视化的工具。

1.3 本文概览

后文的内容按照以下方式组织安排：第二章讲述了本文工作涉及及使用的相关的基本知识、算法和数学工具；第三章介绍了和相关工作和研究现状，包括真实感绘制、实时绘制、材质的表示等各方面；第四章和第五章介绍的算法用于绘制反射材质的场景，其中第四章的算法是基于像素的绘制方法，还包括了对空间动态可变材质的扩展，第五章的算法是基于顶点的绘制方法；第六章的算法用于透明和散射材质场景的绘制；第七章是对全文工作的总结以及对未来工作的前瞻；最后是本文工作所涉及的所有参考文献，攻读博士期间答辩人所有发表的论文，以及致谢。

第2章 基本理论

这一章主要介绍了本文课题相关的基本理论、算法和数学工具等，包括辐射测量学和光度学，双向表面散射反射分布函数和双向反射分布函数，绘制方程，预计算辐射传输，张量，斯涅尔定律和程函方程。

2.1 辐射测量学和光度学(Radiometry & Photometry)

首先我们介绍一下辐射测量学和光度学中一些重要的基本术语及其度量。表2-1中列举了两者之间的对应关系。可以看到，两者的表述有等价的关系。但是，在真实感图形绘制领域，我们一般习惯采用辐射测量学的术语进行表述。以下具体介绍一下绘制算法中常用的一些术语及其物理涵义。

辐射通量(Radiation Flux) 又称电磁通量，常用符号 Φ 表示，等价于功率。它用来表述单位时间流逝的光能：

$$\Phi = \frac{dQ}{dt} \quad (2-1)$$

其中 Q 表示在时间段 t 内流逝的所有波长的所有光子的能量的总和。辐射通量的度量单位是 $Watt$ 。

辐射照度(Irradiance) 一般情况下特指入射的辐射照度，常用符号 E 表示，用以表述单位面积的物体表面受到的入射辐射通量：

$$E = \frac{d\Phi}{dA} \quad (2-2)$$

其中 A 表示物体表面受到入射辐射通量的面积。

与此相对应的出射的辐射照度，又称为辐射度，常用 M 表示，用以表述单位面积的物体表面发出的辐射通量：

$$M = \frac{d\Phi}{dA} \quad (2-3)$$

辐射照度的度量单位是 $Watt/m^2$

辐射强度(Radiant Intensity) 常用符号 I 表示，用以表述单位立体角内物体发出的辐射通量：

$$I = \frac{d\Phi}{d\omega} \quad (2-4)$$

其中 ω 表示方向， $d\omega$ 表示立体角的微分。辐射强度的度量单位是 $Watt/sr$

辐射亮度(Radiance) 是辐射测量学中最为重要和基本的概念，它事实上等价于人眼视觉感知上的亮度。它常用符号 L 表示，用以表述单位面积的物体表面在单位立体角内所传输的辐射通量：

$$L = \frac{dE}{d\omega} = \frac{d^2\Phi}{d\omega dA^\perp} = \frac{d^2\Phi}{d\omega dA \cos \theta} \quad (2-5)$$

其中 $A^\perp$ 表示物体表面在光传输的方向的垂直方向上的投影面积， θ 表示光传输方向和物体表面法向的夹角。辐射亮度的度量单位是 $Watt/m^2sr$ 。如图2-1所示：

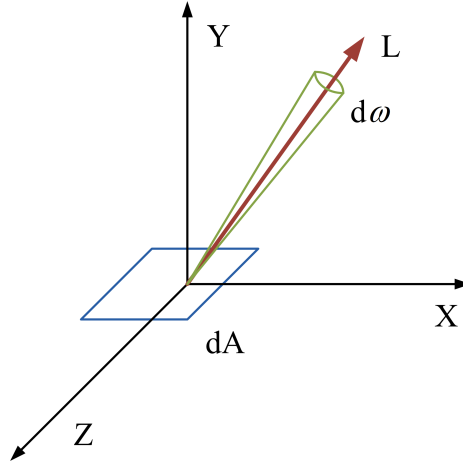


图 2-1 辐射亮度

2.2 双向表面散射反射分布函数和双向反射分布函数(Bidirectional Surface Scattering Reflectance Distribution Functions, BSSRDFs & Bidirectional Reflectance Distribution Functions, BRDFs)

在本文中，我们采用较为简单的几何光学模型，即假设光沿直线方向传播，并且不考虑不同波长的光的属性的差异，因此，所有波动光学模型所描述的物理现象，比如干涉、衍射、极化、荧光效果，在本文中都不涉及。

双向表面散射反射分布函数(bidirectional surface scattering reflectance distribution functions, BSSRDFs)是一种对辐射传输通用性极强的一种描述，可以用来描述任意一个可能的光路上的辐射传输。它表述的是一个物体在表面上的任意一个点

表 2-1 辐射测量学和光度学中对应的基本术语及其度量

辐射测量学	光度学
辐射能(Radiant Energy) <i>Joule</i>	光能(Luminous Energy) <i>Talbot</i>
辐射功率(Radiant Power) <i>Watt</i>	光功率(Luminous Power) <i>Lumen(lm)</i>
辐射照度(Irradiance) <i>Watt/m²</i>	发光度(Luminosity) <i>Lux(lm/m²)</i>
辐射强度(Radiant Intensity) <i>Watt/sr</i>	发光强度(Luminous Intensity) <i>Candela^a(cd or lm/sr)</i>
辐射亮度(Radiance) <i>Watt/m²sr</i>	亮度(Luminance) <i>Nit(cd/m² or lm/m²sr)</i>

的任意一个方向的入射的辐射照度和另外的任意一个点的任意一个方向上出射的辐射亮度之间的反射比率，因此是一个八维的函数：

$$S(x_i, \theta_i \rightarrow x_o, \theta_o) = \frac{dL(x_o \rightarrow \theta_o)}{dE(x_i \leftarrow \theta_i)} = \frac{dL(x_o \rightarrow \theta_o)}{L(x_i \leftarrow \theta_i) \cos(n_{x_i}, \theta_i) d\omega_{\theta_i}} \quad (2-6)$$

其中： S 表示BSSRDF的值； x_i 表示入射的位置； x_o 表示出射的位置； θ_i 表示入射光的方向； θ_o 表示出射光的方向； L 表示光的亮度； E 表示光的照度； n_{x_i} 表示入射点的法向方向； ω 表示空间角。

假设光只在物体的表面受到反射作用，入射点和反射点相同，那么BSSRDF可以简化为双向反射分布函数(bidirectional reflectance distribution functions, BRDFs)。BRDF描述了光在出射方向的亮度和入射方向的照度之间的比值，因此是一个六维函数：

$$f_r(x, \theta_i \rightarrow \theta_o) = \frac{dL(x \rightarrow \theta_o)}{dE(x \leftarrow \theta_i)} = \frac{dL(x \rightarrow \theta_o)}{L(x \leftarrow \theta_i) \cos(n_x, \theta_i) d\omega_{\theta_i}} \quad (2-7)$$

其中： f_r 表示BRDF的值； x 表示反射点的位置； θ_i 表示入射光的方向； θ_o 表示出射光的方向； L 表示光的亮度； E 表示光的照度； n_x 表示反射点的法向方向； ω 表示空间角。如图2-2所示：

BRDF有三个主要性质：

- 非负性：

$$f_r(x, \theta_i \rightarrow \theta_o) \geq 0 \quad (2-8)$$

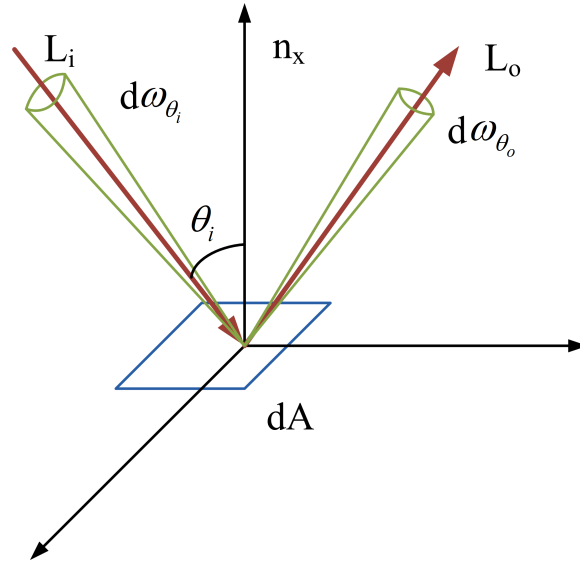


图 2-2 双向反射分布函数

- 对称性:

$$f_r(x, \theta_i \rightarrow \theta_o) = f_r(x, \theta_i \leftarrow \theta_o) = f_r(x, \theta_o \leftrightarrow \theta_i) \quad (2-9)$$

- 能量守恒:

$$\forall \theta_i : \int_{\Omega_x} f_r(x, \theta_i \rightarrow \theta_o) \cos(n_x, \theta_o) d\omega_{\theta_o} \leq 1 \quad (2-10)$$

其中: Ω_x 表示反射点法向对应的半球面的空间角。

本文中使用了两种BRDF模型:

- Blinn-Phong模型

$$f_r(x, \theta_i \leftrightarrow \theta_o) = k_d + k_s \cos^n(n_x, \theta_h) \quad (2-11)$$

- Cook-Torrance模型

$$f_r(x, \theta_i \leftrightarrow \theta_o) = k_d + k_s \frac{F(\beta)}{\pi} \cdot \frac{D(\theta_h) G}{\cos \theta_i \cos \theta_r} \quad (2-12)$$

其中: k_d 表示漫反射分量的系数; k_s 表示高光分量的系数; θ_h 表示入射方向和出射方向的中间方向; β 表示中间方向和法向的夹角; $F(\beta)$ 表示菲涅耳反射效果; $D(\theta_h)$ 表示反射表面微平面的分布; G 表示反射表面微平面的遮挡效果。

2.3 绘制方程(Rendering Equation)

绘制方程是对采用BRDF表示物体材质的场景的辐射传输的数学描述。

如果我们将绘制方程陈述为在入射方向的半球面上积分的形式，那么场景中一个反射点在一个出射方向的亮度，由所有方向的入射照度经过该点的BRDF的作用加和得到：

$$L(x \rightarrow \theta_o) = L_e(x \rightarrow \theta_o) + \int_{\Omega_x} L(x \leftarrow \theta_i) f_r(x, \theta_i \rightarrow \theta_o) \cos(n_x, \theta_i) d\omega_{\theta_i} \quad (2-13)$$

其中： L_e 表示反射点在出射方向上的自发光。这种形式的绘制方程强调的是反射点局部的性质，这种性质就是物体的材质，在这里就是用BRDF进行表示。它所体现的效果就是任意给定半球面方向上入射的辐射照度，就能由该点的BRDF唯一确定在半球面方向上出射的辐射亮度。换句话说，该点的BRDF就决定了该点对所经过的辐射传输施加的作用和改变。

如果我们将绘制方程陈述为在场景中从所有的面入射的光的积分形式：

$$L(x \rightarrow \theta_o) = L_e(x \rightarrow \theta_o) + \int_A L(z \rightarrow x) f_r(x, \theta_{x \rightarrow z} \rightarrow \theta_o) G(x, z) V(x, z) dA_z$$

$$G(x, z) = \frac{\cos(n_x, \theta_{x \rightarrow z}) \cos(n_z, \theta_{z \rightarrow x})}{\|x - z\|^2} \quad (2-14)$$

其中： z 表示场景的面上的点； A 表示面积； V 表示可见性。这种形式的绘制方程强调的是场景中全局概念的辐射传输，可以理解成对场景中任意两个小面片之间辐射传输的描述。事实上全局照明问题是一个全局意义上的问题，除了局部的材质属性外，全局意义上的可见性、遮挡等同样具有直观而重要的物理意义。以上两种形式的绘制方程完全等价，但是由于后者包含了显式的可见性的变量，所以显得更为直观而清晰。

需要指出的是，虽然绘制方程描述的只是一个点对辐射传输的效果，但是其本身的出射光同时也作为别的点的入射光，所有反射点和方向的绘制方程的总合就完整地描述了光传播的整个物理过程。在绘制方程中，可见性、距离、面积等因素由场景几何决定，BRDF代表了场景中材质的特性，再加上光源和视点，就可以决定了场景中的辐射传输的特性，也就决定了全局光照明的绘制结果。

在整个场景中，绘制方程描述的辐射传输是处于平衡状态。但是，我们也可以把辐射传输根据其所经过的反射次数分离，这样构成一个直观的辐射传输的序列，

将式2-13修改得到:

$$L_{i+1}(x \rightarrow \theta_o) = L_e(x \rightarrow \theta_o) + \int_{\Omega_x} L_i(x \leftarrow \theta_i) f_r(x, \theta_i \rightarrow \theta_o) \cos(n_x, \theta_i) d\omega_{\theta_i} \quad (2-15)$$

其中 L_i 表示该辐射亮度在从光源发出后, 经过了 i 次反射。这样前一次的反射的出射辐射亮度成为后一次反射的入射辐射亮度, 就形成了一个Neumann级数。辐射传输的无穷Neumann级数的描述, 同样在物理上等价于前面两个辐射传输的平衡状态的描述。

2.4 预计算辐射传输(Precomputed Radiance Transfer, PRT)

虽然光在场景中的辐射传输的复杂度很高, 而且多次反射也会带来更复杂的物理过程, 但是最终的绘制结果和光照成线性关系。也就是说, 如果我们已经得到了一组光照各自的绘制结果, 那么如果采用这组光照的一个线性组合作为一个新的光照, 只需要以同样的系数组合原先的绘制结果即可, 不需要重新模拟辐射传输的整个过程。

遵循这样的思路, PRT能够实现静态场景和静态材质下的实时重光照。首先选择一组基光照 $\{l_i\}$, 并预计算得到相应的绘制结果 $\{r_i\}$ 。绘制时选取一个光照 L 并投影到基上 $L \approx \sum_i c_i l_i$, 那么相应的绘制结果就能快速得到 $R \approx \sum_i c_i r_i$ 。

需要注意的是, 利用绘制结果和光照的线性关系快速重光照最早应用于基于像素的重光照, 但是基于顶点的实时绘制很大程度上依赖于图形处理单元(graphics processing unit, GPU)的强大性能, 因此复杂光源照射下的基于顶点的实时重光照直到PRT的提出才得以实现。

2.5 张量(Tensor)

张量就是高阶向量的广义概念。通常意义的向量是一阶张量, 矩阵是二阶张量。我们称张量 $A \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$ 的阶为 N ; 它的一个元素表示为 $a_{i_1 i_2 \dots i_N}$; 它的一个模 n 的向量为 $A_{(n)} \in \mathbb{R}^{I_n \times (I_1 I_2 \dots I_{n-1} I_{n+1} \dots I_N)}$, 获得一个模 n 的向量的操作被称为摊平张量。

张量 $A \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_n \times \dots \times I_N}$ 和矩阵 $M \in \mathbb{R}^{J_n \times I_n}$ 的模 n 的积表示为 $B = A \times_n M$, 其中 $B \in \mathbb{R}^{I_1 \times I_2 \times \dots \times J_n \times \dots \times I_N}$, 并且 $b_{i_1 \dots i_{n-1} j_n i_{n+1} \dots i_N} = \sum_{i_n} a_{i_1 \dots i_{n-1} i_n i_{n+1} \dots i_N} m_{j_n i_n}$ 。

一个高维度的张量，可以分解为一个同阶低维度的张量和一组矩阵的积：

$$\begin{aligned} D &\approx Z \times_1 U_1 \times_2 U_2 \dots \times_n U_n \dots \times_N U_N \\ D &\in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N} \\ Z &\in \mathbb{R}^{\hat{I}_1 \times \hat{I}_2 \times \dots \times \hat{I}_N} \\ \forall i : U_i &\in \mathbb{R}^{I_i \times \hat{I}_i}, \hat{I}_i \leq I_i \end{aligned} \quad (2-16)$$

其中 Z 被称为张量核， U 是各阶对应的投影矩阵。

显然我们可以用原始张量和投影矩阵得到张量核：

$$Z \approx D \times_1 U_1^T \times_2 U_2^T \dots \times_n U_n^T \dots \times_N U_N^T \quad (2-17)$$

张量分解可以使用模 n 特征值分解的方法得到。具体方法是，给定所有投影矩阵一个初始值，然后循环更新每一个投影矩阵。更新的方法是：首先使用其他所有的投影矩阵将张量投影到低维度上，然后在需要更新的矩阵所在的维度上摊平并进行特征值分解，得到的特征向量的投影矩阵就是所需要的更新。最后，设定循环收敛的标准，比如投影矩阵收敛或者张量核的迹收敛，结束循环，用得到的投影矩阵和原始张量得到张量核，完成分解。

2.6 斯涅尔定律和程函方程(Snell's law & Eikonal Equation)

斯涅尔定律是最为常用的计算折射光线光路的方法，主要用于计算发生在两种不同折射率材质交界面的折射现象：

$$\frac{\sin \theta_i}{\sin \theta_r} = \frac{n_r}{n_i} \quad (2-18)$$

其中 θ_i 表示入射角， θ_r 表示折射角， n_i 表示入射的材质的折射率， n_r 表示出射的材质的折射率。如图2-3所示：

虽然斯涅尔定律非常直观简单，但是难于计算折射率连续变化的材质内部的折射光路，因此我们需要使用程函方程来描述光路和折射率的关系：

$$|\nabla \Phi|^2 = n^2 \quad (2-19)$$

程函方程特指这种形式的数学方程。而在我们的特定的折射的物理描述中， Φ 指辐射传播的波振面， n 指材质的折射率。这个方程就是非均质折射率材质

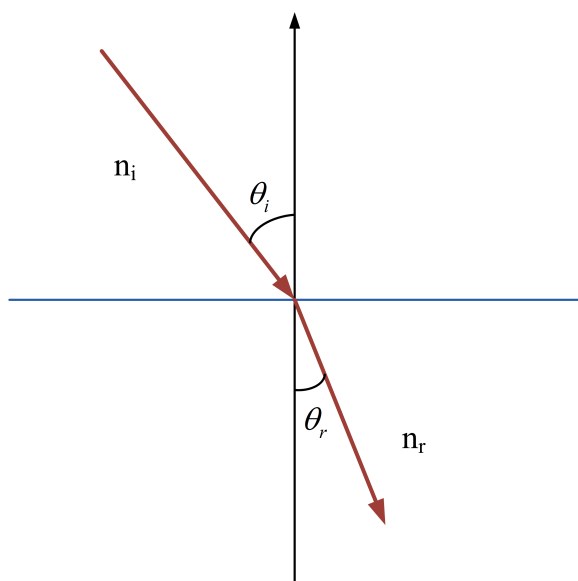
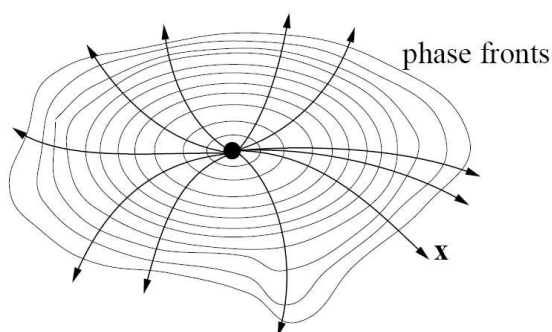


图 2-3 斯涅尔定律

内辐射传播的数学描述，它所描述的物理原理是光线的传播路径和波振面始终保持垂直，如图2-4所示：

图 2-4 光的传播路径和波振面垂直，此图引自文献^[85]

如果引入合适的微分变量 s ，而辐射传输的空间位置 x 是 s 的函数，我们可以得到相应的微分形式的程函方程：

$$\frac{d}{ds} \left(n \frac{dx}{ds} \right) = \nabla n \quad (2-20)$$

程函方程和斯涅尔定律虽然形式不同，但对物理现象的描述完全等价。

第3章 相关工作和研究现状

本章介绍了本文课题的相关工作以及研究现状，包括全局光照明，图像空间重光照，预计算辐射传输，双向反射分布函数，材质的编辑和浏览，张量的应用，反射材质场景的可变材质的实时绘制，折射和散射材质场景的可变材质的实时绘制。

3.1 全局光照明

由于光的传播的复杂度很高，因此直接模拟光的传播来绘制全局光照明效果一般不可能达到实时绘制的性能。这部分主要讨论非实时的全局光照明绘制算法。

非实时的全局光照明绘制算法主要分为两类：有限元辐射度方法和光线跟踪算法。这两种算法都符合绘制方程对光传播物理过程的数学描述。

辐射度方法 是采用有限元的方法求解场景中的辐射传输。主要的思想是将场景离散成大量细小的面片，然后根据这些面片之间在平衡状态下的辐射传输关系建立线性系统，通过求解这个线性系统得到全局光照明绘制结果。

Goral^[29]，Torrance^[13]和Greenburg^[65]等人最早将辐射度方法引入计算机图形学。最初的辐射度算法由于硬件和算法的性能的限制，需要假设每个面片向所有方向反射的辐射亮度都相同，即所有面片的材质都只能是漫反射材质，这样的话任何两个面片之间的辐射传输关系可以简化为他们之间的可见性的函数，从而简化了系统的建立和求解。从Immel^[44]开始，辐射度算法克服了漫反射材质的限制，实现了高光材质的绘制。主要的思路有两种：一种是将方向性引入辐射传输的系统，因此辐射传输的关系的复杂度就提高了许多，原先建立的线性系统只是可见性的函数，而现在任意三个面片的有序的三元组都需要独立地表示一部分辐射传输^[4]，并在这样的系统下完成模拟整个场景的辐射传输直到收敛到最后的全局光照明的绘制结果^[9]。另一种方法是采用光线跟踪的方法补充辐射度方法中的非漫反射材质部分的全局光照明效果^[79, 81, 82, 92, 93]。此外，还有大量的工作致力于提高辐射度算法的性能^[14, 30, 37, 89, 107]，又或者是将算法扩展到动态场景^[23]。

光线跟踪 是通过模拟光线在场景中的传播过程来得到全局光照明的绘制结果，具体采用大量采样离散光线的统计结果来估计连续的辐射传输的真实效果^[80]。

Appel^[3]提出了光线投射算法来消除场景中的隐藏面,应当是所有光线跟踪算法最早出现的雏形,虽然不涉及辐射传输的模拟和计算,但是已经能体现出基于像素的绘制效果,以及对复杂场景的兼容性,并且避免了场景三角化等操作。Whitted^[102]将其扩展为递归算法来模拟场景中的辐射传输,并且提出了一个与之相匹配的全局光照明模型,从而建立了一个完整的用于计算全局光照的光线跟踪算法,但是这个算法所适用的场景只能采用漫反射和镜面反射的材质。针对材质上的限制, Cook^[15]提出分布光线跟踪算法,从而使得光线跟踪算法能够处理介于漫反射和镜面反射之间的高光材质的绘制。Kajiya^[47]提出的绘制方程是对辐射传播的物理过程的数学描述。由于非常清晰、简洁、直观、通用,绘制方程具有非常重要的意义,并且广泛应用于后来的各种全局光照的算法中。而同时据此提出的蒙特卡罗(Monte Carlo)光线跟踪算法,是对一般性全局光照明绘制问题的一个解决方案。蒙特卡罗方法从本质上而言是一种基于统计的,并通过随机采样得以实现的随机游走方法。相比于辐射度的解决方案,蒙特卡罗光线跟踪方法具有更好的性能和通用性:不但时间和空间的复杂度较低,并能达到像素级的绘制效果,更重要的是能适用于许多复杂而特殊的光照、几何和材质。因此成为目前主流的研究题目和内容。Jensen^[45]提出的光子贴图(photon mapping)算法是对蒙特卡罗光线跟踪算法的一个重要的发展。光子贴图算法通过建立独立于场景的光子传播的数据结构来加速绘制。它所付出的代价是以有偏估计代替经典的蒙特卡罗方法的无偏估计,在空间上需要额外维护一个光子贴图的数据结构,并且为其建立加速结构,如Kd-tree,为绘制做加速准备。而它所获得的益处有很多:首先能够在有限的采样数量下很大程度上消减蒙特卡罗采样必然带来的采样噪点;其次通过建立中间数据结构使得计算的时间复杂度得到数量级上的降低,从而获得性能上的巨大提升;此外还方便我们添加启发式的先验规则以扩展支持不同类型的几何结构和绘制效果,比如采用焦散贴图的方法,就能以较少的光子数获得高质量的焦散的绘制效果。

3.2 图像空间的重光照

图像空间的重光照是特指静态场景、静态视点、静态材质在不同光照条件下的快速重光照算法。其中一些算法,对视点和几何可以做小范围的改变。

最早用于图像空间的重光照算法的策略需要利用光照和辐射传输的线性关

系。这和后文3.3中介绍的预计算辐射传输的思想是相同的。其中代表性的工作有Dorsey^[21, 22]。Dobashi^[20]也做了类似的工作，他们把可选择的光照投影到基上进行预计算，并且利用增量计算来加速绘制过程。

另一类图像空间的重光照算法是利用帧缓存的策略。由于是图像空间的算法，得到的是逐像素的绘制精度，因此每个像素有相对独立的决定绘制结果的几何或光照信息。因此我们可以为绘制图像建立额外的帧缓存，帧缓存的每个像素存储对应像素的重光照所需的几何或光照信息。Saito^[76]最早提出使用几何帧缓存存储每个像素对应的几何信息。之后，Gershbein^[28]和Pellacini^[69]等人将几何帧缓存扩展为深度帧缓存，不但存储几何信息，还存储光照信息，同时由于绘制结果和帧缓存都是维度相同的二维图像，因此非常适合使用GPU进行加速。

还有一类的图像空间的重光照算法利用的是多光源(many lights)的策略，即用大量的简单光源，如点光源或平行光源等来近似模拟复杂光源和全局照明的间接光照。Walter^[94]的方法利用了大量的点光源来模拟复杂光源和间接光照。而之后Hasan^[39]的方法进一步利用了辐射传输的稀疏性和相关性，采用采样部分辐射传输替代穷举所有辐射传输的方法，将场景规模和点光源的规模提高到十分惊人的程度，并获得可观的性能和高质量的绘制结果。需要指出的是，许多图像空间重光照的算法是帧缓存和多光源算法的混合体，比如Hasan^[38]就是兼有帧缓存和多光源的思想，并且是第一个将类似的重光照算法提高到实时的绘制性能的算法。

3.3 预计算辐射传输

同样基于光照和绘制效果的线性关系，并利用GPU的优越性能，Sloan^[84]提出了PRT框架，这是第一个复杂光照下的实时全局照明绘制算法。和前文3.2中的图像空间的重光照算法相比，主要有以下的特点：PRT是基于顶点的绘制算法，无法达到逐像素的绘制精度；PRT能够支持动态环境光照，动态视点，静态场景；PRT能够达到实时的绘制性能，绘制时的时间复杂度和辐射传播的复杂度无关，只和场景中的顶点数量和基光照的数量有关，辐射传输的过程模拟包含在预计算的过程中。随后，Sloan^[83]又使用聚类主元分析(Clustered Principle Component Analysis, CPCA)对预计算得到的辐射传输进行压缩和加速，从而提高了算法所支持的场景的规模。由于Sloan采用的是球面调和(spherical harmonic, SH)基作为光照的基，所以难以得到高频的绘制效果，如小光源、金属高光、尖锐的阴影边缘等。因此，Ng^[61]引入Haar小波基，得到了高频的光照效果。对于只包含漫反射

材质的绘制场景, 可以在绘制时改变光照和视点。由于硬件运算能力和内存空间的限制, 对于包含非漫反射材质的场景, 无法在绘制时改变视点, 这样就蜕化成一个图像空间的全局光照的重光照算法。对此, Ng^[62]将绘制方程表述为一个三重积(triple product)问题, 虽然在当时的硬件条件下还无法达到实时或交互级的绘制性能, 但是三重积是对静态场景、动态光照、动态视点的一个正确、清晰而通用的数学描述, 同样具有很重要的意义。除了三重积的思路, Liu^[54]和Wang^[97]通过对非漫反射的BRDF进行分解, 同样能得到高频动态光照、高光材质、动态视点的实时的绘制性能。而Tsai^[90]将球面径向基函数(spherical radial basis functions, SRBF)引入PRT框架, 绘制效果介于SH基和Haar小波基之间, 同时利用聚类张量近似(Clustered Tensor Approximation, CTA)有效地压缩了预计算得到的辐射传输的数据, 从而大大降低了绘制的时间和空间的代价, 性能上得到数量级上的提升。Mei^[57]的方法实现了动态场景在环境光下的阴影和间接光照的绘制。根据物体遮挡性和局部光源的可叠加性, Zhou^[108]提出了阴影场的框架, 实现了动态场景在环境光源和复杂局部光源下的直接光照效果的实时或交互级绘制。Sun^[86]将三重积推广为更一般意义下的多重积(multiple product)问题, 同时采用阴影场的绘制框架, 能够支持复杂光照下高光材质物体非旋转的刚体运动场景的实时绘制, 只是在切换运动物体时需要数秒的更新时间。Ren^[74]采用一些球来近似变形物体, 同时加速了阴影场的叠加算法, 从而将阴影场的算法扩展以支持可变形几何场景的直接光照的绘制效果。

3.4 双向反射分布函数

BRDF^[59]是一种常用的材质的简化模型, 广泛运用于不考虑透明、半透明、散射的场景绘制。BRDF是关于光源入射方向和出射方向的函数, 入射点和出射点是物体表面的同一个点。根据不同的标准, BRDF有多种分类方式。如果物体表面的BRDF处处不同, 那么就是一个六维函数, 称为空间动态(spatial variant) BRDF^[1, 33, 52, 70]; 如果一个物体表面的BRDF处处相同, 那么就是一个四维函数, 称为空间静态(spatial static)BRDF^[2, 5, 6, 87]。许多BRDF模型关于反射点的法向中心对称, 称之为各向同性BRDF^[7, 16, 71]; 其它的我们称之为各向异性BRDF^[98]。如果各向同性BRDF同时又是空间静态BRDF, 可以简化为三维函数。显然, 空间动态BRDF和各向异性BRDF不但维度更高, 而且更具有通用性和一般性。

许多经典的BRDF模型使用参数来描述其形态或者表面结构的假设, 并有显式解析表达。它们中间有一些是基于人们对BRDF的形态的主观印象, 比如Phong^[71]、Blinn-Phong^[7]; 有些是基于对BRDF产生机理的假设, 比如Cook-Torrance^[16]; 还有一些是基于对测量结果的拟合, 比如Ward^[98]等。这类的BRDF虽然清晰简洁, 但是距离现实世界中的真实材质, 还是有很大的距离。Ngan^[63]将采集得到的BRDF拟合成不同的解析形式的BRDF, 然后将拟合得到的BRDF和原始采集的BRDF分别进行绘制并比较结果, 得到了许多有意义的结论, 比如: 基于BRDF产生的物理机理假设的BRDF模型, 相比于基于BRDF形态主观印象的BRDF模型, 能更好地逼近真实数据; 基于多叶(lobe)的BRDF模型, 相比于基于单叶的BRDF模型, 能更好地逼近真实数据; 考虑菲涅尔(Fresnel)反射效果的模型, 相比于其他的BRDF模型, 能更好地逼近真实数据; 等等。但是无一例外的, 所有的拟合得到的解析形式的BRDF, 不论在误差的统计数据上, 还是在绘制效果上, 和采集得到的真实的BRDF有很大的距离。原因主要有两点, 一个是采集和拟合的系统误差, 一个是解析模型本身就不足以精确描述BRDF。因此, 最近几年人们开始热衷于采用数据驱动(data driven)的BRDF^[55], 即直接将从真实材质采集得到的BRDF采样作为BRDF进行操作使用, 并不拟合为某一种解析形式的BRDF。这种思想是认为采集得到的数据集本身代表了BRDF的所有所能得到的性质, 通过数据分析和处理的方法能够最大程度地获取和保留BRDF的性质和效果, 虽然不能得到显式的BRDF的数学描述, 但是通过PCA、小波或者流形(manifold)等方法进行分析, 建立线形或者非线性的BRDF空间来进行压缩和表示, 相对于解析形式的BRDF能获得更好的分析逼近和绘制结果。

3.5 材质的编辑和浏览

材质的编辑指的是用户修改现有的BRDF从而得到新的BRDF。如果是解析形式的BRDF, 那么直接修改其参数, 就能实现BRDF的编辑。许多商业软件所提供的BRDF编辑功能, 就是采用这种方法, 比如Maya和3DsMax。这种编辑方式的优点是简单直接, 不需要额外的操作; 此外, 许多BRDF的参数有比较直观的物理意义, 这种编辑方式有时候也比较直观, 其改变引起的绘制效果的改变也能被用户所主观预测。也有一些软件支持用户订制的BRDF^[26]。但是许多时候BRDF的参数并不足够直观, 改变BRDF后的绘制效果随机而不可预测, 因此不是合适的编辑方式。Ben-Artzi^[6]分离出BRDF的用户控制分量(user-controlled variable)提供给用户

进行材质编辑, 用户控制分量不但便于预计算和绘制, 而且和绘制效果有非常直观的关联, 因此用户可以很方便直观地把BRDF编辑成自己需要得到的效果。

材质的浏览指的是用户在给定的BRDF空间中选择需要的BRDF。显然, 最简单的BRDF空间由所有的BRDF的采样组成, 浏览的时候只需要从中间选择需要的采样即可。因此, 只需要有一个可变BRDF的快速绘制的方法, 就能实现BRDF的浏览和可视化, McCool^[56]便是采用这样的方式, 他们通过分解BRDF加速绘制。而在数据驱动的BRDF的思路下^[55], 采样得到的BRDF被认为能构成一个完整的BRDF的空间, 那么问题在于如何表示这个空间并进行插值得到采样以外的同一个空间内的BRDF。Matusik^[55]尝试了PCA、小波、流形等方法, 最终采用了将数据分片镶嵌构成流形(manifold charting)的方法构筑BRDF的流形空间, 并进行插值浏览。Ngan^[64]主要是解决材质浏览时的插值步长的问题。他们认为应当用绘制效果的差异来衡量BRDF的差异大小, 这样更符合人眼认知心理; 而且通过实验证明, 绘制效果的差异和BRDF参数的差异不具有可比性。因此他们提出了一种新的BRDF的浏览方法, 使得BRDF插值的步长和绘制结果的差异相匹配。

3.6 张量的应用

张量就是高阶的向量。在图形学中, 最广泛应用的当然是一阶张量, 即向量, 以及二阶张量, 即矩阵。但是更高阶的张量, 比如三阶张量等, 也有很多的应用。

高阶张量的一种应用是表示物理上本身具有高阶向量含义的数据。在采用BRDF表示材质的绘制场景的全局光照明问题中, 建立的辐射传输系统是所有可见的有序面片三元组, 而所有这样的三元组的集合就是一个三阶张量, 因此Aupperle^[4]就采用一个三阶张量来表示一个绘制场景中的全局光照明辐射传输。

高阶张量的另一种应用在于可以在多个维度上进行线性压缩, 而并不像PCA只能在一个矩阵的两个维度上进行压缩。Vasilescu^[91]用三阶张量来表示双向纹理函数(bidirectional texture functions, BTFs), 三个维度分别为光照、视点和BTF采样, 并在光照和视点两个维度上进行压缩, 从而实现了BTF的快速绘制。Wang^[96]用张量表示许多种高阶的可视化数据, 并且在多个维度上进行压缩, 从而能够实现快速的可视化。Tsai^[90]用三阶向量表示预计算得到的辐射传输, 三个维度分别是光照、传输和场景的顶点, 然后采用聚类张量近似的方法, 将顶点分为多个不同的簇, 每个簇在三个维度上都进行压缩, 从而极大地降低了绘制时的时间和空间的复杂度。

3.7 反射材质场景的可变材质的实时绘制

反射材质场景指的是采用BRDF表示物体材质的绘制场景。在前文3.1中所提到的传统的离线的全局光照明算法，在3.2中所提到的图像空间的重光照算法，以及在3.3中所提到的预计算辐射传输算法，这些算法中的大多数都是用于这种场景的全局光照明绘制。

由于全局光照的辐射传输和物体的材质成非线性关系，Ben-Artzi^[6]抛弃了间接光照，只绘制直接光照效果。由于直接光照相关的所有辐射传输都经过场景中物体的一次反射，所以它们都和反射处的材质成线性关系，也就是说每个像素的绘制结果和该像素可见的反射点的BRDF成正比；然后选定一组BRDF的线性基，预计算每个像素在每个基上的绘制结果，在修改BRDF时重新组合预计算的结果，就能实时绘制固定光照、固定视点、空间静态可变材质、基于像素的直接光照的结果。为了引入全局光照的绘制效果，Ben-Artzi^[5]对前面的方法^[6]进行扩展，采用了一种多项式的表示形式来描述全局光照明辐射传输和物体BRDF的非线性关系，同时也能根据反射次数不同对辐射传输进行不同程度的压缩和近似。基于这种多项式的预计算和绘制，实现了固定光照、固定视点、可变材质、基于像素的全局光照的实时绘制。这两种方法的局限性主要有两点：不能支持空间动态可变材质，不能同时改变光照和视点。而后者对于实际应用尤为重要。

3.8 折射和散射材质场景的可变材质的实时绘制

折射材质场景的绘制，和一般非折射场景绘制的最大区别在于光线的传播路径未必是直线。如果场景有数个均质透明材质区域组成，光的传播路径在不同材质的交界面上会发生偏折。如果材质区域内部的折射率是非均质的话，那么传播路径可能是方向连续变化的曲线。

有许多工作都是针对非直线的光线跟踪的应用。使得光线传播路径改变方向的原因可以是折射^[78, 85]，也可以是别的原因，比如说重力场^[101]，或者是人为的对线形空间的非线性扭曲^[32, 77]等。这些方法大多关注如何得到特定物理约束下正确的传播路径，而并不致力于性能的提高，所以大多是离线算法。

对于强调性能的交互级或实时算法，大多需要利用GPU的并行性带来的性能上的提升^[35, 36, 41, 50, 67, 95, 100]。尤其是一系列基于图像空间的折射算法^[103-105]，不但绘制

性能高，而且能很容易达到实时的绘制性能。

但是这些绝大多数交互级或实时的折射材质的绘制算法都假设只在物体表面发生折射，不考虑折射率连续变化的效果，同时也很难同时实现非均质的散射效果^[35, 36, 41, 50, 95, 100]。其中还有许多算法需要预先决定每条光线发生折射的次数^[67]，最典型的是一些基于图像空间的折射算法^[103-105]。

离线的非线性光子贴图算法^[34]能够克服这些约束，场景中的折射率可以处处不同，还能同时绘制散射的效果，并不需要限制折射发生的次数。但是，这个方法难以达到实时的绘制速度；另外由于作者着力于大气折射这种低频的绘制效果，因此只在光子传播时采用曲线的传播路径，而从视点发出的光线仍然延直线传播。

基于程函方程的绘制方法^[43]采用均匀采样的非压缩体数据表示整个绘制场景，用程函方程决定辐射传输和视点发射光线的曲线路径，从而能在改变视点时达到实时的绘制性能，并且能得到折射、反射、散射等丰富的绘制效果。所不足的是在于改变光照时无法达到交互级的绘制性能，而且也不支持对场景进行编辑修改。

第4章 基于像素的可变材质的实时全局光照明绘制

这一章详细介绍了基于像素的可变材质的实时全局光照明绘制算法：首先在后文4.1中实现了空间静态可变材质的实时全局光照明绘制；在4.2中将4.1的算法扩展到了空间动态可变材质的交互级全局光照明绘制。

现有的基于预计算的全局光照明绘制算法都假设场景中物体的材质固定不变，这样从入射光照到出射辐射亮度之间的传输变换就是线性变换。通过对这种线性变换的预计算，可以在动态光源下实现全局光照明的实时绘制。但是当材质可以改变时，这种线性变换不再成立，因此许多现有算法无法直接用于动态材质的场景。在4.1中我们提出了一种方法：在修改场景中物体材质时，可以实时得到场景在直接光照和间接光照下的绘制效果。我们将最终到达视点的辐射亮度根据其之前经过的反射次数及相应的反射材质分为多个部分，每个部分和先后反射的材质的乘积成正比，从而把该非线性问题转化为线性问题。我们又将所有可选的材质都表示为一组基的线性组合。我们将这组基作为材质赋予场景中的物体，就有各种不同的组合方式，我们预计算每种组合下所有部分的出射辐射亮度。在绘制时，我们根据各物体材质投影到基上的系数线性组合预计算的数据就能实时得到最终的全局光照明的绘制结果。

如果在绘制过程中不但允许用户对物体的材质作修改，而且允许对一个物体的不同部分的材质作不同地修改，我们称之为空间动态可变材质。相比于空间静态可变材质，空间动态可变材质对最终的出射的辐射亮度的影响更为复杂，目前没有任何交互级全局光照明绘制算法能够支持空间动态可变材质。在4.2，我们将一个空间动态可变材质物体划分成许多空间静态可变材质子区域来近似模拟，然后采用在4.1的方法进行预计算和绘制。但是进行这样简单的扩展存在许多问题。最主要的原因是子区域的数量远远高于场景中可变材质物体的数量，大量的子区域就带来了大量的传播路径，相应的需要预计算并存储的辐射传输的数据量就远远高于空间静态可变材质的情况。由此也就给绘制带来巨大的运算负荷，无法达到交互级的绘制性能。因此4.2中工作的重要贡献在于对这种需求采用了合理的近似和压缩策略。我们提出的这些近似和压缩策略满足以下四个要求：有效减少辐射传输的数据量，没有明显的压缩带来的绘制瑕疵，尽量保留高频和主要的绘制效果，有利于提高绘制的性能。采用我们的近似和压缩方法，我们实现了空间动态可变材质的交互

级全局光照明绘制, 而且能得到高质量的绘制效果。

我们的方法适用于物体几何、光照和视点都不改变的场景。我们使用双向反射分布函数来表示物体的材质, 不考虑折射或者半透明的情况。我们的实现最多包含了两次反射, 可以绘制得到一些很有趣的全局光照明效果, 比如渗色、焦散等等。

4.1 空间静态可变材质的实时全局光照明绘制

物体的材质描述了物体对光的反射特性, 包括反射光的强度、颜色和方向分布等。直接光照下材质可以表现为物体的颜色、明暗、高光或者漫反射等。在全局光照的框架下, 需要考虑从光源出射的光经过反射后照亮场景的效果。比如, 反射光能进一步地照亮场景; 反射物体的颜色能影响其它物体的颜色, 即渗色(color bleeding); 高光材质可以形成焦散(caustics)。当物体采用可变材质时, 也就是说物体的材质被改变时, 光照效果都会受到相应的影响。比如, 物体颜色的变化, 会使得周围受到它反射的物体的颜色也产生相应的变化。再比如, 材质逐渐从漫反射变为高光镜面反射时, 不但可能会从接近于木头、粉刷墙面之类的效果变得接近于金属的效果, 而且周围受到其反射光照射的物体也可能会逐渐形成程度不同的焦散。

采用光线跟踪算法按照绘制方程计算模拟光的传播无法达到实时的绘制性能。采用OpenGL或者DirectX等绘制引擎, 在GPU的支持下, 基本功能无法处理遮挡产生的阴影和多次反射的效果。阴影贴图(shadow mapping)不支持复杂光源的软阴影。Sloan^[84]提出的PRT框架能实现复杂光源照射下的场景的重光照, 但是物体的材质固定不变。直到Ben-Artzi^[6]才将材质作为独立动态变化的因素纳入实时绘制的框架, 能支持复杂光源的直接光照效果, 但不能实现间接光照的效果, 所以不是一个全局光照明的算法。

图4-1包含不同绘制技术的效果。图4.1(a)采用点光源, 没有阴影和间接光照, 实际上是DirectX或者OpenGL的结果。图4.1(b)是采用点光源的直接光照的结果, 有硬阴影。图4.1(c)是面光源的直接光照的结果, 有软阴影。图4.1(d)是面光源的直接光照和间接光照的结果, 场景变亮, 阴影区被照亮, Bunny上有明显的渗色效果: 墙壁的红色使得Bunny也显现红色。可见, 采用复杂光源和全局光照明技术可以极大地提升绘制的视觉效果。

我们致力于复杂光照下改变材质的全局光照明效果的实时绘制。具体描述为: 在静态场景中采用复杂光源(我们采用的是面光源), 材质变化时能实时地得到绘制效果。绘制效果包括直接光照效果, 比如材质效果和软阴影, 以及间接光照的效

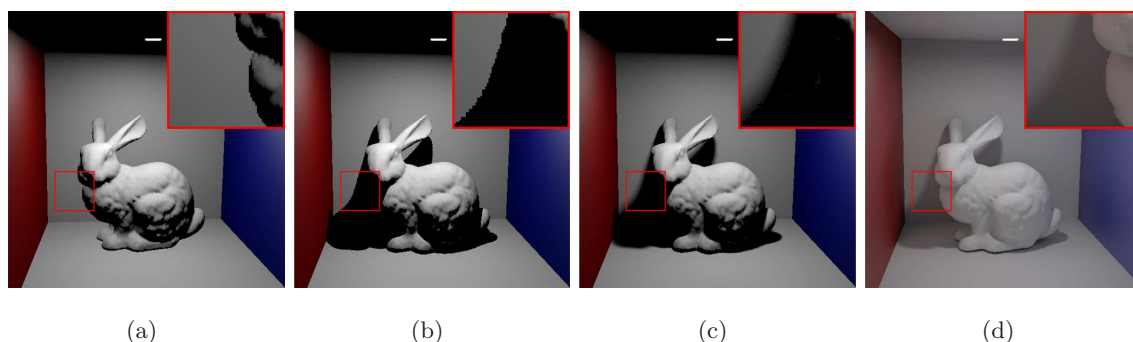


图 4-1 不同的绘制技术得到的结果。(a)点光源直接光照, 不考虑物体的遮挡; (b)点光源直接光照, 考虑物体的遮挡; (c)面光源直接光照, 考虑物体的遮挡; (d)面光源直接光照和间接光照, 考虑物体的遮挡和反射。

果, 比如阴影区被反射光照明, 渗色和焦散等。我们采用双向反射分布函数描述材质, 并且采用Blinn-Phong^[7]的BRDF模型描述物体的材质, BRDF的编辑和修改可以通过改变其参数来实现。

我们的算法基于如下假设: 场景几何结构、光照和视点不变, 场景由多个独立的物体组成, 不同物体采用不同材质。针对绘制效果和材质的非线性关系, 将最后出射的辐射亮度根据其在场景中反射的次数和材质分类, 每一类都和各自反射物体的BRDF成正比。比如直接光照的辐射亮度与其反射的材质成正比, 经过两次反射的辐射亮度和先后两次反射的材质的乘积成正比, 同样可以扩展到多次反射得到的辐射亮度, 从而间接建立绘制效果和材质的线性关系。首先预计算每一部分的辐射亮度, 然后根据当前的材质分布线性组合预计算的结果, 就能得到最终的绘制效果。算法包含以下步骤: 首先, 将BRDF的定义域进行离散, 用向量表示每个BRDF, 构成一个BRDF空间; 密集采样BRDF然后PCA降维得到一组数量有限的BRDF的基; 然后, 将这些基分配给场景中的各个物体, 不同的分配方法得到场景材质分布的不同组合, 对所有组合中的每一个使用路径跟踪的方法绘制全局光照的效果, 将出射的辐射亮度根据不同的反射次数和不同的反射路径分别存储; 在绘制时根据当前材质计算其系数, 并以此线性组合预计算的数据得到最终结果。我们使用GPU的半透明混合(alpha blending)来加速线性组合, 达到实时的绘制速度。我们的算法实现了一次反射(直接光照)和两次反射的效果。

综上所述, 我们算法的创新主要有两点: 一是将全局光照的出射辐射亮度根据光线到达视点之前经过的反射次数和反射材质分类计算, 从而将动态材质的全局光照这个非线性问题转化为一组线性问题的叠加; 二是用一组数量有限的基表示可

选BRDF所组成的空间，每一个可选BRDF都能表示成这组基的线性组合。

我们的方法可在场景建模时进行材质的选择和编辑，目前大多数工具只提供直接光照甚至不考虑遮挡的反馈效果。我们的方法还能快速生成动态材质的绘制效果序列，并且序列的生成可以实时响应用户的交互。

4.1.1 算法概述

若物体没有自发光，并且为了方便后文的推演，同时减少变量的下标的层数，我们将式2-14的绘制方程重新写作：

$$\begin{aligned} L(x, x') &= \int_{\Omega/2} L(x', x'') \rho(x, x', x'') G(x, x', x'') dx'' \\ G(x, x', x'') &= V(x, x') V(x', x'') \cos \theta \end{aligned} \quad (4-1)$$

其中 x, x', x'' 表示场景中的点， $L(x, x')$ 表示点 x' 射向点 x 的辐射亮度， $V(x, x')$ 表示点 x 和点 x' 的可见性， $\Omega/2$ 表示积分范围是点 x' 的法向所在的半球面内的所有点 x'' ， $\rho(x, x', x'')$ 表示光线沿点 x'' 到点 x' 的方向入射，沿点 x' 到点 x 的方向出射时的BRDF的值， θ 表示从点 x' 到点 x'' 的方向和点 x' 的法向的夹角。

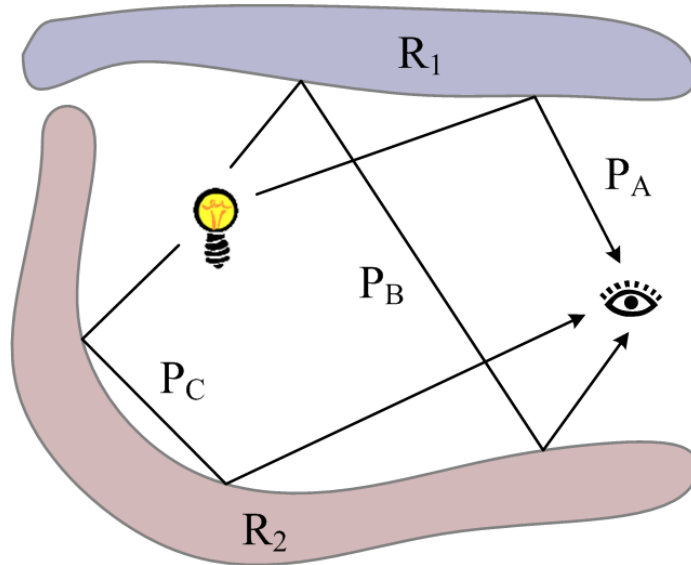


图 4-2 光的传播路径。 P_A ：光经过一次反射到达视点。 P_B, P_C ：光经过两次反射到达视点。

作如下假设：场景由 N_r 个物体组成， (R, ρ) 表示一个物体及其BRDF， ρ_R 表示物体 R 的BRDF。图4-2中有两个物体 (R_1, ρ_{R_1}) 和 (R_2, ρ_{R_2}) 。从光源出射的光线可能

先后经过多次反射，一个经过 n 次反射的过程记作 P_n ：

$$P_n = (R_1, \rho_{R_1}) (R_2, \rho_{R_2}) \cdots (R_n, \rho_{R_n}) \quad (4-2)$$

采用如下的表示： $L_n(x, x')$ 表示点 x' 射向点 x 的辐射亮度，并且光在到达点 x' 前经过了 n 次反射，但对反射路径没有约束。 $\tilde{L}(x, x', P_n)$ 表示点 x' 射向点 x 的辐射亮度，而且光在到达点 x' 前按 P_n 所描述的路径经过了 n 次反射。如图4-2， P_A 的路径可描述为 P_0 ， P_B 的路径可描述为 $P_1 = (R_1, \rho_{R_1})$ 。将入射的辐射亮度根据不同的反射路径和次数进行进一步区分，并以此重新陈述绘制方程，那么点 x' 射向点 x 的辐射亮度包含所有反射次数后到达 x' 的辐射亮度经过 x' 反射到达 x 的辐射亮度，如下：

$$L(x, x') = \sum_{n=0}^{\infty} L_n(x, x') \quad (4-3)$$

经过同样反射次数，不同反射路径到达 x' 的辐射亮度经过 x' 反射到达 x 的辐射亮度可递归地表示如下：

$$L_{n+1}(x, x') = \sum_{P_n} \sum_{R=1}^{N_r} \tilde{L}(x, x', P_n(R, \rho_R)) \quad (4-4)$$

如果得到一组BRDF的基，我们可以将所有可选的BRDF都表示成这组基的线性组合。如下所示：

$$\rho_R(x, x', x'') = \sum_{i=1}^{N_b} c_{R_i} \tilde{\rho}_i(x, x', x'') \quad (4-5)$$

$\tilde{\rho}_i$ 表示第 i 个基， c_{R_i} 表示物体 R 的BRDF对应于第 i 个基的系数， N_b 是基的数量。根据式4-1，到达 x' 前最后一次反射对最终从 x' 到达 x 的辐射亮度的影响可投影到相同数量的基上：

$$\tilde{L}(x, x', P_n(R, \rho_R)) = \sum_{i=1}^{N_b} c_{R_i} \tilde{L}(x, x', P_n(R, \tilde{\rho}_i)) \quad (4-6)$$

如图4-2，和 P_B 相关的 P_1 包括 $(R_1, \tilde{\rho}_1), \dots, (R_1, \tilde{\rho}_{N_b})$ 。从式4-4的递归形式，

式4-6也可以得到类似的递归形式：

$$L_{n+1}(x, x') = \sum_{P_n} \sum_{R=1}^{N_r} \sum_{i=1}^{N_b} c_{R_i} \tilde{L}(x, x', P_n(R, \tilde{\rho}_i)) \quad (4-7)$$

式4-7对 n 递归可以得到：

$$L_n(x, x') = \sum_{\{R_i\}=1}^{N_r} \sum_{\{\rho_i\}=1}^{N_b} \left(\tilde{L}(x, x', P_n) \prod_{i=1}^n c_{R_i \rho_i} \right) \quad (4-8)$$

用 R' 表示点 x' 所在物体，用 $\rho_{R'}$ 表示相应的BRDF。同样可以将 $\rho_{R'}$ 表示成BRDF的基的线性组合，如下：

$$\rho_{R'} = \sum_{i=1}^{N_b} c_{R'_i} \tilde{\rho}_i \quad (4-9)$$

我们引入如下的表示形式：

$$\tilde{L}'(x, x', P_n, \rho) = \tilde{L}(x, x', P_n) |_{\rho_{R'} = \rho} \quad (4-10)$$

那么将式4-9 和式4-10带入，所在的物体的BRDF对最终从 x' 到达 x 辐射亮度的作用同样可以表示为一组线性组合，系数同样为 x' 所在的物体的BRDF在基上的系数，式4-8可以得到如下形式：

$$L_n(x, x') = \sum_{\rho'=1}^{N_b} c_{R' \rho'} \sum_{\{R_i\}=1}^{N_r} \sum_{\{\rho_i\}=1}^{N_b} \left(\tilde{L}(x, x', P_n, \rho') \prod_{i=1}^n c_{R_i \rho_i} \right) \quad (4-11)$$

如图4-2， P_B 相关的 $\tilde{L}'$ 形式如 $\tilde{L}'(x, x', (R_1, \tilde{\rho}_i), \tilde{\rho}_j)$ ，一共 N_b^2 个。只处理两次反射的效果，式4-3 和式4-11简化为：

$$\tilde{L}(x_e, x') = L_0(x_e, x') + L_1(x_e, x') \quad (4-12)$$

$$L_0(x_e, x') = \sum_{\rho'=1}^{N_b} c_{R' \rho'} \tilde{L}'(x_e, x', P_0, \rho') \quad (4-13)$$

$$L_1(x_e, x') = \sum_{\rho'=1}^{N_b} c_{R'\rho'} \sum_{R_1=1}^{N_r} \sum_{\rho_1=1}^{N_b} \left(\tilde{L}'(x_e, x', P_1, \rho') c_{R_1\rho_1} \right) \quad (4-14)$$

其中 x_e 表示视点位置。 $\tilde{L}(x_e, x')$ 就是我们需要的最终结果。

综上所述，首先确定BRDF的基 $\tilde{\rho}_i$ ，然后预计算所有的 $\tilde{L}'$ ，在绘制时计算选定的BRDF的系数，如式4-5，再根据式4-3和式4-11得到最后的结果。如果不考虑超过两次反射的情况，可从式4-3和式4-11简化为式4-12、式4-13和式4-14。

4.1.2 双向反射分布函数的线性表述和降维

我们将BRDF的参数空间进行离散，每一个BRDF都可以用一个固定长度的向量表示。在可选的BRDF构成的空间中，我们采样足够数量的BRDF，并认为这些样本能够线性表示整个空间。我们使用PCA对这些样本进行降维。后文4.1.5中提供的实验数据中显示，少量的基就能以很小的误差线性表示每一个样本。由于BRDF空间的连续性和我们采样的密集性，这组基对任意一个BRDF的线性组合的误差都很小，如式4-5我们使用符合Blinn-Phong模型的BRDF构成我们的BRDF空间。Blinn-Phong^[7]模型的BRDF表示如下：

$$\rho_{BP} = \frac{K_d}{\pi} + K_s \frac{s+2}{2\pi} \cos^s \delta \quad (4-15)$$

其中 δ 表示入射方向和出射方向的中间方向和反射点法向的夹角， s 表示物体表面的粗糙程度，为正实数， K_d 表示漫反射的反射率， K_s 表示非漫反射的反射率。考虑能量守恒和总反射率非负：

$$\begin{aligned} K_d, K_s &\geq 0 \\ K_d + K_s &\leq 1 \end{aligned} \quad (4-16)$$

Blinn-Phong模型是一种各向同性的BRDF模型，其参数空间是一个三维空间。我们将它定义在如下空间：

$$\begin{aligned} &(\theta_i, \theta_o, \phi) \\ &\theta_i, \theta_o \in [0, \pi/2] \\ &\phi \in [0, \pi] \end{aligned} \quad (4-17)$$

θ_i 表示入射方向和法向的夹角, θ_o 表示出射方向和法向的夹角, ϕ 表示入射方向和出射方向在反射表面上的投影的夹角。不把 ϕ 定义在 $[0, 2\pi]$ 是因为 ρ_{BP} 在 $[0, \pi]$ 和 $[\pi, 2\pi]$ 是对称的。我们把这三个分量在各自的定义域均匀采样, 一个BRDF就能表示为一个三阶的张量, 展开后就得到一个向量。

所有 $K_s = 0$ 的 ρ_{BP} 都是线性相关的, 实际上都是反射率不同的漫反射BRDF。当 $K_d = 0$ 而且 s 不变时, 所有的BRDF都是线性相关的。因此, 我们采样如下: $K_d = 1$ 且 $K_s = 0$; $K_d = 0$ 且 $K_s = 1$, s 在选定的范围内均匀采样 $N_s - 1$ 次。这样得到 N_s 个线性无关的BRDF, 不妨将第一个定为漫反射采样。我们将他们记作:

$$\{\rho_{s1}, \rho_{s2}, \dots, \rho_{sN_s}\} \quad (4-18)$$

漫反射经常占有主要而敏感的视觉效果, 所以漫反射的BRDF采样直接作为一个基。其它 $N_s - 1$ 个BRDF进行PCA得到 $N_b - 1$ 个特征向量。这样总共得到 N_b 个基, 不妨将第一个定为漫反射的基, 我们将它们记为:

$$\{\tilde{\rho}_1, \tilde{\rho}_2, \dots, \tilde{\rho}_{N_b}\} \quad (4-19)$$

并且, 对于所有的BRDF样本, 我们都可以得到式4-5的系数:

$$\rho_{s_i} \approx \sum_{j=1}^{N_b} c_{s_i j} \tilde{\rho}_j \quad (4-20)$$

显然, 由于第一个基是独立选取的漫反射分量, 而后面的基是PCA得到的, 所以:

$$\begin{aligned} \{c_{1j} | j \neq 1\} &= 0 \\ \{c_{i1} | i \neq 1\} &= 0 \end{aligned} \quad (4-21)$$

基的每一项都是一个未必符合物理意义的BRDF, 可能有负值或者反射率超过1, 但同样可以作为物体的材质计算全局光照明的效果, 这并不影响我们的预计算和最后绘制的结果。那么对于某一个物体 R 的形如式4-15的BRDF, ρ_R 只要它的 s 在采样的范围内, 就能通过线性插值 $\{c_{ij} | i, j \neq 1\}$ 得到类似于式4-5的表述形

式:

$$\rho_R \approx K_d \tilde{\rho}_1 + K_s \sum_{i=2}^{N_b} \tilde{c}_{Ri} \tilde{\rho}_i \quad (4-22)$$

4.1.3 预计算

获得形如式4-19的BRDF的基之后, 场景中所有材质可变的物体都从这些基中选择BRDF, 我们预计算这些基的各种分布组合所对应的光照效果就能得到式4-13和式4-14中的 $\tilde{L}'(x_e, x', P_0, \rho')$ 和 $\tilde{L}'(x_e, x', P_1, \rho')$ 了。我们采用Kajiya^[47]的路径跟踪完成预计算, 在需要计算BRDF时, 根据 $(\theta_i, \theta_o, \phi)$, 通过三线性插值相应的 $\tilde{\rho}$ 得到。

我们从视点向每一个绘制像素发射一根光线, 若与场景相交, 就测试交点和所有光源采样点是否可见, 若可见, 就根据交点的BRDF计算出射的辐射亮度, 并根据交点的BRDF的基加到相应的 $\tilde{L}'(x_e, x', P_0, \rho')$ 。然后在该交点的法向半球内均匀散射光线, 如果和场景相交, 就按照相同的方法计算沿着这些散射光线出射的直接光照形成的辐射亮度, 进一步计算它所贡献的经过两次反射到达视点的辐射亮度, 并且根据散射光线的交点所在的物体及先后两个交点所采用的BRDF的基加到相应的 $\tilde{L}'(x_e, x', P_1, \rho')$ 。根据式4-11, 在绘制图像分辨率、散射光线的数量和光源采样点的数量都相同的情况下, 预计算和绘制的时间和空间的代价取决于BRDF的基的数量以及场景中的材质可变的物体的数量。此外, 我们认为场景中有一个物体 $\hat{R}$, 这个物体的材质不变。在BRDF的基中有一个 $\hat{\rho}$, 是一个空间变化的BRDF, 对应于物体 $\hat{R}$ 。那么, 式4-5中的 c_{Ri} 满足:

$$\begin{aligned} c_{Ri} \Big| \left(R = \hat{R}; \rho_i = \hat{\rho} \right) &= 1 \\ c_{Ri} \Big| \left(R = \hat{R}; \rho_i \neq \hat{\rho} \right) &= 0 \\ c_{Ri} \Big| \left(R \neq \hat{R}; \rho_i = \hat{\rho} \right) &= 0 \end{aligned} \quad (4-23)$$

并且, 对于式4-2, 形如 $(\hat{R}, \rho) | \rho \neq \hat{\rho}$ 或是 $(R, \hat{\rho}) | R \neq \hat{R}$ 的路径是不存在的。因此, 我们定义如下运算:

$$B(R) = \begin{cases} \{\tilde{\rho}_i | \tilde{\rho}_i \neq \hat{\rho}\}, & R \neq \hat{R} \\ \{\hat{\rho}\}, & R = \hat{R} \end{cases} \quad (4-24)$$

那么将该材质不变的物体引入式4-11就得到如下形式:

$$L_n(x, x') = \sum_{\rho' \in B(R')} c_{R'\rho'} \sum_{\{R_i\}=1}^{N_r} \sum_{\{\rho_i\} \in B(\rho_i)} \left(\tilde{L}'(x, x', P_n, \rho') \prod_{i=1}^n c_{R'\rho_i} \right) \quad (4-25)$$

同样, 式4-13 和式4-14变换为如下形式:

$$L_0(x_e, x') = \sum_{\rho' \in B(R')} c_{R'\rho'} \tilde{L}'(x_e, x', P_0, \rho') \quad (4-26)$$

$$L_1(x_e, x') = \sum_{\rho' \in B(R')} c_{R'\rho'} \sum_{R_1=1}^{N_r} \sum_{\rho_1 \in B(\rho_1)} \left(\tilde{L}'(x_e, x', P_1, \rho') c_{R_1\rho_1} \right) \quad (4-27)$$

我们认为纹理只作用于BRDF的漫反射部分。材质可变部分只有第一个基, 即漫反射的BRDF, 受到纹理的作用。材质不变的部分预计算的结果把漫反射和非漫反射部分合在一起, 所以也需要考虑纹理的作用。我们采用RGB三通道的纹理, 因此, 对于所有预计算的结果, 如果反射的路径中包含了材质固定的物体, 或者经过的材质可变物体采用了漫反射的BRDF, 预计算时需要考虑纹理的作用, 生成的结果包含RGB三个通道。不失一般性, 我们依然如上一节假设 $\tilde{\rho}_1$ 为漫反射BRDF的基, 显然 $R_1 \neq \hat{R}$ 且 $\rho_1 \neq \hat{\rho}$, 并且定义:

$$\begin{aligned} \bar{R} &\in \{R_i\} - \{R_1, \hat{R}\} \\ \bar{\rho} &\in \{\rho_i\} - \{\rho_1, \hat{\rho}\} \end{aligned} \quad (4-28)$$

综上所述, 场景中如果包含材质不变物体, 并贴有RGB三通道纹理, 根据式4-26 和式4-27, 预计算的结果如表4-1。预计算的结果采用一组和最终绘制结果同样分辨率的高动态范围(high dynamic range, HDR)图像来存储, 每张图像中的一个像素代表相应的点 x' 在表4-1中某一项的一个值。表4-1可得, 共需 $(N_r - 1)(N_b - 2)^2 + (N_b - 2)$ 张单通道图像, $2N_r N_b - 3N_r + 3$ 张三通道图像。此外还需要一张同样分辨率的场景材质分布映射图像。

路径积分算法产生散射光线的方向时加入随机扰动, 可以避免和场景结构相关的走样, 但会引入噪点。如果噪点在不同的图像之间相互独立, 我们绘制时线性组合多幅图像的系数将会放大这些噪点, 严重影响最终效果。对此, 我们只做一次路径跟踪, 将表4-1每一项相应的BRDF的分布依次代入, 生成所有的预计算

表 4-1 预计算的数据

$P_n(R', \rho_{R'})$	颜色空间	预计算得到的HDR图像的数量
$(\hat{R}, \hat{\rho})$	RGB	1
$(\hat{R}, \hat{\rho}) (\hat{R}, \hat{\rho})$	RGB	1
$(\bar{R}, \rho_1) (\hat{R}, \hat{\rho})$	RGB	$N_r - 1$
$(\bar{R}, \bar{\rho}) (\hat{R}, \hat{\rho})$	RGB	$(N_r - 1) (N_b - 2)$
$(\bar{R}, \rho_1)$	RGB	1
$(\bar{R}, \bar{\rho})$	Gray	$N_b - 2$
$(\hat{R}, \hat{\rho}) (\bar{R}, \rho_1)$	RGB	1
$(\bar{R}, \rho_1) (\bar{R}, \rho_1)$	RGB	$N_r - 1$
$(\bar{R}, \bar{\rho}) (\bar{R}, \rho_1)$	RGB	$(N_r - 1) (N_b - 2)$
$(\hat{R}, \hat{\rho}) (\bar{R}, \bar{\rho})$	RGB	$N_b - 2$
$(\bar{R}, \rho_1) (\bar{R}, \bar{\rho})$	RGB	$(N_r - 1) (N_b - 2)$
$(\bar{R}, \bar{\rho}) (\bar{R}, \bar{\rho})$	Gray	$(N_r - 1) (N_b - 2)^2$

图像。这样就能避免绘制时放大噪点，还使得预计算的时间和基及场景中物体的数量无关，只和图像分辨率、散射光线和光源的采样点的数量相关。因为在我们的路径跟踪过程，主要运算代价在于光线和场景的求交计算。由于BRDF的各个基的反射叶(reflectance lobe)的分布差异很大，因此我们不能对BRDF做重要性采样(important sampling)。这也正是我们采用路径跟踪而不是分布式光线跟踪(distributed ray tracing)的原因。

4.1.4 绘制

根据用户实时指定的场景中BRDF的分布，我们可以将各物体的BRDF表述为基的线性组合。然后根据式4-12 式4-26 式4-27 线性组合预计算的图像，就得到最终绘制结果。为能达到实时的绘制速度，我们使用GPU实现最后线性组合预计算图像的操作。我们使用多遍绘制(multi-pass rendering)，在每一遍绘制中将一张预计算的图像和材质物体分布映射图像作为纹理传入GPU，并用各物体的系数乘以相应的像素，经过多遍之后对所有的预计算图像都进行了同样的操作。每一遍的结果通过半透明混合的方法叠加在一起，就能得到最后的结果。如果每张预计算图像都要绘制一遍，那么由于绘制的遍数太多，性能会受到很大影响。所以每遍绘制可

以组合多张纹理，而不是只操作一张纹理，这样可以大量减少绘制的遍数，从而可以得到实时的绘制速度。

4.1.5 实验结果

我们用VC++2005和DirectX9在Windows XP下实现以上算法，运行环境是3.2 GHz 的Pentium 4 PC，1 Gbytes 内存，Nvidia GeForce 6800 GT 显示卡。我们对 $(\theta_i, \theta_o, \phi)$ 采用 $30 \times 30 \times 60$ 的采样分辨率， $s \in [1, 200]$ ， $N_s = 201$ ， $N_b = 6$ 个基。采样BRDF的重建的 L^2 范数的误差为0.7979%，其中最大的单个样本的 L^2 范数的误差是6.747%。实验的场景包含两个材质可变的物体，绘制分辨率为 512×512 。预计算的结果包括55张单通道的HDR图像，29张RGB三通道的HDR图像和1张场景材质分布映射图像，我们使用pfm(portable float map)的格式进行存储，总共需要140 Mbytes，绘制性能为67 fps。一些实验数据有参考图像。参考图像的绘制同样采用路径跟踪的方法得到，而且光源采样数量、散射光线的数量、物体BRDF的参数都和被比较的数据相同，但是BRDF采用原始数据，而不是被压缩之后的。在表4-2中记录了文中提供的三个场景的规模和相应的预处理的数据。

如图4-3所示的场景的光源是顶部的一个面光源。图4.3(a)和图4.3(e) 是直接光照的效果，天花板和Bunny造成的阴影区域是全黑的。图4.3(c)和图4.3(g) 包含了直接光照和间接光照的效果，天花板和Bunny造成的阴影区域被反射光照亮，而且Bunny和背后墙面由于受到左右两边的墙壁的反射形成蓝色和红色的渗色。图4.3(a)至图4.3(d)，场景材质漫反射明显；图4.3(e)至图4.3(h)，场景材质高光明显。比较图4.3(c)和图4.3(g)：在红框区域内，Bunny蓝色的渗色效果，图4.3(g)明显强烈许多；在绿框区域内，图4.3(g)有明显的Bunny的倒影；在蓝框区域内，图4.3(g)可以看到右侧红墙和白色地面之间的边界在后面的墙上的清晰的投影。

如图4-4所示的场景采用点光源以凸现焦散的效果，但是我们的算法也可以使用复杂光源。我们知道，反射物体的材质高光越强，焦散越明显；接受反射光的

表 4-2 所示的基于像素的空间静态可变材质的实验结果的预计算的性能

	场景的面片数量	每个像素的采样光线数量	光源的采样数量		预计算时间 (小时)
			直接光照	间接光照	
Ring	579	4096	1	1	1.65
Bunny	69487	4096	98	8	30.08
Gargoyle	45678	4096	98	8	22.75

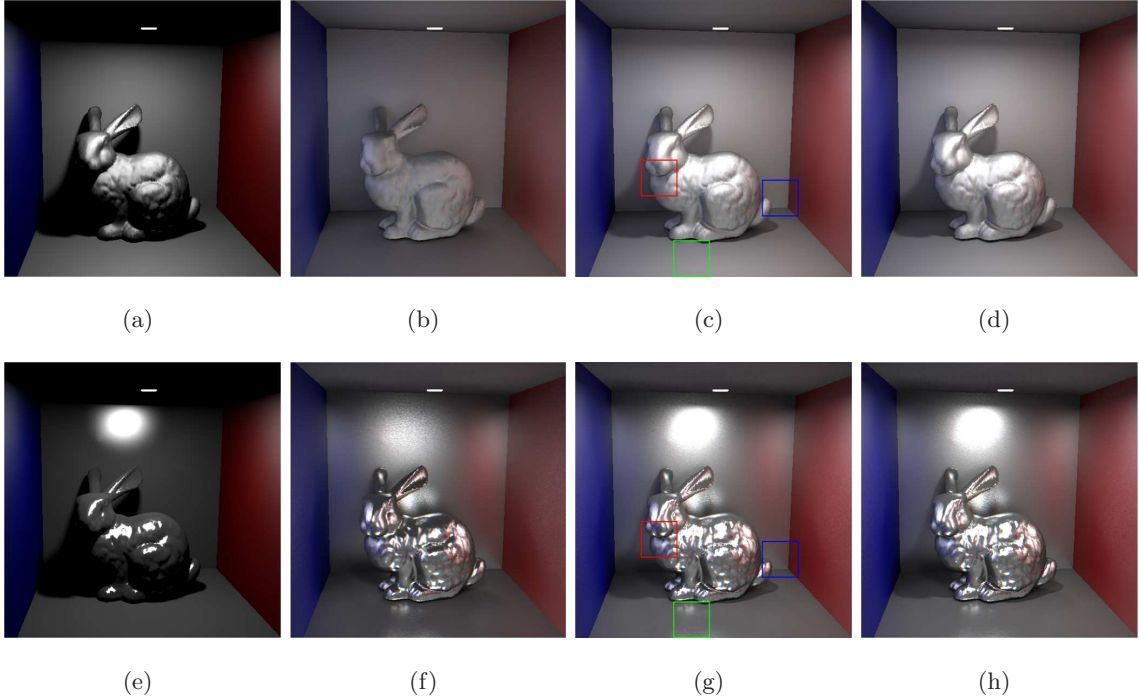


图 4-3 Bunny和白色墙壁(包括地面和天花板)的BRDF是可编辑的。Bunny $K_d = 0.2$ $K_s = 0.4$, 白色墙壁 $K_d = 0.3$ $K_s = 0.3$ 。从(a)到(d): Bunny $n_s = 5$, 白色墙壁 $n_s = 5$, L^2 范数的误差为0.0150%; 从(e)到(h): Bunny $n_s = 180$, 白色墙壁 $n_s = 180$, L^2 范数的误差为0.0021%。(a)(e): 直接光照的效果; (b)(f): 间接光照的效果; (c)(g): 最终绘制效果; (d)(h): 参考图像。

物体的材质高光越强, 反射物体形成的倒影越明显。如图4.4(a), 圆环和平面的材质漫反射明显, 只是显现出渗色效果; 如图4.4(b), 圆环漫反射明显, 平面高光明显, 圆环投影到平面上的倒影圆形边界十分清晰; 如图4.4(c), 圆环高光明显, 平面漫反射明显, 可见圆环内部投到平面上的焦散: 两条圆弧线及连接处的亮斑; 如图4.4(d), 圆环和平面高光明显, 焦散效果相比于图4.4(c)焦散的形状和分布都有明显不同。其中, 图4.4(a)至图4.4(d)是我们实时绘制的结果, 图4.4(e)至图4.4(h)是参考图像。

如图4-5所示的场景中可看到多个物体反射和遮挡造成的全局光照效果。光源在图像中不可见, 是顶部靠近视点处的一个面光源。Gargoyle的材质从图4.5(a)至图4.5(d), 从类似于石膏的漫反射材质, 变为接近于塑料的材质, 再变为金属感强烈的高光材质。纹理只作用于BRDF漫反射部分, 因此飞机上面的纹理, 在漫反射强烈时十分明显, 随着高光的增强, 纹理就随之变弱至不可见。飞机、Gargoyle、圆环面之间的互相影响, 除了渗色之外, 最明显的是图4.5(d)中,

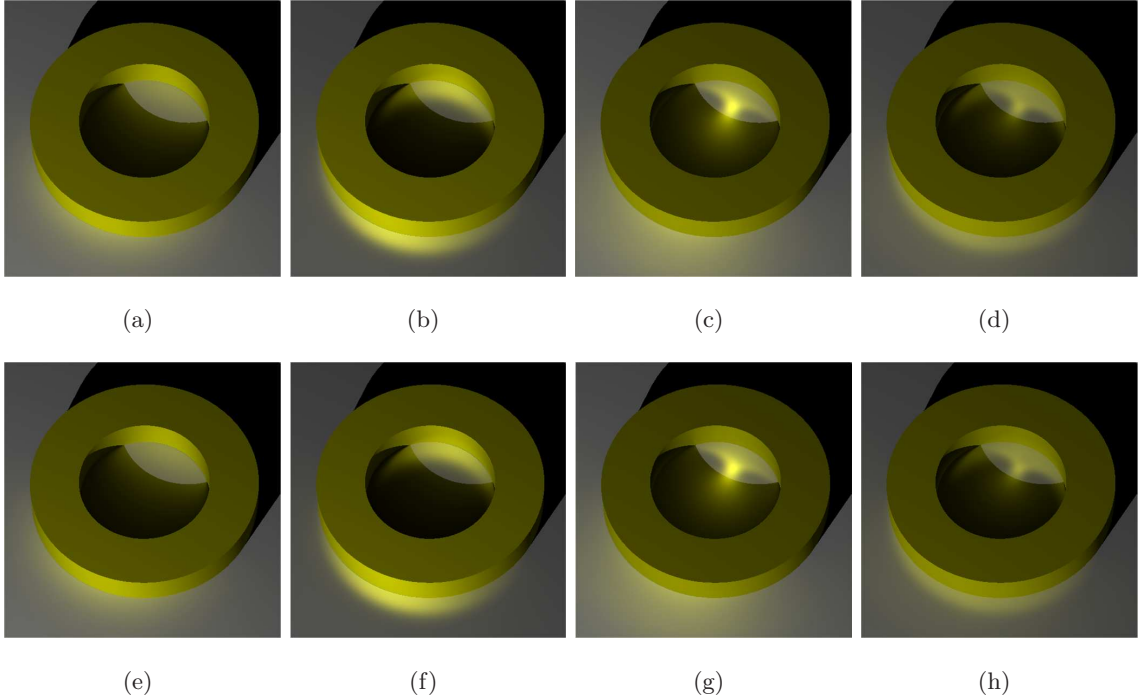


图 4-4 圆环和平面的BRDF是可编辑的。其中圆环是黄色的，不包括蓝色通道的颜色分量。圆环 $K_d = 0.2$ $K_s = 0.2$ ，平面 $K_d = 0.2$ $K_s = 0.2$ 。从(a)向(d)：我们的结果；从(e)向(h)：相应的参考图像。(a)(e)：圆环 $n_s = 5$ ，平面 $n_s = 5$ ， L^2 范数的误差0.2698%；(b)(f)：圆环 $n_s = 5$ ，平面 $n_s = 180$ ， L^2 范数的误差0.3114%；(c)(g)：圆环 $n_s = 180$ ，平面 $n_s = 5$ ， L^2 范数的误差0.1624%；(d)(h)：圆环 $n_s = 180$ ，平面 $n_s = 180$ ， L^2 范数的误差0.2640%。

圆环面在Gargoyle的身体和底座上形成了一个环形的倒影。此外，图4.5(d)中飞机反射到后面墙面的蓝色以及图4.5(c)中Gargoyle的翅膀反射到右边墙面的红色都十分明显。

4.1.6 结论

在4.1的工作中，我们提供了一个材质可变的全局光照的预计算和实时绘制的框架。主要的思想是将全局照明问题分解为一组与材质线性相关的问题的叠加。在这个框架下，我们将BRDF空间离散采样，并使用主元分析法得到较少数量的基。所有的预计算都在这组基上进行：场景的材质采用BRDF的基，并用路径跟踪得到各种组合下的绘制结果。最后根据用户指定的BRDF，使用GPU线性组合预计算得到的图像，从而达到实时的绘制速度。

我们的算法中，预计算的存储空间和绘制时的性能都和场景中可变材质物体

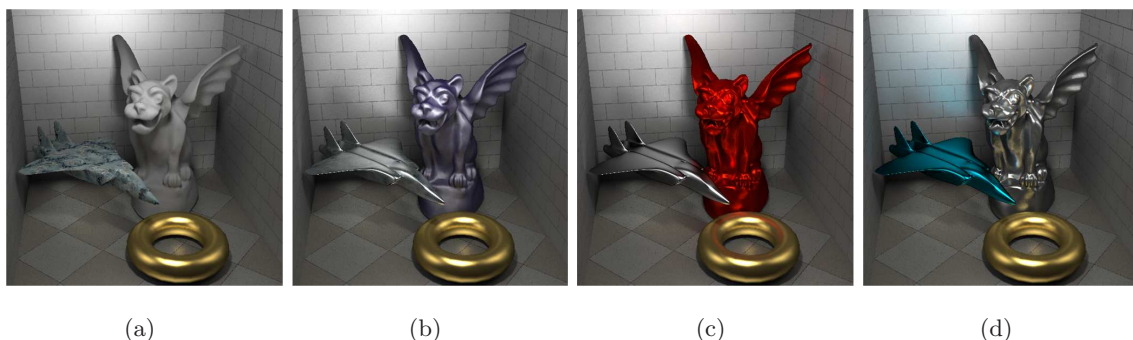


图 4-5 Gargoyle和飞机的材质是可编辑的。四周墙壁和飞机都贴上了纹理。

的数量和BRDF空间的基的数量是密切相关的。我们选取Blinn-Phong的BRDF模型, 其中 $s \in [1, 200]$, 从纯漫反射到高光十分强烈的材质, 可以覆盖很大的范围, 结果中也能看到明显的对比效果。误差分析中可知6个基(包括漫反射项)已经可以达到足够高的精度, 并且最终绘制效果和参考图像之间肉眼基本无法辨别差异。我们的实验中采用了两个可变材质物体, 其它都是材质不变的物体, 这样的配置对于许多材质编辑的应用已经足够, 因为一个场景中对全局光照效果有明显影响的物体数量十分有限。但是, 如果大量引入镜面或者接近镜面的材质, 那么BRDF空间需要更多的基来表示, 或者采用更多数量的可变材质的物体, 这都会大大增加预计算的存储空间, 降低绘制性能。这些问题的解决需要进一步的工作。

4.2 空间动态可变材质的交互级全局光照绘制

一种简单的可变材质是空间静态可变材质, 即一个物体各个部分的材质完全相同, 而且经过用户修改后也是如此, 目前的可变材质的实时或交互级全局光照绘制算法都是在这样的假设下实现的。如果一个物体的各部分的材质可能不同, 并且用户可以对不同部分的材质做不同修改, 我们称之为空间动态可变材质。相比于空间静态可变材质, 空间动态可变材质对光传播的影响更加复杂, 目前尚没有任何支持空间动态可变材质的交互级全局光照绘制的算法。

现实世界中, 大多数物体都被不同的材质覆盖不同的部分, 因此空间动态可变材质相对于空间静态可变材质更为通用, 而且能带来更加丰富的视觉效果。如图4-6所示的四张图都是全局光照的绘制效果: 图4.6(a)和图4.6(b)主要体现了不同颜色物体之间的渗色效果, 图4.6(c)和图4.6(d)主要体现了高光材质的圆环反射到平面上的焦散效果。图4.6(a)和图4.6(c)是空间静态可变材质, 在同一个物体上

的材质处处相同，图4.6(b)和图4.6(d) 是空间动态可变材质，一个物体上的不同部分被不同的材质覆盖。可以明显看到空间动态可变材质能体现更丰富的细节特征和高频的材质效果。

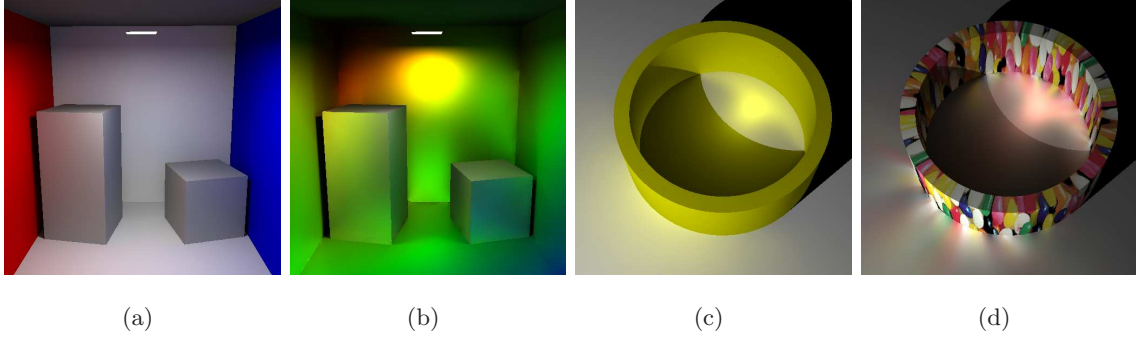


图 4-6 空间静态和空间动态材质的全局光照照明绘制。(a)(c)空间静态材质；(b)(d)空间动态材质。

在本章节中，我们致力于为空间动态可变材质交互级生成全局光照的绘制效果。具体描述为：采用和4.1相同的设置，在静态场景中采用复杂光源，并且在空间动态材质变化时能交互级绘制全局光照效果。我们采用Blinn-Phong^[7]模型的BRDF描述物体的材质。

Sloan^[84]提出的预计算辐射传输和Zhou^[108]提出的阴影场的方法，将光照或遮挡反射等因素的影响分解到线性基上，将积分运算转化为基的系数的操作，实现了动态光源下的实时绘制。但是，物体材质和全局光照照明绘制效果之间不存在这样的线性关系，因此Ben-Artzi^[6]只完成了直接光照下可变材质的实时绘制。为了引入间接光照，Ben-Artzi^[5]采用多项式描述全局光照的辐射传输，并基于此进行预计算和绘制，前文4.1的方法是将辐射强度根据传输路径进行分类，从而建立起每一类辐射传输和各自相关材质的线性关系，实现了空间静态可变材质的全局光照照明绘制。我们继续沿用前文4.1的思路，但是关键在于：引入空间动态材质后，传输路径的分类的数量变得极其巨大，不可能达到实时或交互级的绘制性能。这就是本章节所需要解决的关键问题。

我们采用的算法具体描述为：将空间动态可变材质物体近似模拟成许多空间静态可变材质的子区域。然后，采用前文4.1的方法进行预计算。由于子区域数量多，这样得到的预计算结果数据量巨大，不能直接用于绘制。为了减少数据量，我们不考虑BRDF的高光部分的相互反射，但是我们计算漫反射之间，以及漫反射和高光之间的相互作用；类似的假设普遍用于重光照技术中，比如Hasan^[38]假设只考

虑漫反射的反射作用，不同的是我们的假设中还保留了从高光到漫反射的间接光照，所以我们能够实现焦散等高频的间接光照的绘制效果。我们发现，预计算结果具有很大的相关性和稀疏性，因此我们采用PCA对不同光路的数据进行压缩，然后对得到的结果再使用Haar小波进行压缩。这样最后得到的结果有合理的数据量，能达到交互级的绘制性能，还能保证高质量的绘制效果。

综上所述，我们的创新主要有两点：一是采用划分子区域的方法来模拟空间动态可变材质；二是利用预计算辐射传输数据的相关性和稀疏性进行压缩，从而达到交互级绘制的性能。

我们的方法可在场景建模时为空间动态材质的选择和编辑提供全局光照的反馈效果。其中最为典型的应用是纹理贴图的选择和编辑，而目前大多数工具只能提供局部光照的反馈效果。

4.2.1 算法概述

我们同样采用式4-1的绘制方程的数学形式。

在式4-1的描述中，出射的辐射亮度可能经过场景不确定次数的反射；为了在出射的辐射亮度和物体材质之间建立线性的映射关系，我们需要将光的传播路径引入式4-1，而为了支持空间动态可变材质，我们不需要每个物体的材质保持一致，因此我们将每一个物体 R 分为 n_R 个内部材质一致的子区域来近似模拟：

$$\begin{aligned} L(x, x') &= \sum_{n=0}^{\infty} L_n(x, x') \\ L_{n+1}(x, x') &= \sum_{P_n} \sum_{R=1}^{N_r} \sum_{r=1}^{n_R} \tilde{L}(x, x', P_n(R, r, \rho_{Rr})) \end{aligned} \quad (4-29)$$

N_r 表示场景被划分的物体数量， (R, r, ρ_{Rr}) 表示一个物体、子区域及其BRDF， ρ_{Rr} 表示物体 R 的子区域 r 的BRDF。如图4-7，场景中共有两个空间动态可变材质物体，每个又被划分为4个子区域，图中标出的6条光路，根据图示的颜色分为三组，任意一组在前文4.1空间静态可变材质的算法中被同样的物体反射，因此属于同一个光路，一起计算；但是为模拟空间动态材质，必须将不同子区域反射的光路独立计算。其 P_n 表示从光源出射的光线经过 n 次反射的过程： $P_n = (R_1, r_1, \rho_{R_1 r_1})(R_2, r_2, \rho_{R_2 r_2}) \cdots (R_n, r_n, \rho_{R_n r_n})$ 。 $L_n(x, x')$ 表示点 x' 射向点 x 的辐射亮度，并且光在到达点 x' 前经过了 n 次反射，但对反射路径没有约束。 $\tilde{L}(x, x', P_n)$ 表示点 x' 射向点 x 的辐射亮度，而且光在到达点 x' 前按 P_n 所描述的

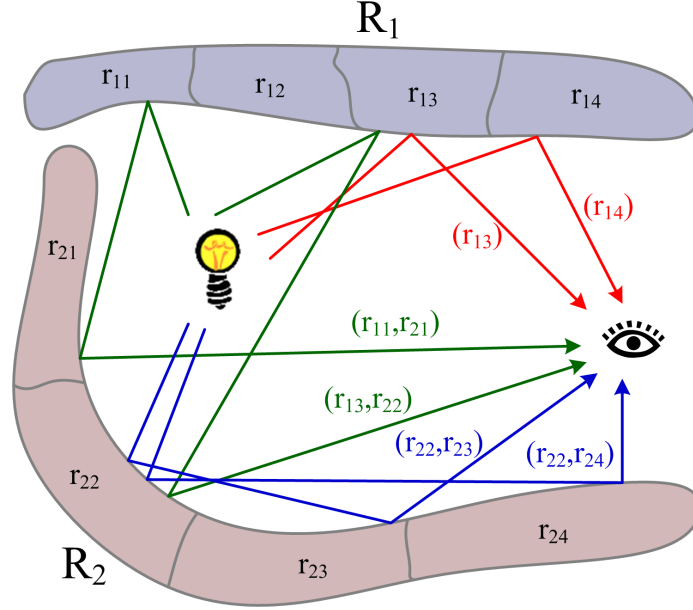


图 4-7 光的传播路径。红色：经过 (R_1) 的直接光照。绿色：经过 (R_1, R_2) 的间接光照。蓝色：经过 (R_2, R_2) 的间接光照。

路径经过了 n 次反射。显然 $\tilde{L}(x, x', P_n)$ 和 P_n 包含的材质的乘积线性相关。我们采用和前文4.1.2相同的方法得到BRDF空间的一组基，将每个子区域的BRDF都表示成这组基的线性组合：

$$\rho_{R_r}(x, x', x'') = \sum_{i=1}^{N_b} c_{R_r i} \tilde{\rho}_i(x, x', x'') \quad (4-30)$$

$\tilde{\rho}_i$ 表示第 i 个BRDF基， $c_{R_r i}$ 表示 ρ_{R_r} 对应于第 i 个基的系数， N_b 是基的数量。那么式4-29可以得到：

$$L_{n+1}(x, x') = \sum_{P_n} \sum_{R=1}^{N_r} \sum_{r=1}^{n_R} \sum_{i=1}^{N_b} c_{R_r i} \tilde{L}(x, x', P_n(R, r, \tilde{\rho}_i)) \quad (4-31)$$

式4-31递归可得：

$$L_n(x, x') = \sum_{\{R_i\}=1}^{N_r} \sum_{\{r_i\}=1}^{n_{R_i}} \sum_{\{\rho_i\}=1}^{N_b} \left(\tilde{L}(x, x', P_n) \prod_{i=1}^n c_{R_i r_i \rho_i} \right) \quad (4-32)$$

subject to $\rho_{R_i r_i} = \tilde{\rho}_{\rho_i}$

我们引入如下的表示形式：

$$\tilde{L}'(x, x', P_n, \rho) = \tilde{L}(x, x', P_n) \text{ if } \rho_{R'r'} = \rho \quad (4-33)$$

其中 $\rho_{R'r'}$ 表示反射点 x' 所在的子区域的BRDF。将式4-33代入式4-32可表示为：

$$L_n(x, x') = \sum_{\rho'=1}^{N_b} c_{R'r'\rho'} \sum_{\{R_i\}=1}^{N_r} \sum_{\{r_i\}=1}^{n_{R_i}} \sum_{\{\rho_i\}=1}^{N_b} \left(\tilde{L}'(x, x', P_n, \rho') \prod_{i=1}^n c_{R_i r_i \rho_i} \right) \quad (4-34)$$

subject to $\rho_{R_i r_i} = \tilde{\rho}_{\rho_i}$

在我们的实现中，只处理两次反射的效果，因此式4-29和式4-34可简化得到：

$$\begin{aligned} \tilde{L}(x_e, x') &= L_0(x_e, x') + L_1(x_e, x') \\ L_0(x_e, x') &= \sum_{\rho'=1}^{N_b} c_{R'r'\rho'} \tilde{L}'(x_e, x', P_0, \rho') \\ L_1(x_e, x') &= \sum_{\rho'=1}^{N_b} c_{R'r'\rho'} \sum_{R_1=1}^{N_r} \sum_{r_1=1}^{n_{R_1}} \sum_{\rho_1=1}^{N_b} \left(\tilde{L}'(x_e, x', P_1, \rho') c_{R_1 r_1 \rho_1} \right) \end{aligned} \quad (4-35)$$

subject to $\rho_{R_1 r_1} = \tilde{\rho}_{\rho_1}$

其中 x_e 表示视点位置。 $\tilde{L}(x_e, x')$ 就是我们需要的最终结果。

我们为了减少预计算的数据量，我们不考虑BRDF非漫反射部分的互相影响。和前文4.1.2使用同样的假设 $\tilde{\rho}_1$ 表示漫反射BRDF的基，其他的都是非漫反射BRDF的基，因此在式4-35中，我们进一步假设：

$$\tilde{L}'(x_e, x', P_1, \rho') = 0 \text{ if } \rho' \neq \tilde{\rho}_1 \text{ and } \rho_{R_1 r_1} \neq \tilde{\rho}_1 \quad (4-36)$$

类似的假设普遍用于重光照技术中，比如Hasan^[38]假设所有的反射都是漫反射；和Hasan^[38]不同的是我们的假设中还保留了从非漫反射到漫反射的间接光照，所以我们能够保留焦散等高频的间接光照的绘制效果。

以上数学描述中，我们如果把每个子区域都看作一个独立的物体，那么实际上可以等价于前文4.1的方法。但是，前文4.1的方法的方法中并没有对 $\tilde{L}'$ 进行压缩。我们可以看到，由于我们引入了子区域，因此 $\tilde{L}'$ 数量急剧增加，不进行合理的压缩是无法进行绘制的。我们将子区域作为独立于物体的变量引入，是因为我们对于不

同的物体所对应的 $\tilde{L}'$ 分别进行压缩，而不是将所有的 $\tilde{L}'$ 一起压缩。如图4-7，相同颜色的光路是被同样的物体反射得到，其所包含的不同子区域对应的光路的预计算的结果被作为一组进行压缩。我们采用这样的压缩策略主要有四个原因：一般情况下物体的划分有实际意义，比如室内的地板和墙壁，但是子区域没有，因此我们可以认为一个物体内部的子区域相关性更强，更利于压缩；不同物体之间的材质差异明显，同一物体内部的材质许多时候体现相似的特点，如果在一起压缩，可能会导致两种差异巨大的材质互相污染串色；将数据分为多个组，各自压缩到数量较少的基，相比于所有数据压缩到一组数量较大的基，在绘制效果相当的情况下，前者的绘制性能更为优越；对于不同的物体，可以根据实际的视觉效果敏感程度采用不同的压缩率。

4.2.2 预计算及数据压缩

场景物体的划分由用户指定。对于每一个物体，我们先赋予形变尽量小的纹理坐标，并在纹理坐标空间中均匀划分，然后对应到原始的几何上形成子区域。我们采用DirectX的UVAtlas工具赋予纹理坐标，得到良好的效果。实际上子区域的划分结果对于效果而言是十分重要的，如果划分得过于粗糙，就会丢失过多的材质变化的细节，划分得过于细致，就会给绘制带来巨大的代价。显而易见，子区域划分越细密，预计算结果的压缩空间就越大。因此我们采用高密划分，高度压缩的方法首先将场景划分成尽量小的子区域，以获取尽量多的材质的空间动态的效果，然后在压缩时可以大量抛弃不敏感或者暗淡的部分，来平衡绘制效果和绘制性能。

我们采用和前文4.1一样的方法可以得到所有的 $\tilde{L}'$ 。所有可见的点 x' 就是对应绘制结果中的每个像素，因此可以将 $\tilde{L}'$ 存储成高动态范围(high dynamic range, HDR)图像：将 $\tilde{L}'(x_e, x', P_0, \rho')$ 存储成 $I(P_0, \rho')$ ，将 $\tilde{L}'(x_e, x', P_1, \rho')$ 存储成 $I(P_1, \rho')$ 。此外，由于最终绘制的要求，还需要同样分辨率的场景材质分布映射图像，不但记录每个像素所在的物体和子区域，而且记录每个像素的纹理坐标。

得到 I 之后，需要对其进行压缩。我们保留所有的 $I(P_0, \rho')$ ，只对 $I(P_1, \rho')$ 进行压缩。这是因为 $I(P_0, \rho')$ 用于直接光照的结果，体现了大量的高频细节，压缩很容易影响绘制效果，而且 $I(P_0, \rho')$ 的数据量并不大，与物体和子区域的数量无关；相反 $I(P_1, \rho')$ 用于间接光照，较为低频，而且数据量巨大，需要进行压缩。根据以上理由，我们对不同物体的 $I(P_1, \rho')$ 分别进行压缩。事实上 $I(P_1, \rho')$ 有很明显的稀疏性和相关性：稀疏性是因为我们将子区域划分得较小，互相可见的子区域数量

较少, 而且许多子区域由于和光源可见程度低, 或者距离可见的其他子区域距离远, 对全局光照明的贡献就很弱了; 相关性是因为辐射强度经过的反射次数越多就越低频^[24], 同一物体的子区域互相之间材质较为接近, 空间位置相似。这种场景内光照传输的稀疏性和相关性被广泛应用于全局光照明的算法中^[24, 38]。一个物体 R_1 对应的 $I(P_1, \rho')$ 需要进一步根据 ρ' 的不同划分为不同的组分别压缩, 这是因为不同的 ρ' 的图像的亮度有很大的区别, 如果 ρ' 对应于BRDF的基的高频部分, 图像的亮度都很弱, 但是对于绘制效果而言却十分敏感而重要, 如果放在一起压缩, 会丢失这些高频的绘制效果。

那么一个 $(R_1, \tilde{\rho})$ 对应的 $I(P_1, \rho')$ 可以看作是一个三维数据: 三个维度包括 r_1 和图像的宽度和高度。我们要对这三个维度都进行压缩: 我们对这些 $I(P_1, \rho')$ 图像在维度 r_1 使用PCA进行压缩, 得到一组基; 然后对这组基中的每一个基图像用Haar小波进行压缩。我们采用PCA压缩所有的 $I(P_1, \rho')$ 而不是采用Haar小波有三个原因: 在较高的压缩率下, Haar会引入高频误差, 严重影响绘制效果, 而PCA引入的误差较为低频, 视觉效果不敏感; 在图像空间的两个维度上由于需要保持高频细节效果, 所以压缩率受到限制, 因此在维度 r_1 上压缩时需要采用较高的压缩率; 在维度 r_1 上进行PCA压缩是一种非常有效的抑制图像噪点的方法^[58]。

在进行Haar小波压缩之前, 我们需要对每个基图像进行中值滤波。我们在进行预计算时, 会引入路径跟踪固有的噪点, 而噪点对Haar小波压缩的影响非常大。这些噪点相对于最终的绘制结果而言是不明显的, 但是当它们分别作为预计算的每个数据存储时, 却明显许多, 基于和前文4.1.3同样的理由, 我们不能对路径跟踪进行重要性采样^[12], 因此通过提高采样光线数量的方法减弱噪点的代价就会非常大。对图像进行PCA压缩本身就具有降噪的作用^[58], 但是效果并不足以保证Haar小波压缩时保留主要信息而不是保留大量的噪点信息。中值滤波是一个行之有效的降噪方法^[53], 我们需要注意的是, 在中值滤波过程中, 不同的物体互相对比明显, 因此不能将不同物体像素放在一起滤波, 否则会造成明显的串色而影响绘制效果; 但是同一物体的不同子区域之间却必须一起滤波, 否则绘制结果中子区域的交界处会出现明显的不连续。这样经过滤波的基图像进行Haar小波压缩时能在较高的压缩率下依然保留良好的绘制效果。

4.2.3 材质编辑和绘制

我们同样采用如式4-15的Blinn-Phong的BRDF模型^[7]。

如前文所述, 对于每个空间动态可变材质物体, 我们都赋予了纹理坐标。因此, 我们可以用纹理的方式给场景进行材质编辑。我们对每个空间动态可变材质物体赋予三张三通道的HDR图像作为材质纹理, 分别表示RGB三通道的 K_d , K_s 和 s , 该物体的每个点的材质都可以根据其纹理坐标在这三张图像上取得相应的值。通过编辑修改相应的纹理, 我们就能完成材质编辑。

我们根据式4-35, 使用 $I(P_0, \rho')$ 和压缩后的 $I(P_1, \rho')$ 进行绘制。对于点 x' 的材质 ρ' , 我们根据预计算时得到场景材质分布映射图像中的纹理坐标, 索引材质纹理得到该点准确的BRDF参数。对于 P_1 中的 $\rho_{R_1 r_1}$, 我们通过对材质纹理的降采样得到, 降采样的分辨率取决于在纹理坐标空间划分子区域的分辨率。

由于参与绘制的数据量太大, 难以放入显存, 因此我们采用CPU进行绘制, 而没有采用GPU进行加速。 $I(P_0, \rho')$ 由于不需要压缩, 因此绘制方法和前文4.1.4相类似, 所不同的是我们采用CPU绘制, 并且对每一个像素都对应不同的材质系数, 而不像前文4.1.4中同一物体的像素具有相同的系数。 $I(P_1, \rho')$ 以压缩之后的小波系数的形式存在, 因此我们在计算组合 $I(P_1, \rho')$ 时也是采用在小波空间中计算, 并将最后的得到的小波系数投影到图像空间, 并与 $I(P_0, \rho')$ 的结果相加, 就得到了最后的绘制结果。绘制性能的主要代价在于组合 $I(P_1, \rho')$, 由于 $I(P_1, \rho')$ 数量较大, 因此有很高的并行性, 能采用多线程进行加速。我们将 $I(P_1, \rho')$ 分为多组, 每组分别由一个线程进行组合, 最后再将所有线程的结果相加, 这样能有效提高绘制性能。 $I(P_0, \rho')$ 相关的数据量并不大, 因此可以在GPU中实现, 但是由于绘制的主要代价在 $I(P_1, \rho')$, 相比之下GPU对于 $I(P_0, \rho')$ 的绘制的加速效果可以忽略不计, 因此我们没有采用这样的方法, 而是全部采用CPU进行绘制。

4.2.4 实验结果

我们用VC++2005和DirectX9在Windows XP下实现以上算法, 运行环境是dual 3.20 GHz Xeon CPU, 4.0 Gbytes RAM。BRDF的采样分辨率和前文4.1.5相同, $s \in [1, 200]$, 对于直接光照我们采用 $N_b = 6$ 个基, 间接光照我们采用 $N_b = 4$ 个基。实验的场景包含一个空间动态材质可变的物体, 子区域数量为 $32 \times 32 = 1024$, 一个空间静态材质可变的物体, PCA压缩率为12.5%, Haar小波压缩保留13.3%的系数, 绘制分辨率为 512×512 , 修改材质时的绘制速度为2.41 fps。预计算得到的数据量为21.0 Gbytes, 压缩后为781 Mbytes。

对于直接光照的BRDF的基函数的个数选择, 我们采用前文4.1的结论, 前

文4.1已经给出的数值误差和绘制结果与参考图像的比较证明这个结论的有效性。对于两次反射的间接光照，我们采用4个基函数，这是因为反射次数越多，光照效果就越低频，因此采用较少的基函数。采样BRDF的4个基函数的重建的 L^2 范数的误差为5.53%。子区域的数量取决于系统能够承受多大的数据量和运算代价，因为我们采用高密划分，高度压缩的策略，并没有对于子区域的划分作特别处理，用户可以在性能和效果的权衡下选择自己的参数设置。在下面的实验结果和参考图像的对比可以看出，我们的参数设置能在有限的预计算时间内生成良好的视觉效果。

表4-3中记录了所有实验结果相关的场景的规模和预计算的参数和性能。

表 4-3 所示的基于像素的空间动态可变材质的实验结果的预计算的性能

	场景的面片数量	每个像素的采样光线数量	光源的采样数量		时间(小时)	
			直接光照	间接光照	预计算	压缩
Cubes	38	16384	800	2	8.0	8.2
Bunny	23.2k	6400	800	2	7.8	8.4
Tube	1.2k	10000	1	1	2.2	7.3

后文中给出的数据中，每张绘制效果下面的三张纹理，从左到右分别赋给代表空间动态可变材质物体的 K_d ， K_s 和 s 。

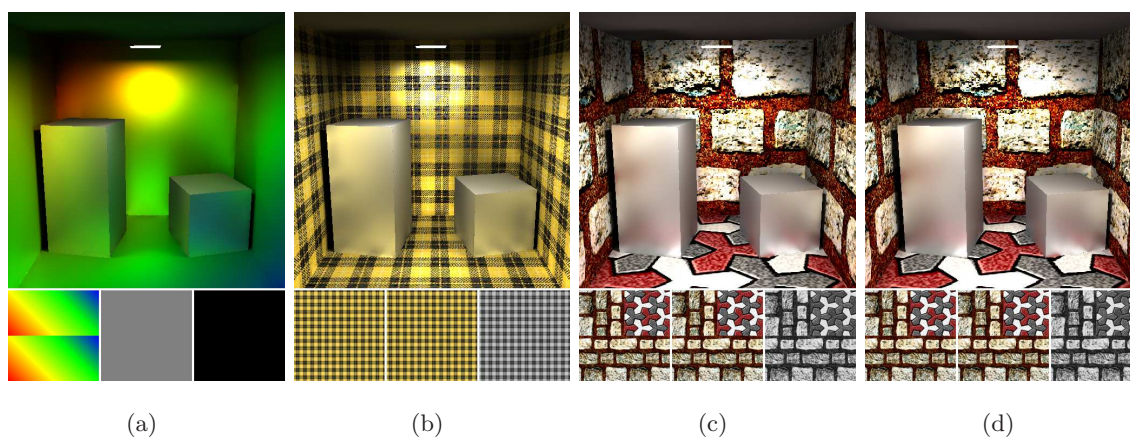


图 4-8 Cubes。(d)是(c)的参考图像， L^2 范数的误差为 $7.518e - 004$ 。

如图4-8，左右墙面、地板和cubes后面的墙面构成一个空间动态可变材质物体。Cubes被赋予高光材质，我们可以看到周围墙面在cube上明显的渗色效果。其中图4.8(b)和图4.8(c)墙面的材质有明显的区域和形态，我们可以在cubes上看到较为清晰的投影。

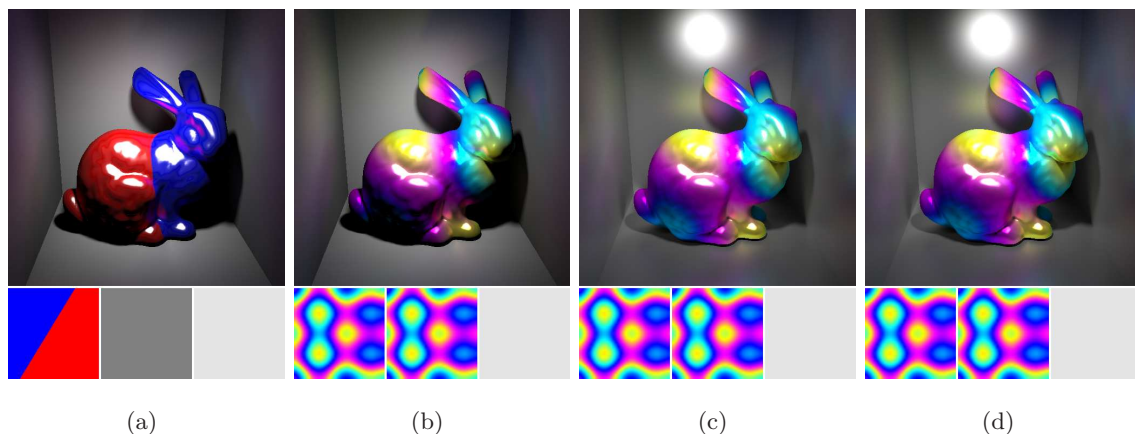


图 4-9 Bunny。 (d)是(c)的参考图像, L^2 范数的误差为 $2.8787e - 005$ 。

如图4-9, Bunny是一个空间动态可变材质物体。四周的平面是一个空间静态可变材质物体, 我们可以看到Bunny到四周墙面的明显的渗色效果。其中图4.9(a)和图4.9(d) 四周平面被赋予接近漫反射的材质; 图4.9(c)四周平面被赋予高光材质, 受到Bunny的渗色效果也就更明显。此外, 我们可以看到图4.9(a)中, Bunny上红色区域和蓝色区域的相互渗色的效果。

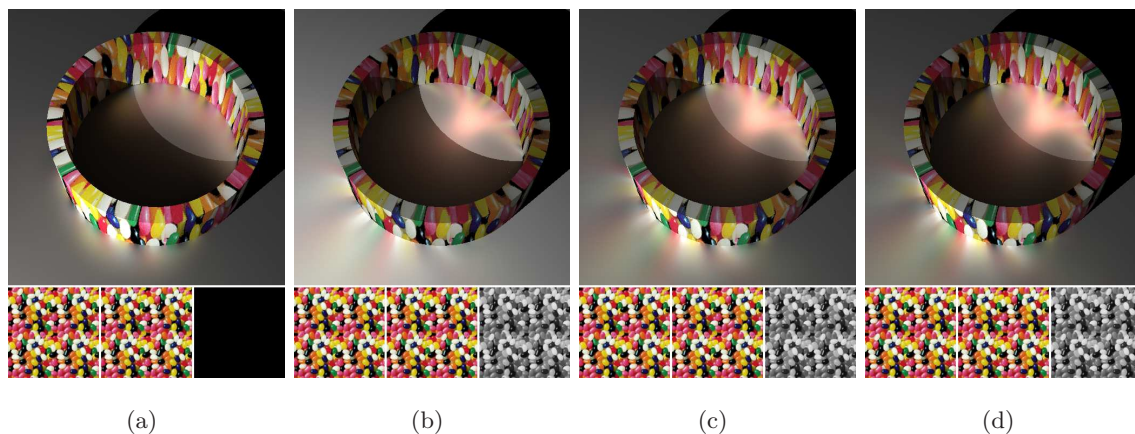


图 4-10 Tube。 (d)是(c)的参考图像, L^2 范数的误差为 $3.4280e - 004$ 。

如图4-10, tube是一个空间动态可变材质物体, 平面是一个空间静态可变材质物体。我们可以看到tube上高频的材质分布能在平面上投影出清晰的投影。其中图4.9(b)和图4.9(c) 圆环具有高光的材质, 因此在平面上有明显的焦散。图4.9(a)和图4.9(c) 平面被赋予了高光材质, 因此图4.9(a)圆环在平面上有清晰的倒影, 图4.9(c)的焦散比图4.9(b)更为明显和强烈。

4.2.5 结论

在4.1的工作中，我们实现了空间动态可变材质的交互级全局光照明绘制。我们将空间动态可变材质物体划分为多个空间静态可变材质的子区域。虽然会产生巨大的预计算的数据量，但是由于子区域之间的辐射传输有很高的稀疏性和相关性，因此可以将预计算得到的数据压缩至可以接受的范围来达到交互级的绘制性能。

4.2.5.1 局限性讨论

我们的方法具有十分现实的应用背景，比如能有效提高游戏、电影等的三维场景建模时材质编辑的效率。空间动态材质及其全局光照明对于场景的视觉效果是至关重要而且十分敏感的，但是同时具有极高的时间和空间的复杂度，在现有的硬件条件和技术水平下是难以在不做任何约束或假定的一般性情况下解决这个问题的。因此，我们的方法出于数据量和性能的考虑，对绘制的条件作了一些相应的限制，主要有以下三点：场景几何结构、光照条件和视点方向固定不变；只考虑一次反射形成的直接光照和两次反射性成的间接光照的效果，不考虑两次以上的反射形成的间接光照效果；不考虑BRDF的高光部分的相互反射效果。

后两个对间接光照的假设会影响全局光照的绘制效果，但是如果不做这样的假设，不但会带来巨大的数据量，严重降低预计算和绘制的性能外，而且能得到的绘制效果的提升还非常有限，因为在许多现实场景和材质条件下，两次反射已经能描述主要的光照效果，并且在它们的衬托下，其它被忽略的光照效果并不会带来明显的视觉差异。因此这两个假设广泛用于全局光照的算法中，如Hasan^[38]只考虑漫反射的间接光照效果，Ben-Artzi^[6]只处理直接光照的绘制效果。而在我们的实验数据中，我们算法的绘制结果和参考图象十分接近，充分说明这两个假设并不会影响我们的算法在许多场合的应用价值。只有当场景中有高光材质之间强烈的反射时，比如多次的镜面反射和折射，这两个假设可能造成绘制结果和真实效果的较大差异，这也是我们方法的局限性。如果需要解决对这类场景的应用问题，需要进一步的研究工作，采用更具针对性的方法。

而对我们算法的实际应用局限最大的，是对光源和视点固定不变的假设，因为这要求在场景设计和编辑的时候就要求先确定光照和视点方向，然后再进行材质编辑，而不能同时进行。而对于几何结构不变的假设却不会太局限我们方法的应用，因为在实际应用中，比如室内装潢、电影及游戏的场景设计中，很少是通过改变几何结构来调整光照效果的，事实上都是先确定几何结构再进行光照效果的设计。后

文第5章能在绘制全局光照明效果的同时改变光照、视点和材质，但是却不能处理空间动态可变材质，也就是说无法承担空间动态可变材质所带来的巨大的数据量和预计算绘制的运算代价，因此也无法解决这个问题。

综上所述，虽然我们的算法依赖一些假设和约定，但是并不会影响算法的有效性和实际应用价值。原因总结起来有四点：这些假设是出于对数据量和运算性能考虑的合理假设，并为相关工作广泛采用；对于大多数应用场景能够提供足够的效果和性能的保证；我们首次解决了空间动态可变材质的交互级全局光照明绘制，这是前人工作所不能解决的问题；对用户的额外要求（先确定光照和视点然后进行材质编辑）并不严苛，虽然不是十分完美，但是相对于非交互级地编辑和预览绘制效果，已经能带来极大的便利。

第5章 基于顶点的空间静态可变材质的交互级全局光照照明绘制

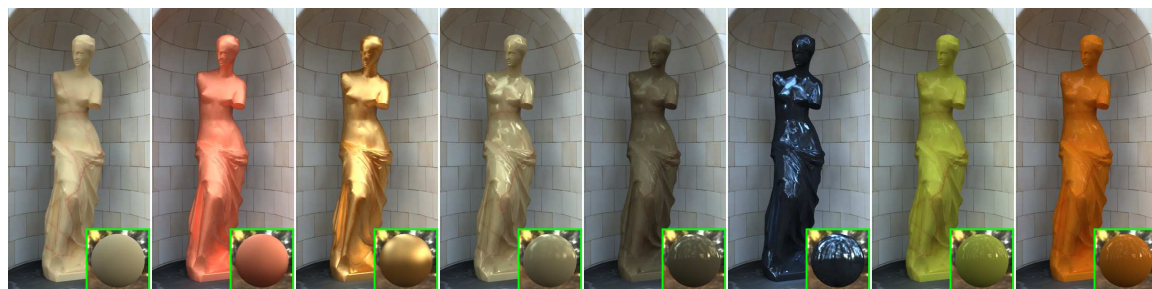


图 5-1 不同BRDF的维纳斯雕像的绘制效果。右下角是对应BRDF的球模型的参考图像。在周围的墙上可以看到随着维纳斯的反射材质的改变而产生的明显的间接光照效果的不同。这个绘制场景有39.7K个顶点，预计算时间为12.3小时，内存占用389 Mbytes。如果视点、材质和光照都被改变，绘制性能为2.62 fps；如果只有视点和材质被改变，绘制性能为10.94 fps；如果只有视点被改变，绘制性能为57.09 fps。

这一章详细介绍了基于顶点的空间静态可变材质的交互级全局光照照明绘制算法。和第4章的算法相比，本章的算法最重要的优点在于，不但可以改变物体的材质，还可以改变场景的环境光照和视点。

全局光照的绘制结果取决于多个参数，包括物体几何、光照、材质和视点等。实时全局光照算法的目标，就是在实时绘制全局光照效果的过程中，允许用户通过交互修改其中一部分甚至全部参数。Sloan^[84]提出的PRT绘制框架，主要解决的是改变光照和视点条件下的实时全局光照照明绘制问题。Ben-Artzi^[5, 6]以及第4章中的方法，解决的是改变材质条件下的实时全局光照照明绘制问题。在提出我们的方法之前，这两类方法是互相不兼容的。换句话说，已有的实时全局光照照明绘制算法不能兼顾动态光照和可变材质，两者只能选择其一。但是在场景建模、计算机辅助设计等应用领域，用户所期望的是能同时改变光照、材质和视点。用户一般首先完成场景几何的建模，然后编辑选择合适的光照和材质。强制要求先编辑光照，后编辑材质，是对用户提出的一种不切实际的要求和约束，反之亦然。因此，我们需要的是一种能够同时改变光照、材质和视点的绘制框架。而这正是本章的研究内容，也是本章的算法具有很高的应用价值的原因所在。

为了实现上述目标，我们主要采用了三种策略：光路切分，预计算传输张量(precomputed transfer tensors, PTTs)，针对BRDF的镜面项分离和张量分解。光路切分的目的，在于减小辐射传输的复杂度，使得现有的硬件资源能够负荷其存储量，并达到交互级的绘制性能。PTT的目的，在于将材质表示为辐射传输的一个独立变量，从而使得预计算得到的辐射传输，能够在绘制时根据不同的材质组合得到相应传输结果。我们采用BRDF表示物体的材质。针对BRDF的镜面项分离和张量分解，帮助我们实现了在动态局部入射辐射亮度、可变材质、动态视点条件下的物体表面的快速着色。

我们的算法实现了静态几何，动态环境光照、动态视点、空间静态可变材质的交互级全局光照明绘制，在光照不变的情况下，能够达到实时的绘制性能。虽然我们所涉及的问题的存储和运算的复杂度很高，但是我们通过选择合理的基函数和压缩方法，保留了大量的高频光照效果，所以我们的方法能够获得高质量的绘制图像。我们的方法适用于材质编辑、BRDF动画等应用。

5.1 算法概述

Sloan^[84]提出的PRT绘制框架，不支持可变材质的实时全局光照明绘制。其原因在于将材质参数作为不变量内嵌在预计算得到的辐射传输的数据中，在绘制时也就无法根据不同的材质组合得到相应的辐射传输。Ben-Artzi^[5, 6]以及第4章中的方法，不支持动态光照和动态视点。其原因在于其采用的辐射传输的数学表示，耗尽了所有的存储和运算资源；换句话说，其所采用的将物体材质作为独立变量的辐射传输的表示，效用很低，因此无法在有限的资源内，既处理可变材质的数据空间，又处理动态光照和动态视点的数据空间。

所以，我们的思路也可依次确定两个要点：将辐射传输表示为材质参数的函数，并基于此进行预计算和绘制；采用合理而有效的辐射传输的表示，能够在有限的资源内，包含动态光照、可变材质、动态视点所涵盖的所有数据空间。

我们沿用PRT的基于顶点的绘制框架，即计算场景中每个顶点的辐射亮度，然后插值得到每个像素的辐射亮度，不同于第4章中独立绘制每个像素的辐射亮度的方式。我们将全局光照的辐射传输的整个过程切分为两个部分：第一个部分包含从光源到最后一次反射的反射点的传播过程，称之为入射辐射传输；第二个部分包含余下的最后一次反射并到达视点的传播过程，称之为顶点着色。显然，这两部分的辐射传输的数据空间的加和，要远远小于整个辐射传输过程的数据空间。我们必

须将入射辐射传输表示为光照和材质的函数。为此，我们将全局光照的辐射传输分解成许多不同的部分，每个部分对应不同的反射路径。每一部分的辐射传输也就能够表示成光照和对应的反射路径上所有的反射材质的函数。所有部分的辐射传输函数的总和就是PTT。PTT中根据传播路径将辐射传输划分为多个部分方法，和第4章中的辐射传输分离的方法相近，所不同的是PTT只包含前一部分的辐射传输，而不是辐射传输的完整过程。对于顶点着色部分，我们的目标是实现物体表面的点在动态的局部入射光照、动态的视点、动态的BRDF下的快速着色。为此，我们采用一个三阶张量表示BRDF空间，然后通过张量分解将BRDF空间的光照、视点、BRDF参数等三个原先互相耦合的动态因素分离开，从而实现了物体表面的点的快速着色。如果BRDF空间中存在高光强烈甚至接近镜面反射的BRDF，那么为了高质量的绘制效果，需要保留大量的BRDF的基会数量巨大，但是这样会严重影响绘制的性能。为了克服这个问题，我们在张量分解前先将BRDF张量的镜面项分离开，以保证BRDF张量足够低频，只需要少量基就足够保证重建的精度。在绘制时，分离出来的镜面项采用预计算得到的可见性信息进行绘制，然后加和到BRDF张量绘制的结果上，就能得到高质量的绘制图像。

5.2 基于预计算传输张量的重光照

图5-2直观地描述了间接光照的物理过程。绘制场景采用环境光照，光源处于无穷远的位置。从光源发出的光在场景中传播，形成了间接光照和阴影等全局光照明效果。如果场景的空间几何结构固定不变的话，场景中的辐射传输由物体的材质决定。我们将场景中到达物体表面，并即将经过最后一次反射然后到达视点的辐射亮度称为辐射传输的入射辐射亮度(transferred incident radiance)。当辐射传输的入射辐射亮度到达物体表面时，他最后一次反射的结果取决于该点的BRDF。该点最后出射到视点的辐射亮度取决于辐射传输的入射辐射亮度、视点和该点的BRDF。

重光照的目的是在绘制时改变场景的光照 L 和视点 ω_o ，或者对于更通用的情况，改变场景的反射材质 H 。我们用矩阵 $n_R \times n_P$ 表示场景的材质，其中 n_R 是场景中采用可变材质的物体的数量， n_P 是用以表示整个BRDF空间的一组线性基的数量。基于Kajiya的绘制方程^[47]，场景中一个点 x 的出射的辐射亮度可以表示为：

$$B_x(L, H, \omega_o) = \int_{\Omega} I_x(L, H, \omega_i) f(H_x, \omega_o, \omega_i) (\omega_i \cdot \mathbf{n}_x) d\omega_i. \quad (5-1)$$

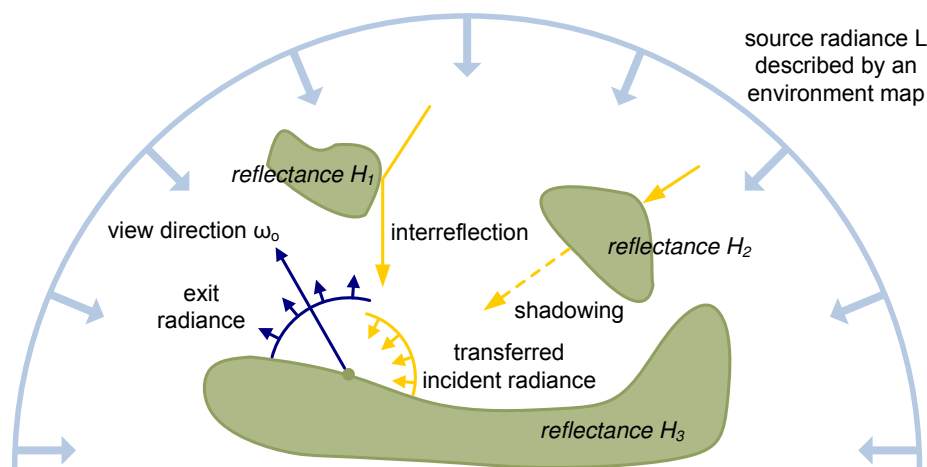


图 5-2 包含间接光照效果的重光照

其中 B_x 表示最终到达视点的辐射亮度； f 表示点 x 的反射材质的BRDF的数值。从这个重光照方程中我们可以看到两个要素。一个是达到点 x 的辐射传输的入射辐射亮度 $I_x(L, H, \omega_i)$ ，它依赖于场景中物体的BRDF。另一个是点 x 自身的BRDF的数值 $f(H_x, \omega_o, \omega_i)$ ，它和入射的辐射亮度的方向 ω_i 、入射辐射亮度的方向 ω_o 、该点的反射参数 H_x 有关。

我们根据不同的环境光照、场景BRDF分布、局部入射方向的不同设置，预计算各自的辐射传输的入射辐射亮度。我们对于点 x 的可变BRDF的整个空间采用张量表示，然后将这个张量的入射方向、出射方向、反射参数的三个维度分解开。基于这两个策略，我们能够在绘制时快速地进行式5-1的计算得到该点最终到达视点的辐射亮度。我们将介绍这些方法的详细内容和实现细节。

5.2.1 辐射传输的入射辐射亮度

最终到达物体表面的局部入射的辐射亮度，可能经过各种不同的反射路径，直观的物理过程如图5-3所示。对于一条特定的反射路径，最终的辐射亮度由反射路径所经过的反射物体的材质所决定。我们将BRDF空间的采样经过PCA得到的 n_P 个线性基 p ，并以此表示整个BRDF空间。对于任何一个可变材质的物体，用户可以通过修改BRDF的基 p 的系数 $H(r, p)$ 来改变它的BRDF。

有了绘制场景的BRDF空间的线性基，我们将每条反射路径对应的辐射传输进一步分解为几个部分，每个部分的辐射传输对应于反射路径上的物体的BRDF是采用不同的基的组合。举例来说，有一部分的辐射传输，首先经过物体 r_1 反射，并

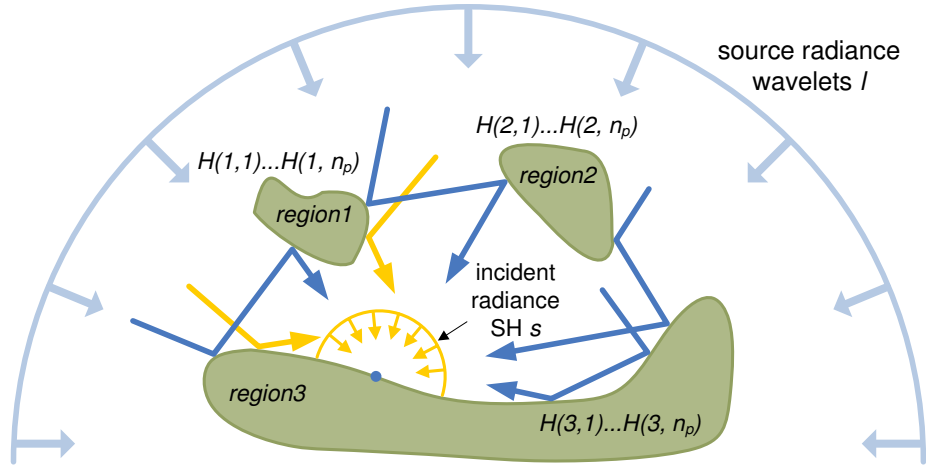


图 5-3 间接光照的光路包含了一个BRDF基函数序列，采用PTT表示。图中黄色的光线代表经过一次反射的间接入射光照，其中 $H(r, p)$ 代表物体 r 对应的BRDF的基函数 p 的系数。图中蓝色的光线代表经过两次反射的间接入射光照

且 r_1 的材质是采用BRDF的基 p_1 ，并且基的系数是 $H(r_1, p_1)$ ，然后经过物体 r_2 的反射，并且 r_2 的材质是采用BRDF的基 p_2 ，并且基的系数是 $H(r_2, p_2)$ ，最后到达点 x 形成辐射传输的入射辐射亮度的一部分，我们就将这部分的辐射传输单独预计算并存储，并作为区域 r_1 、 r_2 以及BRDF的基 p_1 、 p_2 的函数。为了在绘制时能快速地计算间接光照，我们将整体的辐射传输的入射辐射亮度分为这样的许多部分，这样的每一部分都是场景的光照的基，以及物体BRDF空间的基的分布的函数，并独立预计算和存储。所有各部分的预计算的结果在一起被存储为PTT，这样在绘制的时候，只要指定环境光照 L 的系数和场景中可变材质物体的BRDF投影到基上的系数 H ，就能快速得到最终的辐射传输的入射辐射亮度。

为了方便处理PTT中每一项所经过反射的BRDF的序列，我们将光的反射路径根据其反射次数分为不同的组，这样所有的辐射传输的入射辐射亮度可以表示如下：

$$I_x = I_x^0 + I_x^1 + I_x^2 + \dots \quad (5-2)$$

其中 I_x^k 表示这部分的辐射传播的入射辐射亮度在最后到达物体表面经过了 k 次反射。在绝大多数场景中，存在了数之不尽的辐射传播的反射路径。但是，由于在物体表面的每次反射都会吸收大量的辐射，随着 k 的增加，辐射传输的能量和亮度会非常迅速地减小，因此在具体的实现中，我们只考虑 $k \leq 2$ 的反射路径的

辐射传输。除此之外，由于反射后出射的辐射亮度会发散到很多方向，因此间接光照的辐射传输随着反射次数 k 的增加也会变得更加低频。根据全局光照的辐射传输的这个特点，对不同反射次数的辐射传播的入射辐射亮度采用不同的表述方法，直接光照的 I_x^0 采用小波来表示，经过一次反射的间接光照的 I_x^1 采用四阶的SH来表示，经过两次反射的间接光照的 I_x^2 用一个标量来表示。

直接光照 I_x^0 的计算较为简单，只需要将环境光和PTT中的 $T_x^0(\omega)$ 点积即可，而PTT的 $T_x^0(\omega)$ 就是物体表面的点对于环境光照的关于方向 ω 的可见性函数。所有顶点关于整个环境光照的可见性函数需要预计算得到并存储。

如图5-3所示，一次反射的间接光照的 I_x^1 包含了 n_R 个不同的辐射传播的路径，而每条传播路径所经过的反射物体可以使用 n_P 个不同的BRDF的基函数。为了加速绘制时 I_x^1 的计算，我们预计算得到的PTT的 T_x^1 中的每一项都对应某一条反射路径和某一个BRDF的基函数。这样，我们就得到了经过一次反射的PTT的4阶张量 $T_x^1(l, r, p, s)$ ，其中光照采用的小波基用 l 表示，不同的反射物体用 r 表示，不同的BRDF的基函数用 p 表示，而得到的辐射传输的入射辐射亮度的SH基用 s 表示。这个张量通过采样不同的 l 、 r 、 p 得到，对每一个采样我们用光线跟踪算法计算每个点 x 得到的辐射传输的入射辐射亮度，然后将这个入射辐射亮度投影到SH基 s 得到最终的 T_x^1 的结果。

对经过两次反射的间接光照的处理也很类似，我们需要计算的有 $n_R \times n_R$ 个不同的辐射传输的反射路径，每条反射路径所包含的两个反射物体一共可以采用 $n_P \times n_P$ 中不同的BRDF的基函数的组合。由于我们采用一个标量来表示 I_x^2 ，我们需要预计算的两次反射的PTT是一个5阶张量 $T_x^2(l, r_1, p_1, r_2, p_2)$ ，其中 r 和 p 的下标指明了辐射传输经过的第一次和第二次反射。

Aupperle^[4]采用了三阶张量表示场景中面片的有序的三元组之间的辐射传输的关系，这种局部的传输关系通过组合递归描述整个场景中的全局光照的辐射传输。相比之下，我们的PTT的不同之处是包含了整个从光源到达物体表面所经过的多次反射的整个光路，同时不包含到达视点前的最后一次反射，除了包含不同反射物体的信息外，PTT还包含了不同的BRDF的基函数的信息，因此不只是光路的函数，而是光路和材质的函数。

5.2.2 BRDF

为了能快速计算物体表面点的BRDF的值，我们用张量来表示可变材质

的BRDF空间，并且用张量分解的方法将入射方向、出射方向、反射参数等三个部分分离开。张量中的BRDF通过采样得到 $f(h, \omega_o, \omega_i)$ ，其中 h 是BRDF采样的参数， ω_o 是出射方向的参数， ω_i 是入射方向的参数。如果给定一个BRDF的解析表示，我们就通过采样解析式的参数产生BRDF采样，如果是采用数据驱动的BRDF模型，我们就直接采用采集得到的BRDF。

为了表示BRDF函数，我们将每一个BRDF函数离散采样，所有的BRDF样本综合 $f(h, \omega_o, \omega_i)$ 就离散采样成三阶的张量 $F \in \mathbb{R}^{N_h \times N_{\omega_o} \times N_{\omega_i}}$ ，其中的 N_h 、 N_{ω_o} 、 N_{ω_i} 分别对应 h 、 ω_o 、 ω_i 的采样分辨率。采用^[51]的方法将这个张量分解为三个各自独立依赖于 h 、 ω_o 、 ω_i 的部分， F 可以表示为：

$$F \approx D \times_h U^h \times_{\omega_o} U^{\omega_o} \times_{\omega_i} U^{\omega_i} \quad (5-3)$$

而一个指定的BRDF的值可以这样得到：

$$f(h, \omega_o, \omega_i) \approx \sum_{i=1}^{n_h} \sum_{j=1}^{n_{\omega_o}} \sum_{k=1}^{n_{\omega_i}} D(i, j, k) U^h(h, i) U^{\omega_o}(\omega_o, j) U^{\omega_i}(\omega_i, k) \quad (5-4)$$

其中 $D \in \mathbb{R}^{n_h \times n_{\omega_o} \times n_{\omega_i}}$ 是BRDF的张量核，而反射参数、出射方向、入射方向的采样分辨率满足 $n_h \leq N_h$ 、 $n_{\omega_o} \leq N_{\omega_o}$ 、 $n_{\omega_i} \leq N_{\omega_i}$ 。反射参数的投影矩阵 U^h 、出射方向的投影矩阵 U^{ω_o} 和入射方向的投影矩阵 U^{ω_i} 的维度分别是 $N_h \times n_h$ 、 $N_{\omega_o} \times n_{\omega_o}$ 和 $N_{\omega_i} \times n_{\omega_i}$ 。

通过张量分解，我们可以很大程度地压缩BRDF的存储 F ，并且易于计算任意入射辐射亮度分布、任意出射方向、任意BRDF参数下的反射率。如果只是采用特征值分解(singular value decomposition, SVD)的方法将BRDF的入射方向和反射方向分解开^[48]，那么必须对每一个候选的BRDF都要进行SVD分解，而我们的这种三线性分解方法由于同时对三个维度进行分解，因此能更大程度地进行压缩。此外，采用张量分解的方法^[51]能够得到最小的三线性分解的均方根(root mean square, RMS)误差，而且压缩的结果同样能够得到如同^[91, 96]的高质量绘制结果。

我们在处理表面着色的时候采用张量分解的方法压缩BRDF空间，在处理辐射传输的入射辐射亮度的PTT时，我们采用PCA的方法压缩BRDF空间，这样采用两种不同的BRDF空间压缩及表述方法是考虑到基于计算的时间和空间的性能。对于表面着色而言，我们需要在所有的半球面入射方向上进行积分，因此将光照方向的

维度从 N_L 降到 n_L 能带来性能上巨大的收益，而PCA得到的BRDF的基函数没有压缩入射方向的维度。对PTT而言，采用BRDF的张量基函数就非常的低效，因为在预计算时计算物体表面反射的时候，需要计算所有出射方向的辐射亮度。所以张量基在出射方向维度上的压缩就没有意义。此外，由于没有将入射方向和出射方向分解开，BRDF的PCA的基就能避免不必要的精度上的损失，能够在保证绘制效果的情况下尽可能采用较少数量的基函数。

5.2.3 重光照

采用PTT计算辐射传输的入射辐射亮度，并且使用张量分解的BRDF的基函数计算物体表面的着色，我们可以在绘制时快速地计算任意环境光照、任意场景中物体的BRDF、任意的视点下的重光照的结果。假设环境光照 L 的在Haar小波基的表示为 $\{L_l | 1 \leq l \leq N_L\}$ ，那么 $L = \sum C_l^L L_l$ ^[61]。我们将式5-4的BRDF张量分解的结果代入绘制方程式5-1重写得到出射的辐射亮度为：

$$B_x(L, H, \omega_o) \approx \sum_{i=1}^{n_h} \sum_{j=1}^{n_{\omega_o}} \sum_{k=1}^{n_{\omega_i}} D(i, j, k) U^h(H_x, i) U^{\omega_o}(\omega_o, j) \sum_{l=1}^{N_L} C_l^L U_x^I(l, k, H). \quad (5-5)$$

其中 U_x^I 表示物体表面的点 x 受到的辐射传输的入射辐射亮度在经过BRDF的入射方向的投影矩阵的 U^{ω_i} 的作用后得到的系数。在绘制时，我们根据PTT的不同部分分别计算然后加和得到：

$$U_x^I = U_x^{I0} + U_x^{I1} + U_x^{I2} \quad (5-6)$$

其中 U_x^{Iq} 表示中经过场景物体 q 次反射的间接光照的入射辐射亮度作用得到的 U_x^I 的组成部分。它们的每一项都是独立计算的，具体实现如下：

5.2.3.1 直接光照

在直接光照的计算中，辐射传输的入射辐射亮度取决于环境光源的可见性，也就是PTT中的 T_x^0 。为了计算的方便，我们在具体实现中预计算的是 $\tilde{T}_x^0(\omega, k) = U^{\omega_i}(\omega, k)(\omega \cdot \mathbf{n}_x) T_x^0(\omega)$ 也就是可见性和BRDF的基函数的入射的投影矩阵中分量的乘积，类似于^[54, 97]的实现。因此入射的辐射亮度能够被快速计算，如下：

$$U_x^{I0}(l, k) = \int_{\Omega} L_l(\omega) \tilde{T}_x^0(\omega, k) d\omega \quad (5-7)$$

其中 L_l 表示环境光照的第 l 个小波基函数。因为直接光照的入射辐射亮度在到达点 x 之前没有和场景有反射，那么它和场景中物体的材质没有关系。

5.2.3.2 经过一次反射的间接光照明

在绘制时，我们将PTT中的 T^1 和所经过的反射物体的BRDF对应的PCA基的系数乘积，然后加和就得到了需要的经过一次反射的间接光照带来的辐射传输的入射辐射亮度，再通过和矩阵 C^{ω_i} 的内积操作就得到了投影到BRDF的张量基的入射方向的投影矩阵上的结果：

$$U_x^{I1}(l, k, H) = \sum_{s=1}^S C^{\omega_i}(s, k) \sum_{r=1}^{n_R} \sum_{p=1}^{n_P} H(r, p) T_x^1(l, r, p, s) \quad (5-8)$$

其中：

$$C^{\omega_i}(s, k) = \int_{\Omega} L_s^{SH}(\omega) U^{\omega_i}(\omega, k) (\omega \cdot \mathbf{n}_x) d\omega. \quad (5-9)$$

其中 $H(r, p)$ 表示物体 r 的BRDF在其基函数 p 上的系数。和 T^1 相同， C^{ω_i} 也是计算在SH基上的一组系数。

如果只根据以上的推演，经过一次反射的间接光照带来的辐射传输的入射辐射亮度可以直接计算在BRDF的张量基的入射方向的投影矩阵的分量上，而不需要计算在SH基上，这样就减少了一个绘制的步骤，不但减小了存储空间，还提高了绘制性能。但是这样的话我们只能用BRDF的张量基完成最后的物体表面的着色，而不能很好地重建点 x 的入射辐射亮度在整个半球面上的分布。后文5.3.1中所采用的镜面项分离的方法，在绘制时需要得到入射辐射亮度在整个半球面上的分布。此外，使用SH基能够帮助我们在计算重光照时非常明确而清晰地将辐射传输的入射辐射亮度和物体表面的着色这两步操作完全分解开，如果有必要的话，两者中的任何一个都可以被一个等价的算法所替代，而不影响整条算法流水线的可行性。

5.2.3.3 经过两次反射的间接光照明

为了计算经过两次反射的间接光照带来的辐射传输的入射辐射亮度，我们需要考虑计算的是辐射传输先后经过的场景中两个反射物体(r_1, r_2)的材质。因为我们假设经过两次反射的间接光照带来的辐射传输的入射辐射亮度是均匀分布的，因此最

终到达点 x 的入射部分的系数计算为：

$$U_x^{I2}(l, k, H) = C^{\omega_i}(k) \sum_{r_1=1}^{n_R} \sum_{p_1=1}^{n_P} \sum_{r_2=1}^{n_R} \sum_{p_2=1}^{n_P} H(r_1, p_1) H(r_2, p_2) T_x^2(l, r_1, p_1, r_2, p_2) \quad (5-10)$$

其中：

$$C^{\omega_i}(k) = \int_{\Omega} U^{\omega_i}(\omega, k) (\omega \cdot \mathbf{n}_x) d\omega. \quad (5-11)$$

5.3 算法实现

5.3.1 BRDF的镜面项分离

如果BRDF空间中包含很多高光强烈的BRDF，那么在张量基的项数就需要保留很多，而太多的基函数会使得我们无法达到交互级的绘制性能。为了也能在我们的算法中绘制高光强烈的BRDF，我们从BRDF张量中分离出镜面项，在绘制的时候使用预计算的可见性数据进行计算，加回到最终的绘制结果中。其中镜面项按照分辨率为 $64 \times 64 \times 6$ 的立方纹理(cubemap)采样方向。这种镜面项分离的方法把BRDF分为镜面项 f_s 和剩余项 f_r ：

$$f(h, \omega_o, \omega_i) = f_s(h, \omega_o, \omega_i) + f_r(h, \omega_o, \omega_i). \quad (5-12)$$

其中镜面项 f_s 用不同立体角范围内的高斯函数的加和来表示：

$$f_s(h, \omega_o, \omega_i) = \sum_{k=0^\circ, 7^\circ, 14^\circ, 21^\circ} f_s^{(k)}(h, \omega_o, \omega_i) \quad (5-13)$$

其中 $f_s^{(k)}$ 是一个单变量的高斯函数，它的标准差是以反射方向 ω_o 为中心的 k 度的空间角。其中 k 的采样数值是根据立方纹理的分辨率得到的。比如 0° 表示标准的镜面反射，之后每一项镜面项比之前一项，所覆盖的立体角的半径大了立方纹理的一个像素，如图5-4所示。从BRDF采样中分离出镜面项 $f_s^{(k)}$ 所采用的方法是期望最大化的数值算法^[19]，其约束为以镜面反射方向为中心的几个半径不同的立体角内的高斯函数，比如 $k = 0^\circ, 7^\circ, 14^\circ, 21^\circ$ 。从BRDF采样中分离出镜面项 f_s 后，留下的剩余项 f_r 相比之下低频许多，因此更为容易采用张量分解的方法进行压缩近似。

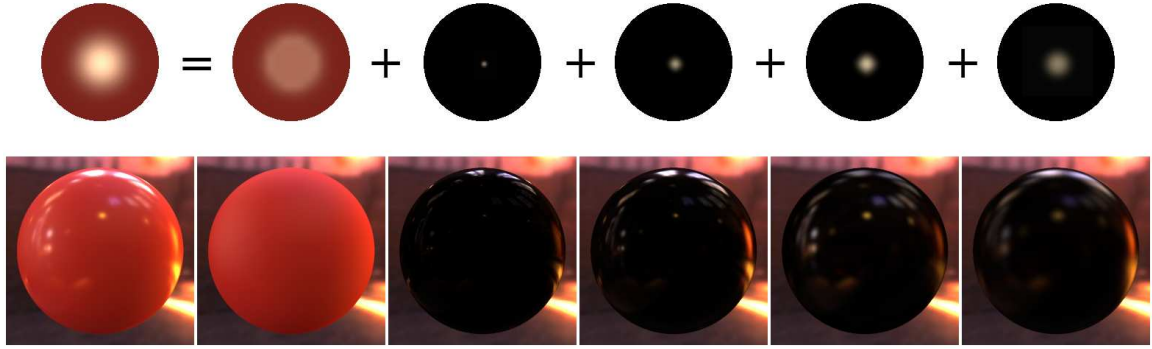


图 5-4 镜面项分离。从左向右：初始BRDF；分离后的剩余部分； $0^\circ, 7^\circ, 14^\circ, 21^\circ$ 的镜面项部分。

BRDF张量使用 f_r 得到，而 f_s 的绘制需要单独处理。我们将环境光使用在立体角 $k = 0^\circ, 7^\circ, 14^\circ$, or 21° 内的高斯函数进行滤波。为了计算直接光照的镜面项的绘制结果，我们将镜面项中的每一项的 $f_s^{(k)}$ 和相应的滤波得到的环境光照在镜面方向上的值相乘。将所有乘积的结果相加就得到了直接光照的镜面项的绘制结果。和环境光照类似，我们也需要将SH基用高斯函数进行滤波。我们通过PTT计算得到经过一次反射的辐射传播的入射辐射亮度的SH的系数，用这些系数和滤波后的SH基点积就能重建入射辐射亮度的整个半球面分布在高斯滤波后的结果，将 f_s 的所有项和这些重建结果的镜面方向的数值点积，就得到了经过一次反射的间接光照的入射辐射亮度带来的镜面项的绘制结果。经过两次反射的间接光照的入射辐射亮度被假设为均匀分布，我们不考虑它所带来的镜面项的绘制结果。为了能和滤波后的环境光照相乘，我们使用浮点数表示记录每个方向在不同的立体角范围内的可见比例。

BRDF的采用镜面项分离后的张量分解的误差如图5-5，我们采用的是100个采集得到的BRDF用Cook-Torrance BRDF模型^[16]拟合的结果^[63]。采用最高的参数设置 $n_h = 6$ 、 $n_{\omega_i} = 32$ ，不采用镜面项分离的张量分解的均方误差(squared error)是0.952465，明显比采用较低参数设置以及镜面项分离的张量分解的均方误差高很多。不同高光程度的BRDF的绘制结果的比较如图5-6。分离的镜面项的项数越多，BRDF重建的绘制效果越好。

Green^[31]将高频的辐射传输采用一组高斯函数的和来表示，与低频的辐射传输分别计算。这些传输函数都是针对场景中的固定的BRDF预计算得到，因此不能用于可变材质的重光照。在我们的BRDF的张量表示中，我们不采用半角的参数化方法表示BRDF^[56]，因为我们需要将BRDF的入射方向和出射方向分解开。也正是由

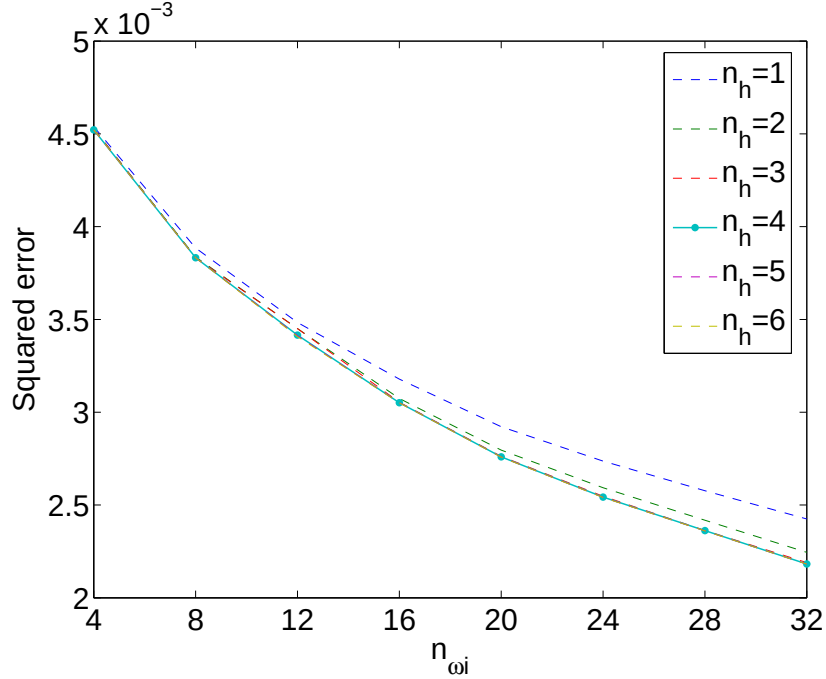


图 5-5 不同的 n_h 和 n_{ω_i} 得到的张量分解的误差

于我们不能采用BRDF的半角参数化方法，所以我们需要采用镜面项分离的方法减小张量分解的数值误差。

5.3.2 BRDF的编辑和浏览

为了方便用户修改物体的BRDF，我们根据所采用的不同模型的BRDF采用不同的BRDF的编辑和浏览方法。我们使用了Blinn-Phong^[7]和Cook-Torrance^[16]两种不同的BRDF模型来表示BRDF的非漫反射部分，BRDF的漫反射部分被单独处理。对于Blinn-Phong模型，BRDF的样本通过采样200个BRDF得到，这200个采样BRDF的高光系数为1到200的整数。绘制时任何一个BRDF根据其高光系数在所有的样本中线性插值得到。

对于Cook-Torrance模型，我们的BRDF样本来自100个从现实材质中采样得到的BRDF拟合得到的结果^[63]，其根据菲涅尔(Fresnel)系数和表面粗糙度系数得到的分布如图5-7所示。将图中这些采样点三角化后，我们在它们所构成的凸包中任取一个点，然后将这个点所在的三角形的三个顶点根据这个点的位置进行重心插值，就是浏览所得到的BRDF。在对Cook-Torrance模型和Blinn-Phong模型的BRDF的浏览中，我们对漫反射和非漫反射的参数和系数都需要进行插值。

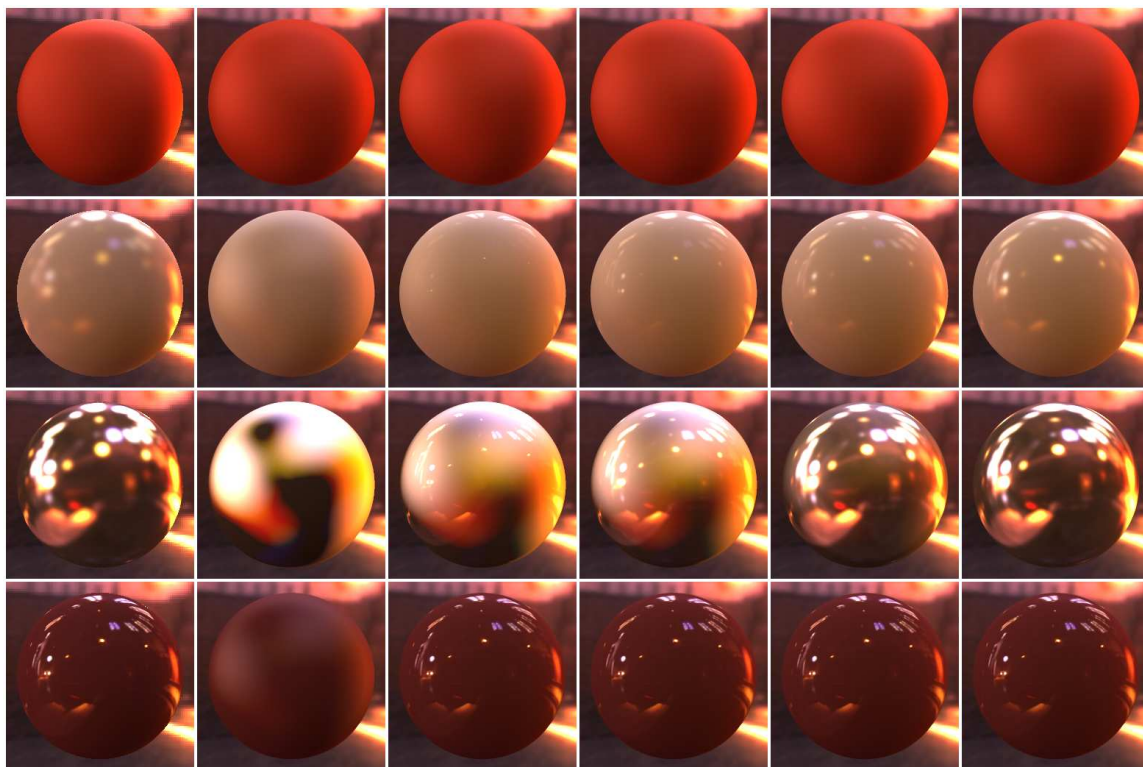


图 5-6 不同项数的镜面项分离得到的BRDF的张量分解的重构结果。从左到右：参考图像；没有镜面项分离；镜面项项数为1 (0°)，2 ($0^\circ, 7^\circ$)，3 ($0^\circ, 7^\circ, 14^\circ$)，4 ($0^\circ, 7^\circ, 14^\circ, 21^\circ$)。

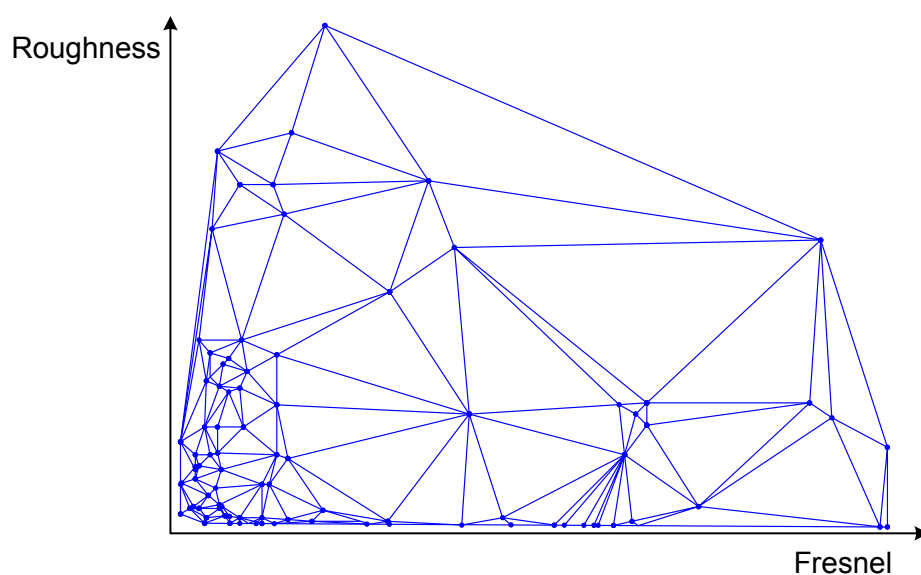


图 5-7 采集得到的BRDF用Cook-Torrance模型进行拟合

5.3.3 实现的参数和运算代价的分析

由于对于BRDF、光照、视点、光路等巨大的采样数量，我们需要在性能和效果上作许多权衡。我们主要沿用了许多现有工作的参数设置，比如^[88]采用经过一次反射形成的间接光照来近似所有的间接光照，^[49]采用的4阶SH基作为光照的基函数来作为间接光照的光源。

5.3.3.1 BRDF的张量表示和分解

在物体表面的着色操作中，BRDF的非漫反射部分用张量表示，漫反射部分独立于张量分解外单独处理。在我们的具体实现里，BRDF的张量的参数设置为 $n_h = 4$ 、 $n_{\omega_i} = n_{\omega_o} = 8$ 。BRDF空间张量分解得到的入射方向和出射方向的投影矩阵 U^{ω_i} 、 U^{ω_o} 的方向采样采用Wang^[97]的采样方法，通过将法向上的单位半球面正投影到单位圆上得到。

在我们的具体实现中，我们的BRDF空间的张量分解的方法对于我们所采用的两种BRDF模型：Blinn-Phong模型以及采集得到然后拟合到Cook-Torrance模型上的100个BRDF^[63]，都能得到很好的数值和绘制结果。为了分析Cook-Torrance模型的100个BRDF的张量分解的结果，我们测试了不同的参数设置 $n_{\omega_i} \times n_h = \{4, 8, 12, 16, 20, 24, 28, 32\} \times \{1, 2, 3, 4, 5, 6\}$ 。具体每个参数设置的均方误差如图5-5。我们最终将 n_h 设置为4，因为误差数值在4收敛，并且4项的近似再加上镜面项的处理能够得到足够好的绘制结果^[97]。基于类似的分析，我们把Blinn-Phong模型的BRDF空间的 n_h 设置为3，因为其变化性较小。由于两个模型的BRDF都是各向同性的，我们采用相同的 n_{ω_o} 和 n_{ω_i} 。在我们的实现里，我们设置 $n_{\omega_i} = n_{\omega_o} = 8$ ，这主要是考虑到保证基本绘制效果的前提下尽可能提高绘制的性能。对于高光越强烈的BRDF，我们采用的镜面项分离的方法能够更明显地提高绘制的效果，如图5-6。本章中所示实验结果用到的张量分解都在一个小时内完成。

5.3.3.2 BRDF的PCA分解

在PTT的预计算和绘制中，我们使用BRDF的PCA的基函数，我们一共用5个基函数(四个非漫反射的基函数和一个漫反射的基)，Blinn-Phong模型和Cook-Torrance模型的BRDF的重建均方误差分别是 $6.23e-4$ and $1.61e-7$ 。这个误差和张量分解的误差相当，尽管计算PCA的基的时候没有采用镜面项分离的方法。

5.3.3.3 直接光照

对每一个顶点，入射的系数 U_x^{I0} 是一个 $N_L \times n_{\omega_i}$ 矩阵。为了得到高频的光照效果， N_L 必须非常大，在我们的实现中将之设置为 $6 \times 64 \times 64$ 。为了存储和绘制的性能，我们采用面积加权的Haar小波来压缩系数^[61]，对每一个 ω_i 保留 $n_L = 96$ 项，这样得到的压缩结果有很好的 L^2 范数的数值误差。压缩得到的辐射传输矩阵可以得到非常高质量的绘制结果^[54, 97]。每个顶点的可见性数据需要采样 $6 \times 64 \times 64$ ，并且采用无损压缩的方式进行存储。

5.3.3.4 经过一次反射的间接光照明

每一个顶点的经过一次反射的间接光照的PTT是一个 $n_R \times n_P \times N_L \times S$ 张量，在我们的具体实现中的设置为 $n_R \leq 3$ 、 $n_P = 5$ 。为了减小存储空间，我们将 N_L 设置为较小的分辨率($n_L = 6 \times 8 \times 8$)，同样可以得到良好的绘制效果，如图5-8。即使这样减小了 N_L ，张量的数据量还是很大，因此我们采用了聚类张量分解(clustered tensor approximation, CTA)的方法进行压缩。和Tsai^[90]采用的CTA的方法相比，为了减少聚类的时间，我们并没有和他们一样采用聚类操作和张量分解交替进行的方法，而是首先采用CPCA^[83]的方法在场景中顶点的维度上进行聚类，然后每个簇中将顶点也作为一个维度就是一个三阶的张量。然后我们对每个簇的张量采用张量分解的方法进行压缩，需要说明的是，在我们的实现中，我们只压缩光照和顶点这两个维度，而另外一个维度 S 则保持不变，因为这个维度已经足够小了。

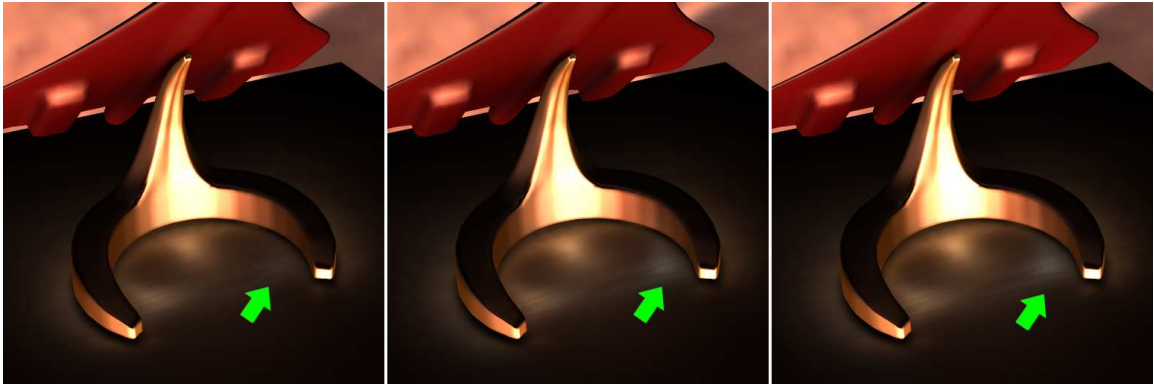


图 5-8 不同光照分辨率下的经过一次反射间接入射光照的绘制效果。从左到右， n_L 分别是 $6 \times 4 \times 4$ ， $6 \times 8 \times 8$ ，and $6 \times 16 \times 16$ 。我们可以判断 $6 \times 4 \times 4$ 的分辨率太低，因为桌面上的焦散十分暗淡。包括直接光照和两次反射的间接光照的绘制效果参看图5-13

通过这样的方法，我们能够减少经过一次反射的间接光照的PTT的存储量，相

比于^[90]我们也能很大程度上减少CTA的运算时间。尽管经过一次反射的间接光照的入射辐射亮度用低频的SH表示,一些高频的绘制效果仍然可以得到。原因主要有三点:我们采用Haar小波基作为光源的基,采用BRDF的PCA的基函数和光线跟踪来进行预计算,采用镜面项分离的方法提高张量分解的精度。不同参数设置下的CTA的数值误差和绘制结果的比较如图5-9和图5-10。在我们的具体实现中,我们采用32个特征值,光照的维度为64。当光照的维度低于64时,其对绘制性能的影响很小,而特征值的影响就很明显。我们可以得到一些经过一次反射的间接光照的高频的绘制效果,比如图5-13中所示的焦散。

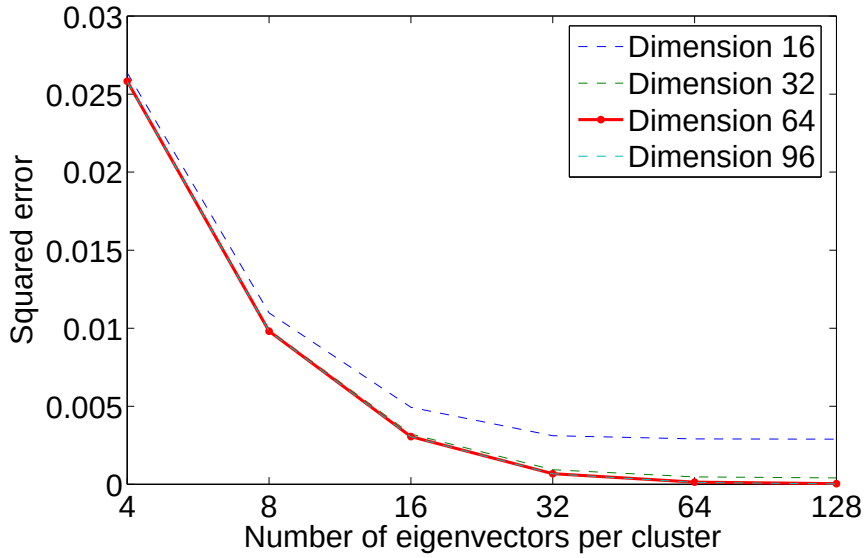


图 5-9 采用不同的光照的压缩维度及采用不同的特征向量的数量时所得到的聚簇张量近似的误差

5.3.3.5 经过两次反射的间接光照明

每个顶点经过两次反射的间接光照的PTT $T_x^{I^2}$ 是一个五阶张量。为有效减少存储空间,我们采用更小分辨率的环境光照($n_L = 6 \times 2 \times 2$)。此外,和经过一次反射的间接光照的PTT的CTA压缩相类似,我们采用CPCA进一步压缩存储空间。

我们假设经过两次反射的间接光照的入射辐射亮度在整个半球面均匀分布,在目前的实现中不考虑镜面项的绘制结果。虽然场景中还会有反射次数更多的间接光照的效果,但是通常人眼对这些效果的感受非常微弱。从原理上说,我们的框架可以扩展到更多次数反射的间接光照和更高的精度,但是需要更大的运算量。



图 5-10 不同的CTA压缩率的绘制效果(采用的特征向量的数量/光照压缩后的维度)。第一行: 8/64, 16/64, 32/16, 32/32。第二行: 32/64, 32/96, 64/64, 参考图像

5.3.3.6 预计算

PTT的预计算可以通过插值和递归的方法有效地加速计算。对于经过一次反射的间接光照的PTT的预计算, 我们对每一个顶点沿立方纹理上所有的方向发射光线, 如果和场景中物体相交, 这个交点的直接光照的入射辐射亮度能够用来计算这个交点对顶点的经过一次反射的间接光照的PTT的贡献。为了便于计算, 场景中这个交点所在的三角形的三个顶点的可见性信息可以通过重心插值近似得到该交点的可见性。如同直接光照的计算结果能用于加速经过一次反射的间接光照的PTT的预计算, 经过一次反射的间接光照的PTT的预计算结果能用于加速经过两次反射的间接光照的PTT的预计算。

5.3.3.7 绘制的性能

在绘制时, 如果光照、视点、材质都在动态改变, 那么我们所展示的这些实验结果的运算量的比例大致如下: 直接光照占25%, 经过一次反射的间接光照占70%, 经过两次反射的间接光照和镜面项的绘制占5%。性能大致和场景的规模成反比。对于直接光照而言, 性能和顶点的数量成反比。对经过一次反射的间接光照而言, 性能取决于CTA得到的张量的数量和压缩后张量的维度。对经过两次反射的间接光照而言, 性能是特征向量数量和聚类数量的函数。

绘制和预计算的性能同样取决于场景中可变材质的物体的数量。对经过一次反射的间接光照而言，性能和场景中可变材质的物体的数量成线性关系。对经过两次反射的间接光照而言，性能和场景中可变材质的物体的数量的平方成线性关系。

5.4 实验结果

我们的系统运行的硬件配置CPU是 Dual 3.20 GHz Xeon，内存 4.0 Gbytes，显卡是 Nvidia GeForce 7800 GT。辐射传输的入射辐射亮度是在CPU上完成运算，最后物体表面的着色是在GPU上完成运算。接下来是采用我们的系统绘制得到的改变BRDF、光照、视点的绘制效果。

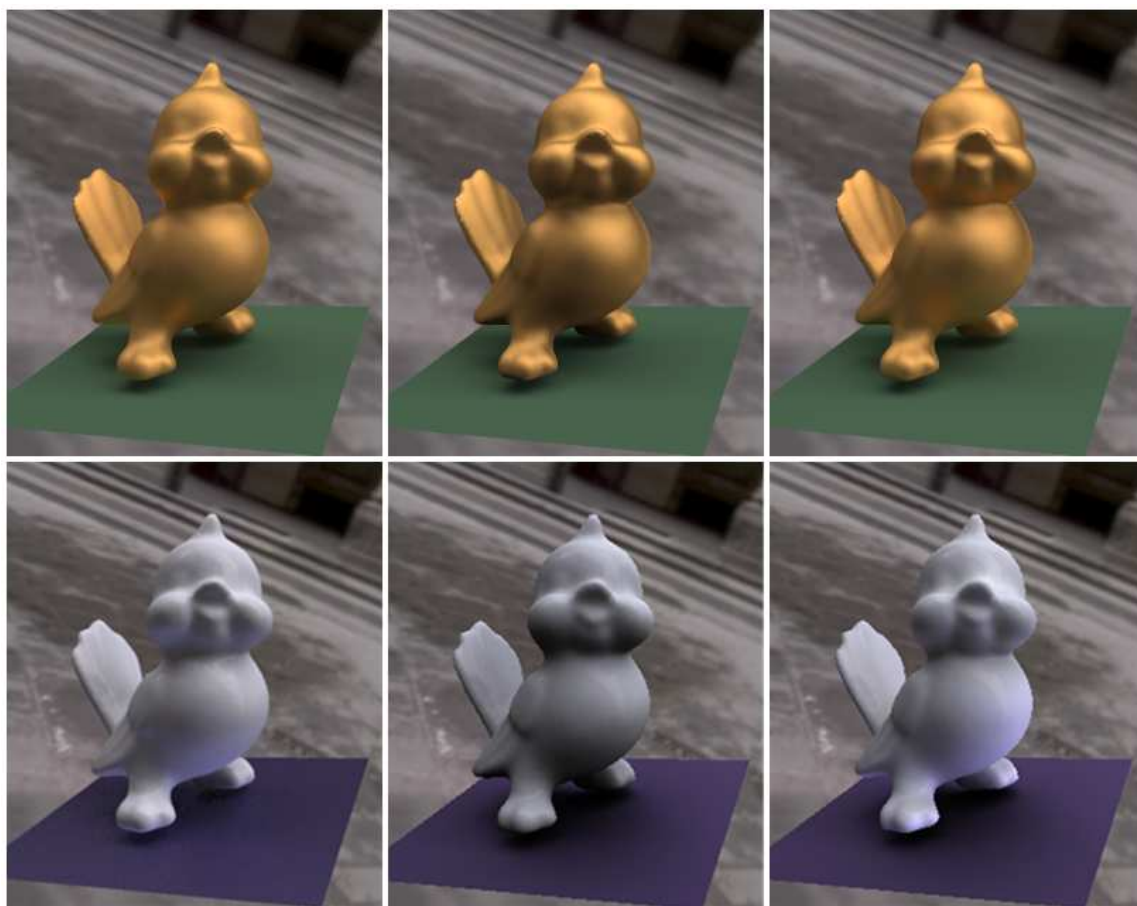


图 5-11 PTT的绘制结果和参考图像的比较。左：光线跟踪的绘制结果；中：我们方法的直接光照的绘制结果；右：我们方法包含间接光照的绘制结果。

图5-11是我们的基于PTT的绘制结果和只有直接光照的绘制结果之间的比较，以及和采用光线跟踪绘制得到的参考图象的比较。采用PTT的绘制结果包含了经

过一次和两次反射的间接光照的绘制效果，因此更接近于光线跟踪绘制的参考图像，比如Tweety胸膛上明显的渗色效果。



图 5-12 室内场景的间接光照的绘制效果。第一行：直接光照的绘制效果；包含了两次反射的间接光照的绘制效果。第二行：地板和沙发的BRDF经过修改编辑后的绘制效果。

图5-12中的室内场景一共包含12个不同的材质，有一些有纹理贴图。图中显示的结果中有的没有间接光照效果，有的有间接光照效果。采用相同的光照，但是地板和沙发采用不同的BRDF，高光材质的反射效果和间接光照效果有明显的不同，比如果场景中的渗色效果产生了剧烈的变化。

用于PTT的BRDF空间的PCA的基函数，以及BRDF空间的张量的基函数，能够提供足够的近似精度，从而得到焦散这样的高频的光照效果，如图5-13的第一排中所示。在第二排中可以看到间接光照效果的变化。当飞机的材质接近漫反射材质，绘制得到的结果中可以看到飞机投影到桌面的红色的渗色效果。随着飞机的材质的高光程度不断提高，飞机反射到桌面的辐射亮度中BRDF高光部分的作用越来越大，因此投影到桌面产生的渗色效果的颜色从红色逐渐变为白色。

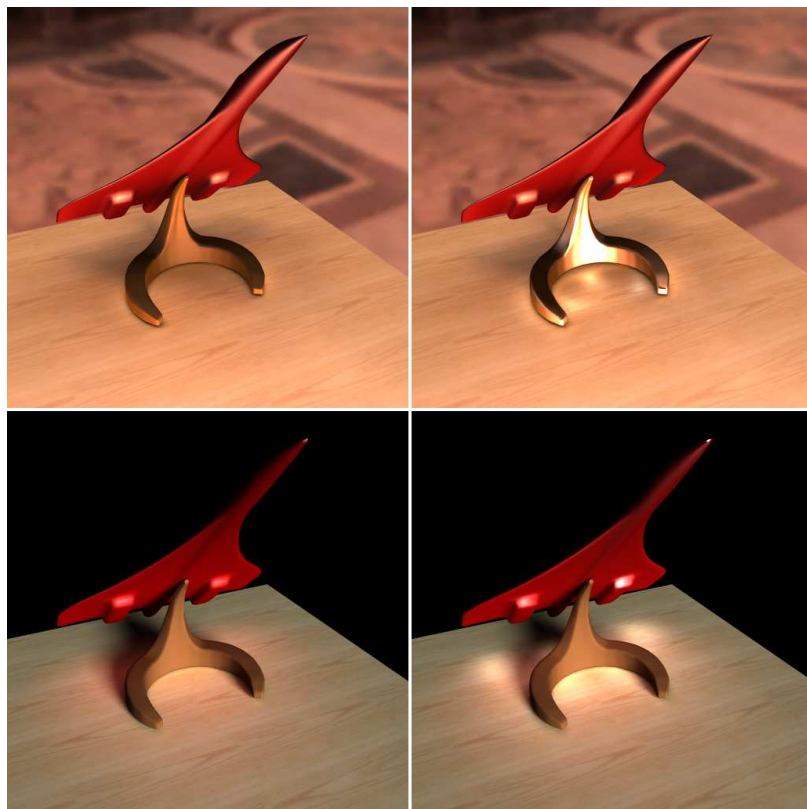


图 5-13 不同的Blinn-Phong模型的BRDF的飞机模型的绘制结果，其中BRDF的漫反射和高光部分的加权系数不变。第一行，增强了飞机模型底座BRDF的高光效果。第二行，增强了飞机的BRDF的高光效果。

如图5-14所示的手表的场景，主要是为了显示镜面项的绘制效果，以及我们的算法在材质设计和编辑方面的有效应用。这四个手表采用了四种不同的金属材质，并且得到了各异的间接光照的绘制效果。

如图5-15所示的是一个国际象棋场景。黑白双方的棋子分别被赋予两种可变材质。光照来自于远处，因此由于棋子的遮挡，如果只绘制直接光照的效果，我们会看到许多全黑的阴影区域。如果包含了由PTT产生的经过一次和两次反射的间接光照效果，被棋子遮挡而产生的阴影区域将被由棋子和棋盘反射带来的间接光照所照亮，使得这些区域也看清楚，带来更好的视觉体验。

如图5-1所示的维纳斯序列图像说明了我们的方法在BRDF动画上的有效应用。在图5-7所示的采集的到的BRDF的菲涅尔-粗糙度空间上，我们沿着一条曲线浏览BRDF空间产生了BRDF动画。在周围的墙上，我们可以看到相应的间接光照的结果。需要注意的是，由于材质变化造成的间接光照效果的变化，不只体现在墙上



图 5-14 编辑BRDF产生的间接光照效果的变化。图中手表采用了不同的金属材料。

收到的反射的颜色，而且体现在反射的辐射亮度在墙上的分布上。

表5-1列举了所有这些实验结果的内存占用和预计算、绘制的性能的数据统计。预计算的时间基本取决于场景中的顶点的数量，压缩的时间还取决于光路的数量。由于维纳斯的场景中只有一个可变材质的物体，因此所消耗的压缩的时间就要少很多。国际象棋和手表的场景中大多数的物体都没有使用纹理贴图，为了减少内存占用，我们计算只包含漫反射材质和非可变材质的光路所对应的PTT时，采用单颜色通道，而不是RGB三个颜色通道。

5.5 结论

绘制结果和场景中物体材质的非线性关系给可变材质的交互级全局光照照明绘制带来很大的困难。为了克服这个困难，我们提出了预计算传输张量的框架。在这个框架下，我们实现了可变材质的交互级重光照。采用PTT、BRDF空间的张量表述和分解、BRDF张量的镜面项分离等方法，我们不但能够产生物体表面不同的着色

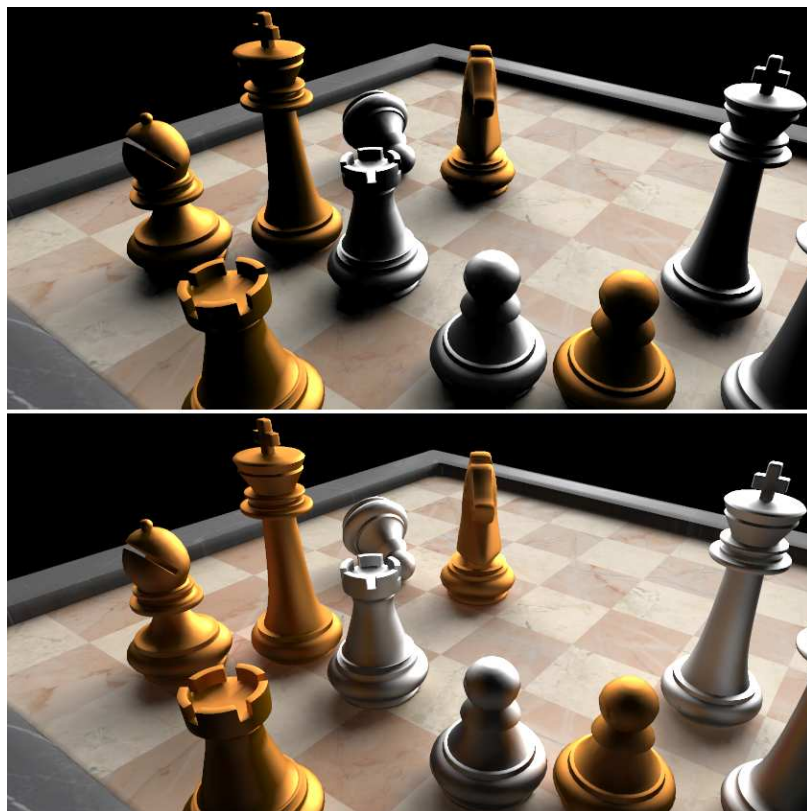


图 5-15 国际象棋。上：直接光照的绘制效果；下：采用PTT实现的包含间接光照的绘制效果。

结果，我们还能绘制生成整个场景中由于互相反射形成的间接光照的绘制效果。

为了实现材质、视点、光照都在改变时的重光照，需要处理巨大的采样空间。我们的工作证明完全可以在合理可控的时间和空间的运算代价下完成所有的预计算和绘制操作。具体的分析如下：

- 绘制性能：当BRDF、光照、视点都在改变时，我们所有的实验结果的绘制性能为1.5–5.9 fps。在许多情况下，这三个因素中同时只有一个会被改变，比如场景设计时所遇到的材质编辑的应用。如表5-1所示，在这种情况下性能可以得到极大的提高。我们注意到我们的绘制操作并行度很高，因此非常适用于采用多CPU进行多线程的加速，这同样可以进一步提高绘制性能。
- 预计算：预计算操作同样可以通过并行的策略得到进一步的加速。并行的加速手段在其他的PRT算法中也得到了应用。比如Kristensen^[49]采用了32个2.8 GHz CPU在4个小时内完成了32K个三角形的预计算。我们的预计算尚未在如此规模的集群上进行加速测试。

表 5-1 实验结果的性能数据。在BRDF空间的表示中，我们用‘BP’表示Blinn-Phong，用‘CT’表示Cook-Torrance。‘V’表示只有视点改变时的绘制性能；‘V&B’表示只有视点和BRDF改变时的绘制性能；‘V,B&L’表示视点、BRDF和光照都改变时的绘制性能。所有时间结果中直接光照的分辨率都是 $64 \times 64 \times 6$ 。

场景	Tweety 图5-11	飞机 图5-13	室内 图5-12	维纳斯 图5-1	国际象棋 图5-15	手表 图5-14
BRDF空间	<i>CT</i>	<i>BP</i>	<i>BP</i>	<i>CT</i>	<i>CT</i>	<i>CT</i>
可变材质物体数量	2	2	2	1	2	2
顶点数量	13.0K	131.1K	35.7K	39.7K	53.9K	62.1K
聚类数量	13	13	35	40	54	62
特征向量的数量	32	32	32	16	16	16
内存占用(Mbytes)	148	144	311	389	539	603
预计算时间(小时数)	2.6	2.5	4.2	5.0	13.7	15.7
压缩时间(小时数)	4.0	6.4	15.1	7.3	10.5	14.3
V(fps)	121.58	139.50	74.52	57.09	48.30	43.06
V&B(fps)	28.73	30.79	12.57	10.94	8.23	7.23
V, B&L(fps)	5.82	5.91	3.40	2.62	1.67	1.53

- 内存占用：由于我们的算法，能在绘制时同时改变BRDF、光照和视点，并得到高频的绘制效果，因此较大的内存占用是非常合理的而且是不可避免的。即使如此，我们的算法的内存占用和其他全频率的PRT的重光照算法的内存占用相当^[61, 62, 97]。

第6章 可变透明和散射材质的交互级全局光照明绘制

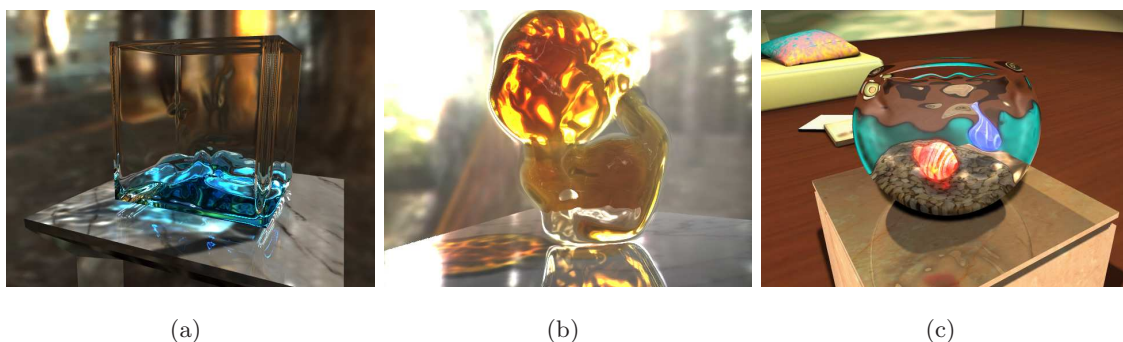


图 6-1 我们的交互级绘制算法得到的绘制结果

在本章中，我们提出了一种可变透明和散射材质物体的交互级重光照绘制算法。算法适用于均质和非均质的物体材质，比如空间动态的折射率和各项异性的散射，并能得到各种折射和散射形成的丰富的绘制效果。用户还可以通过交互改变物体几何、光照、材质和视点。

在第4章和第5章我们所研究的绘制场景中，几何物体采用的是反射材质。除了反射材质，在自然界中还有两种常见而美观的材质：透明和散射材质。透明材质指的是光经过折射进入物体内部继续传播，主要体现的是折射的物理现象。散射材质指的是光在空气或者物体内部传播时，由于参与介质的作用，改变了最终出射辐射的亮度和方向。透明和散射材质能够产生许多美丽的光照效果，比如透明材质可以形成海市蜃楼、日落等大气现象，还有水晶花瓶、水波纹生成的不同形态的焦散；散射材质可以形成烟尘、体阴影、体焦散的光照效果。

透明材质和散射材质的绘制与反射材质的绘制有很大的不同。透明材质的绘制有两个主要特点：第一，折射现象的绘制结果都非常的锐利和高频，一般的近似和优化算法带来的误差在视觉效果上非常敏感；第二，光传播到物体表面时绝大部分的辐射强度都发生折射或镜面反射，因此不需要使用BRDF，在绘制时也就不需要在物体表面的光的入射点上进行半球面入射辐射亮度的积分。散射材质的绘制最主要的特点是：光在传播过程中，并不只在物体表面发生变化，而是在光所能到达的所有位置都可能因为参与介质的作用发生各种改变，因此运算的复杂度要远高于其它类型的物体材质。如果一个物体内部或者空气中的材质处处相同，我们称之为均

质材质，否则我们称之为非均质材质。显然，均质材质绘制的复杂度要低于非均质材质，因为我们可以将其简化为只在不同材质的交界面上计算折射、散射和消散的绘制效果。而非均质材质绘制过程中，必须考虑光在整个传播路径上的连续变化。

由于上述的原因，目前尚无其他算法，可以实时或交互级绘制非均质的透明和散射材质的全局光照明效果。在本章中，我们提出的交互级全局光照明绘制算法，能够绘制折射、反射、消散、单次散射、面阴影、体阴影、面焦散、体焦散等多种光照效果。用户可以通过交互改变物体几何、光照、材质和视点。

我们的算法包含一个完整的绘制算法流水线。通过这个绘制算法流水线：我们首先将物体几何转化为体数据；然后沿着曲线路径跟踪每个光子在这个体数据中的折射路径，以及由于消散造成的能量的衰减；最后逐像素地绘制在任意视点下所见到的辐射亮度分布，从而得到最后的绘制结果。不同的交互只需要改变流水线中不同步骤地输入，就能分别得到动态几何、动态光照、可变材质和动态视点的绘制效果，并且不需要为此进行任何预计算。流水线的所有步骤都在GPU上实现，能够达到交互级的绘制性能。除了重光照和材质编辑外，我们的算法可以作为任何交互级物理仿真或几何变形算法的快速可视化的方法。

6.1 算法概述

最漂亮而有趣的光照现象是由间接光照明产生的，因此我们要求解的是一个全局光照的绘制问题。事实上，最大的困难就是我们的全局光照的绘制方法必须足够精确，从而能获取十分高频且细节丰富的焦散的绘制效果，并且绘制的性能要足够快，以保证用户可以同时改变光照、几何、材质和视点。一些全局光照明算法，比如分布光线跟踪、蒙特卡罗光线跟踪、光子贴图已经被广泛运用，并能获得非常逼真的绘制效果，但是这些离线算法都难以达到交互级的绘制速度。虽然GPU的应用能提供极大的加速，比如光子贴图^[73]和波阵面跟踪^[43]，但是还是无法达到交互级重光照的性能。还有一些方法通过一些近似的策略能够得到实时的绘制性能^[18, 25, 50, 67, 103-105]，但是这些方法都不可避免地缺失了一些重要的全局光照的绘制效果。简而言之，目前尚无一种方法能够很好地交互级绘制可变折射和散射材质的全局光照的重光照效果。

在本章中，我们提出了一个新的绘制算法，绘制场景中的物体可以有任意变化的、处处不同的折射率、消散系数和散射系数。我们的算法能够在场景改变的同时以交互级的性能绘制焦散、散射、消散等效果，目前没有任何其它算法能得到同样

的绘制性能和绘制效果。我们提出的算法是一条新的绘制算法流水线，流水线的每一个步骤都在GPU上实现，最后达到交互级的绘制性能，并且不需要任何预计算。我们能够达到交互级的绘制性能，一个关键之处在于采用体数据表示和存储物体和光照。我们设计的流水线的第一步就是将物体几何快速地体素化为体数据，这样即使我们改变的是物体几何，我们也同样可以维护得到体数据的改变。采用这种体数据对场景和光照的表示和存储，我们的算法流水线的后续的所有步骤都能很方便地利用GPU的并行性来加速运算。

绘制每一帧时，我们的算法流水线首先将输入的物体几何体素化为体数据，再通过光子发射和跟踪计算体数据中的辐射亮度的分布，最后逐像素绘制任意视点下的全局光照明的效果。由于我们允许场景的折射率可以连续变化，因此我们必须能沿着曲线的折射路径跟踪光子以及从视点发出的跟踪光线，而其曲线的折射光路由几何光学的基本方程、即程函方程决定。同时，我们还需要根据材质的消散系数计算光子能量的变化。因为我们的绘制算法是基于体数据的表示，因此我们可以为这个体数据建立八叉树来进一步加速我们的算法；我们在沿着曲线的折射路径跟踪光子时采用这个八叉树的数据结构自适应地选择每步前进的步长。在光子跟踪的每一步中，我们利用GPU的光栅化操作的性能优势来加速在体数据中更新辐射亮度的操作。光子所携带的辐射能量存储在所经过的体素中。最后，我们逐像素进行绘制，为每个像素发射一条跟踪光线，同样根据程函方程计算其折射路径，并根据散射系数计算经过的像素所存储的能量带来的辐射亮度，再叠加可见的环境光的辐射亮度，就得到了最后的绘制结果。如6-1中所示的是我们的一些实验结果，我们的算法流水线在绘制这些场景时可以达到每秒数帧的绘制性能。

6.2 绘制流水线

我们提出的算法，能够交互级绘制动态场景的折射、消散、各项异性的散射等全局全光照明效果。我们的绘制算法流水线如图6-2所示，整个流水线包括数个具体阶段步骤，它们的每一个都在GPU上实现。绘制算法流水线以场景中的物体几何作为输入，并且制定一些点光源或者方向光源以及视点方向；最终输出得到的全局光照明的绘制结果图像。需要注意的是，如果用户通过交互改变视点方向、光照条件、场景几何或材质参数时，往往只有一部分的流水线需要重新计算运行就能得到更新后的绘制结果。

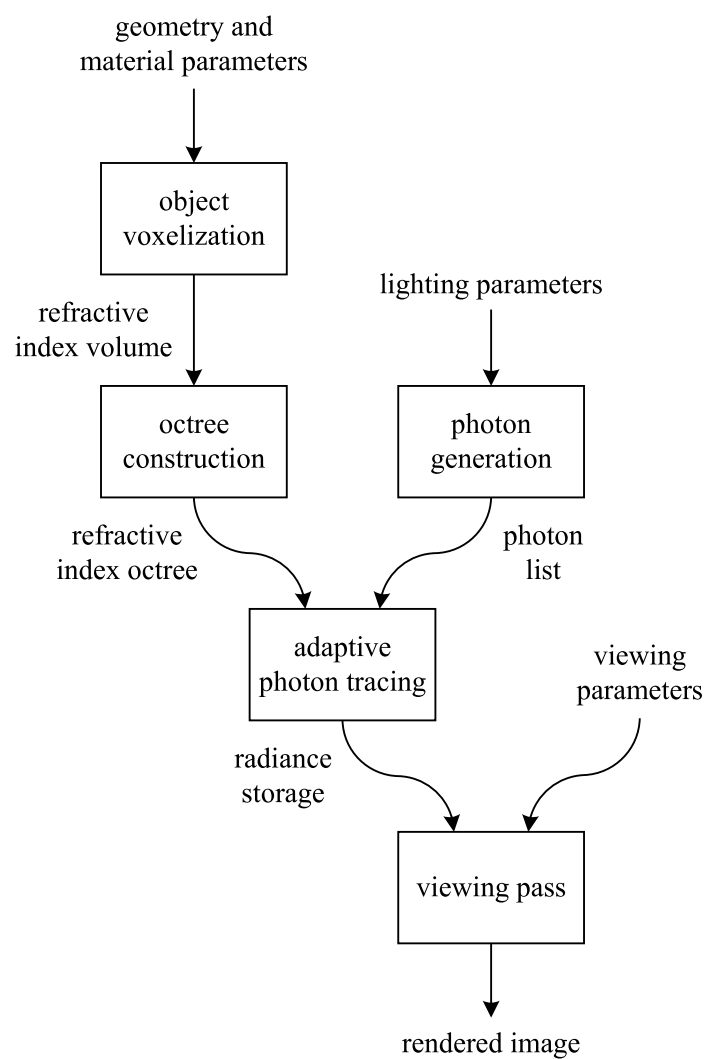


图 6-2 我们的绘制算法流水线

我们的绘制算法依赖于对场景的体数据的表示, 这个体数据由大量的均匀分布的立方体的体素组成; 由于这个特点, 我们用这些体素来记录和表示场景整个空间中各个位置的折射率 n 、散射系数 σ 和消散系数(即散射系数 σ 和吸收系数 α 的和)。我们的绘制算法流水线的第一步就是利用GPU的并行性非常高效地将用三角形网格面片表示的物体几何体素化为体数据。对于一些应用, 比如仿真和物理测量, 我们可直接得到体数据, 那么我们就可以跳过体素化的步骤直接进入流水线的下一步。

算法流水线的后一个步骤, 是通过分析体数据中的折射率为整个场景建立八叉树数据结构的材质表示, 八叉树的每个节点表示在这个节点所在的空间区域内, 折射率几乎是常数。这个八叉树结构和我们为光源生成的光子一样, 作为流水线后面的自适应非线性光子贴图算法步骤的输入数据。在流水线的自适应非线性光子贴图算法步骤中, 光子沿着它的折射光路在场景中传播, 并在它所穿越过的体素上存储它所代表的辐射传输。在最后绘制效果图像的步骤中, 我们从视点对每个像素发出跟踪光线, 光线在场景的整个体数据中沿着折射路径穿越场景, 光线所经过的所有体素所存储的经过光子的辐射传输都被收集加和, 从而得到这根光线生成的在视点所看到的辐射亮度。

因为我们的绘制场景中整个空间的折射率可以连续变化, 所以光子和跟踪光线都沿着曲线的传播路径在场景中传播, 而不像传统的光子贴图算法中沿着直线传播, 也不是简单地沿着折线传播。曲线的传播光路 $\mathbf{x}(s)$ 由场景中的折射率的标量场 n 所决定, 具体由几何光学的基本光学公式, 即程函方程^[8]决定:

$$\frac{d}{ds} \left(n \frac{d\mathbf{x}}{ds} \right) = \nabla n \quad (6-1)$$

如果定义 $\mathbf{v} = n \frac{d\mathbf{x}}{ds}$, 我们可以将式6-1写成一阶微分系统的形式:

$$\frac{d\mathbf{x}}{ds} = \frac{\mathbf{v}}{n} \quad (6-2)$$

$$\frac{d\mathbf{v}}{ds} = \nabla n \quad (6-3)$$

我们将用许多小步长的折线来近似一条光滑的曲线作为折射传播的光路，因此我们可以继续将其写作前向差分(forward-difference)的离散形式：

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \frac{\Delta s}{n} \mathbf{v}_i \quad (6-4)$$

$$\mathbf{v}_{i+1} = \mathbf{v}_i + \Delta s \nabla n \quad (6-5)$$

和同领域之前的工作不同，我们在沿曲线光路跟踪光子的时候，会根据光子目前位置周围区域的折射率分布自适应地选择下一步前进的步长。在折射率几乎不变的空间区域，我们可以用很长的步长一步就能穿越整个区域，而如果这个区域的折射率变化剧烈，我们就要采用较小的步长，较多的步数穿越这个区域，从而能精确地逼近真实的传播路径。我们利用八叉树的数据结构帮助我们在场景空间中的任何一点确定下一步前进的步长。

6.3 绘制流水线的各个步骤

在这部分，我们将详细介绍图6-2所示的绘制算法流水线的各个阶段步骤。

6.3.1 体素化

我们的体素化方法基于Crane^[17]的方法。他们的算法中以密闭的三角形几何网格作为输入，利用GPU的光栅化和裁减操作计算每一个体素中心的位置是在几何体的内部还是外部，如果在内部则赋值为1，如果在外部则赋值为0，每个体素的计算结果放在一起作为一个三维纹理输出。体数据中每一层作为绘制的一帧依次绘制整个几何体，绘制时的近裁减平面设置为该层靠近视点的边缘平面，在片断着色程序中，如果所绘制的片断是三角形的反面，则像素的数值加1，如果是三角形的正面，则像素的数值减1。我们对他们的这个体素化算法作了一些改动，因为在我们的算法流水线中要求体数据中的折射率平滑连续地变化，以避免不必要的绘制的瑕疵，这和基于程函方程的折射物体的绘制方法^[43]中的情况相同。

如果几何物体的表面穿过体数据中的一个体素，那么我们的体素化方法必须为这个体素制定一个浮点值，以表示该体素在这个几何体内部或外部的程度。为此我们首先将这个体数据超采样到四倍的分辨率再进行体素化操作，然后将体素化后的结果降采样到原始的分辨率。这样在降采样的操作时，我们需要为每个最终输出的

体素进行 $4 \times 4 \times 4 = 64$ 次纹理读取的操作, 这代价非常巨大, 对绘制性能有很大的负面影响。因此我们采用的策略是指对几何物体表面边缘附近的体素进行降采样。我们首先在最终输出的分辨率下进行体素化, 然后每个体素对周围的体素进行 $3 \times 3 \times 3$ 次纹理读取的操作, 以此确定这个体素是不是物体几何表面边缘的体素。如果这27个体素都是相同的值, 那么不需要更多的操作就能确定该体素的体素化的结果。如果这些体素中有不同的值, 那么我们就对这个体素对应的超采样结果中对应的区域进行降采样, 得到这个体素的浮点数的体素化结果。

最终, 我们体素化得到的浮点值的结果转化为折射率数值, 然后用大小范围是 $7 \times 7 \times 7$ 或 $9 \times 9 \times 9$ 体素的高斯函数进行模糊和平滑。需要指出的是, 体素化过程中的超采样方法能有效提高体数据对物体几何表面法向的描述(即使用折射率的梯度来表示), 同时模糊操作将物体几何表面边缘狭小的区域扩大到一层过渡带。我们体素化的算法的结果在图6-3中直观可见。

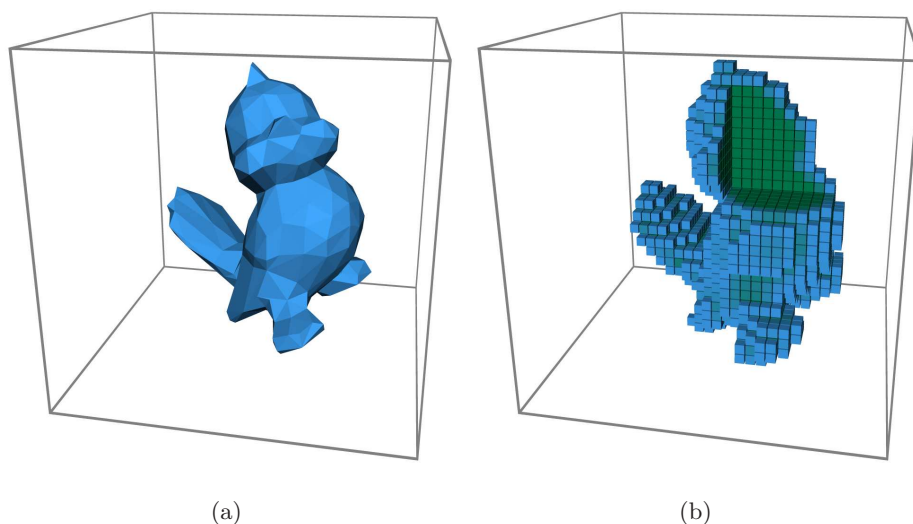


图 6-3 (a) 一个物体的三角形面片的表示。(b) 将这个物体体素化为体数据后的结果。

6.3.2 八叉树的建立

建立八叉树的算法的输入是一个阈值 ϵ 和一个记录场景空间折射率 n 的体数据, 体数据的每一个体素记录的数值都代表了相同大小的一个立方体空间区域的折射率。最终的输出是一个八叉树, 它的每一个叶结点代表了在该节点所覆盖空间区域内, 折射率是常数或者折射率的最大差异不超过阈值 ϵ 。由于我们的算法是在GPU上实现的而不是在CPU上实现的, 因此我们选择的八叉树的数据结构便于

使用GPU的并行处理单元来建树和查询。传统的八叉树采用稀疏存储的方法，每个非叶结点都要存储8个指针指向它的8个子结点。我们的八叉树数据结构采用一个三维数组来密集存储，这个三维数组中的每一个数代表了所对应的体素的空间在八叉树中所存在的叶结点的层次位置。虽然这种密集的三维数组的存储方法相比于传统八叉树的稀疏存储要占用更多的空间，但是它更便于使用GPU来建树和查询。

我们的八叉树数据结构的建树方法和mipmap的建立过程，以及Ziegler^[109]的基于柱状图的金字塔结构的建立过程比较类似。图6-4中直观地显示了整个建树过程，为了方便表示，图中才用了二维平面上四叉树的建立来作类比。我们首先建立一个三维数组的金字塔结构，这个结构中的每一层的每一个数据都代表了其所对用空间区域的折射率的最大值和最小值。然后我们使用这个金字塔数据结构中的结果以及输入的阈值 ε 来进行比较，从而确定八叉树中的每一层的节点如果作为八叉树的叶结点，是否足以表示其所覆盖的所有空间的折射率分布，换句话说，需要确定该空间内的折射率是否都是常数或者最大差异没有超过阈值 ε 。

为了更清楚的描述我们的算法，我们看一个具体的例子。假设我们有一个体数据，边长为 2^K ，折射率记作 n 。我们首先建立一个包含 K 层三维数组的金字塔数据结构，这个结构和mipmap的结构十分类似，只是每一层中每一个元素记录的都是其所对应的空间区域的折射率的最大值和最小值($n_{\min}$ and $n_{\max}$)。然后我们建立另外一个边长为 2^K 的三位数组，每一个元素都是一个整数，用来表示所对应的体素在八叉树结构中所在的叶结点所在的层数，即在该叶结点下的所有体素的折射率的差异在阈值 ε 范围内。我们用0标记八叉树中分辨率最大的层次，用 K 标记八叉树中分辨率最小的层次(即只有一个叶结点)。我们从分辨率最小的层次到分辨率最大的层次依次比较其对应的折射率的变化范围和输入的阈值 ε 来确定八叉树的叶结点。一旦测试结果显示金字塔结构的第 k 层满足条件 $n_{\max} - n_{\min} < \varepsilon$ ，我们就把最终输出的三维数组中这个叶结点覆盖的所有的元素都赋值为 k 。一旦一个体素对应的输出结果被赋值，我们就不需要对这个体素作更多的操作和赋值。

从原理上说，我们建立八叉树的过程中不但要考虑折射率 n 的变化范围，还需要考虑消散系数 κ 的变化范围，在每个叶结点内，这两个参数都要保证基本不变，这是因为我们的自适应非线性光子贴图算法中辐射亮度的计算同时受到这两个材质参数的影响。为此我们之前描述的八叉树的建立算法可以非常容易地扩展到消散系数，只需要在确定是否建立一个叶结点时，不但比较折射率的变化范围和输入的阈值，还要比较消散系数的变化范围和为此输入的另外一个阈值。然而事实上，在我

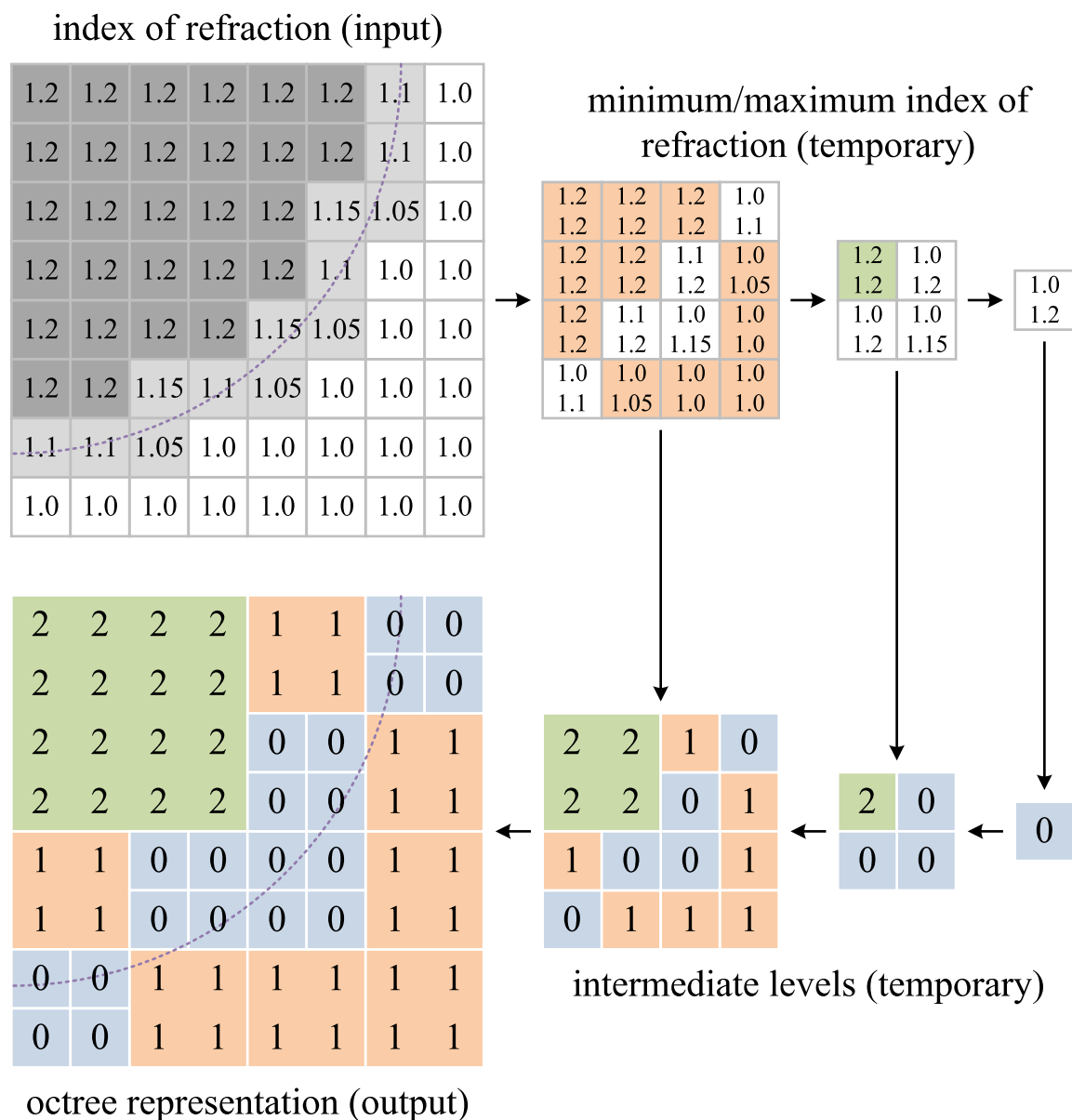


图 6-4 八叉树的建树算法，为了便于描述，采用二维的形式。我们首先建立金字塔结构记录最小和最大的折射率。然后自顶向下分析，折射率变化范围不大于阈值 $\varepsilon = 0.05$ 的，就将当前的层数作为所对应范围内每个体素的层数。

们的实验中消散系数的影响非常不明显,从而将八叉树的建立算法扩展到支持消散系数变化范围的约束,对整个绘制算法流水线的性能基本没有影响。

6.3.3 光子的生成

我们采用和阴影贴图类似的方法生成所有光子的初始位置和速度:我们将视点设置在光源位置,面向场景中心,然后绘制整个场景的包围盒。在片断着色程序中,如果处理的片断是三角形的正面,那么绘制结果中的alpha通道是赋值为1,RGB三个通道记录了该片断的空间位置。然后,我们将这张绘制得到的图象转换为光源发出的光子序列,这个操作可以用几何着色程序实现(geometry shader),或者使用一些通用计算的GPU编程语言实现的“扫描(scan)”操作^[11],比如CUDA^[66]。如果一个像素的alpha通道的值不是0,那么这个像素就生成一个光子,光子的位置由这个像素的RGB值得到,这个光子的方向由光子的位置和光源的位置计算得到。这个光子所携带的辐射能量由光源的属性决定。

当绘制场景由一个或几个透明的物体组成时,物体周围的很多区域都是空的,其折射率也都恒为1。为了除去在这些空区域中光子跟踪的时间,我们尽量把光子在接近物体几何表面的区域生成。因此,我们在采用类似阴影贴图的方法生成光子的时候,我们并不是绘制场景的包围盒,而是绘制场景中折射物体的代理几何体(proxy geometry);如果是一个非常复杂的几何物体,我们可以使用任何一个完全包含折射物体的简化的几何体。

6.3.4 自适应的非线性光子贴图

绘制算法流水线中自适应非线性光子贴图算法的步骤的主要目的,是将生成的光子沿着它的曲线的折射光路传播,同时将自身携带的辐射亮度存储在所经过的体素。自适应光子贴图算法的输入是根据折射率建立的八叉树数据结构、消散系数的RGB三通道的三维数组以及生成的光子序列。每个光子的数据结构包括它的初始位置 $\mathbf{x}_0$ 、初始速度 $\mathbf{v}_0$ 和RGB三通道的初始辐射亮度 $\tilde{L}_0$ 。算法最终输出的是一个RGB三通道的三维数组,记录了体数据中的每个体素收到的辐射亮度的分布。

在光子传播的循环中,我们根据式6-4和式6-5 计算并更新光子的位置和速度。针对每一个光子,我们使用八叉树确定光子在当前所在的八叉树的叶结点中传播的最大步长,也就是在折射率基本不变的区域传播的最大步长,如图6-5所示。我们选择 Δs_{octree} 和用户指定的最短步长 Δs_{min} 中较大的一个作为光子传播的步长 Δs 。最短步长通常是一到两个体素的宽度,用户可以根据自己对绘制性能和绘制质量的权

衡调整最短步长的大小。

在光子传播的每一步中，光子都会因为吸收和散射损失一部分的能量。这种辐射能量指数方式的衰减取决于所在的局部的消散系数，这里我们假设八叉树的同一个叶结点内部的消散系数几乎为常数：

$$\tilde{L}_{i+1} = \tilde{L}_i e^{-\kappa(\mathbf{x}_i) \|\mathbf{x}_{i+1} - \mathbf{x}_i\|} \quad (6-6)$$

我们已经描述过了光子传播中的每一步根据式6-4 和6-5计算更新光子的位置和速度，以及根据式6-6更新光子携带的辐射亮度。现在我们介绍如何在体数据的每个体素中存储辐射传输。在每个体素中，我们将所有经过的光子的辐射亮度和传输方向，根据其辐射亮度进行加权求和，以此作为该体素的辐射传输的近似来存储，并且存储RGB三个颜色通道的辐射亮度，不考虑波长不同引起的区别。这显然是一个对真实的辐射传输的相当粗糙的近似，但是已经足以较精确地绘制重现许多全局光照明的效果。

当光子从位置 $\mathbf{x}_i$ 传播至位置 $\mathbf{x}_{i+1}$ ，它对其所经过的所有体素都有辐射传输的辐射亮度的贡献。因此，当我们完成一个光子的一步传播，我们就在顶点缓存中增加两个顶点，两个顶点的位置分别是光子传播的前后两个位置，还需要记录其对应的传播速度和辐射亮度。一旦将所有的光子都向前传播一步，每个光子都对应两个顶

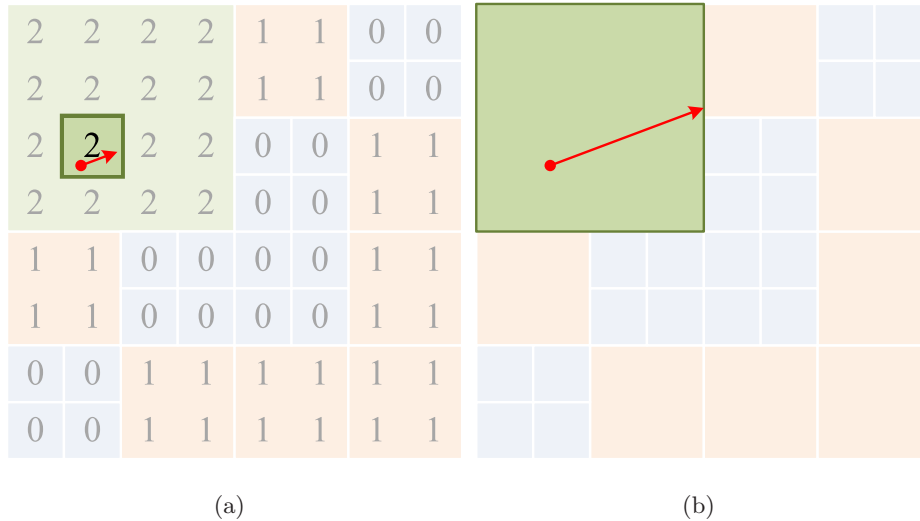


图 6-5 使用八叉树的自适应光子传播。(a)光子的位置可以确定所在八叉树的叶节点在八叉树中的层数。(b)传播步长 Δs_{octree} 确保光子直接传播到达当前叶结点的边缘。

点,我们将整个顶点缓存当作线段序列计算所有光子的这步传播所产生的辐射传输的贡献。我们利用硬件图形流水线来绘制这些线段序列,从而插值这步传播前后的光子的位置、速度、辐射强度。我们采用片断着色程序将每个光子的辐射传输的贡献存储在每个像素中,每个光子的传播方向为所有经过的光子的传播速度的加权的方向,其中权重是每个光子的RGB三个通道的辐射亮度的和。

当我们将所有的光子向前传播一步,并且将所有贡献的辐射亮度存储在体数据中之后,我们把所有离开场景的包围盒的光子以及辐射亮度低于一个给定阈值的光子剔除,然后我们重复之前的操作,把所有剩余的光子继续向前传播一步。我们循环重复这样的光子传播过程,直到剩下很小一部分的光子(一般是最初生成的光子数量的 $1/1000$)停止光子传播的操作。作为自适应非线性光子贴图的最后一步,我们将存储辐射传输分布的体数据用大小为 $3 \times 3 \times 3$ 体素的高斯函数平滑去噪。

我们的自适应非线性光子贴图算法和传统的体数据的光子贴图算法有一些十分重要而关键的差异和改进。第一点也是最重要的一点,我们的方法在体数据中跟踪光子传播的时候采用自适应的传播步长;如前文所述,我们的八叉树数据结构能帮助我们计算在折射率基本不变的区域内所能前进的最长步长。第二点,我们用大量的体素网格来存储每个光子在体数据中传播时贡献的辐射亮度。我们在光子所经过的每个体素都存储光子的辐射亮度,而不是只在发生散射等事件的位置存储该光子的辐射亮度。传统的体数据的光子贴图算法采用Kd-tree的稀疏存储结构存储光子传播的辐射传输,在逐像素的绘制过程中还要进行邻近光子收集的运算操作,这步操作的运算代价非常高。

6.3.5 逐像素的绘制

光子传播的步骤完成之后,我们就能绘制任意视点下的效果图像。我们从视点在每个像素发出一条跟踪光线,在场景中沿曲线的折射光路传播,并收集所经过的体素中存储的辐射亮度。一个体素中的辐射亮度根据该处的散射系数和相位函数(phase function)得到该体素单次散射得到的辐射亮度,然后根据从视点到该像素的折射路径经过的所有体素的消散系数,计算得到从该体素到达视点的辐射亮度。

我们沿着最后输出的绘制图像的每个像素发出一根跟踪光线。如果这根光线和场景的包围盒相交,我们就继续沿着它的折射光路一步一步的跟踪,直到这根光线穿过这个场景。和光子跟踪一样,我们还是按照式6-4和式6-5 计算跟踪光线的传播路径。我们在跟踪从视点发出的光线时,我们采用的固定的传播步长为1个体素的

宽度，并没有采用自适应的方法动态地确定传播步长。采用固定的传播步长主要基于以下几个原因：我们不希望丢失传播路径上某些体素贡献的辐射亮度；我们也不希望因为动态的传播步长而给最终的绘制图像引入不必要的瑕疵；此外，基于具体实现的考虑，相比于光子传播和辐射亮度存储的步骤，逐像素绘制时的计算代价要小很多，因此没有必要通过加速逐像素的绘制来提高整个绘制流水线的性能。

在完成跟踪光线传播的每一步后，我们从存储辐射传输的体素中读取这里存储的辐射亮度及其方向，然后根据该处的相位函数确定其中有多少由于散射沿着跟踪光线的路径射向视点。然后我们将这散射出来的辐射亮度和这里的散射系数 $\sigma(\mathbf{x})$ 相乘，还要和沿着从这里到达视点的传播路径由于吸收和散射造成的总的衰减系数相乘。这样得到的乘积是这个体素最终对视点的辐射亮度的贡献，我们将经过的所有像素计算得到的辐射亮度相加，就得到了这条光线由于散射得到的总的辐射亮度。一旦这条跟踪光线穿过场景，我们只需要再加上所对方向的环境光照的辐射亮度就完成了这个像素的绘制。

6.4 算法实现

在这部分中我们介绍了在算法具体实现中需要注意的一些问题。虽然这些问题都非常重要，但是他们并不会影响读者对整个绘制算法流水线工作原理的理解。

6.4.1 辐射亮度的存储

我们在GPU上实现的自适应光子贴图算法可以分为两个步骤，第一步是光子传播，即计算每步传播后光子新的位置、传播速度和辐射亮度，第二步是辐射存储，即在体数据中存储这步光子传播所贡献的辐射亮度。我们最早采用GPU的绘制流水线实现自适应光子贴图算法，然后用CUDA重新实现完全相同的算法，经过比较发现绘制性能提高了15%至25%。在辐射存储的步骤中，我们利用绘制流水线的光栅化操作将线段序列存储到记录辐射传输分布的三维纹理中。

我们的方法遇到的局限性在于目前的GPU只能在三维纹理的某一层，也就是一张二维纹理上进行线段序列的光栅化操作。为此，我们不得不采用比八叉树或其它方式得到的步长更短的步长进行光子跟踪。这是一个非常关键的局限性，因为光子贴图是我们的算法流水线中运算代价最高的环节，而这个环节的运算时间和所有光子的传播的总的步数成线形关系。因此，我们采用以下两个策略来缓解这个问题：

- (1) 我们存储光子的辐射亮度存储的三维纹理并不是所有切片(slice)的方向都是相同的。我们使用的三维纹理的切片数量是体数据分辨率的三倍, 这三部分的切片的朝向各不相同(这三部分的切片的法向分别是空间中的 x 轴、 y 轴和 z 轴)。一个光子传播一步之前, 我们需要根据光子传播的方向选择存储的切片, 从而尽量保证在这部分切片平行的方向上光子传播的步长最长。
- (2) 我们将每张切片的厚度加倍, 然后采用两张三维纹理作为两个绘制目标(render target)。一个绘制目标记录索引为奇数的切片, 而另一个绘制目标记录索引为偶数的切片(对于三个朝向的切片都采用同样的方法)。当光栅化一条线段时, 我们同时绘制到相邻的两张索引分别为奇数和偶数的切片中, 并且在片断着色程序中确定哪个绘制目标的像素要着色, 而另外一个则不进行着色。

最终, 我们使用了四个绘制目标, 因为我们必须不同的绘制目标中存储辐射亮度和辐射方向, 此外我们还将索引为奇数和偶数的切片分离成两个绘制目标。和光子传播的步长的增加带来的性能上的提高相比, 采用多个绘制目标以及三倍的切片的额外代价可以被忽略。

我们希望有朝一日, 硬件图形流水线能够支持在三维纹理上进行光栅化操作, 而不只局限于二维的纹理上, 这样我们能够避免采用三倍切片和多个绘制目标这样的方法带来的开销, 光子传播的步长也不会因此受到限制。作为另一种选择, 如果CUDA能够提供高效的纹理读写的原子操作, 我们就能使用CUDA实现线段序列的体素化操作, 而不用担心多个光子同时更新同一个体素的辐射传输会带来并行操作引起的错误。如果能够使用CUDA完成线段的体素化, 我们还能避免为了CUDA和OpenGL之间共享光子数据而维护的一个巨大的顶点缓存。

6.4.2 GPU实现中CUDA和OpenGL的选用

在使用GPU的图形流水线实现我们的算法时, 对于是采用CUDA(或者类似的通用计算的语言)还是OpenGL(或者是类似的图形绘制的API), 需要做许多的权衡。在实现一些不容易被分解为顶点着色、几何着色、片断着色等多个部分的算法时, CUDA无疑具有非常明显的优势, 它在实现通用计算的算法时非常有效, 不需要我们将算法生硬的分为多个步骤。同时我们如果不需要光栅化或者纹理滤波等操作时, CUDA也能很方便地避免多余的运算操作。

但是, 当我们要将通用计算和图形绘制流水线混合使用时, 这种通用运算带来的优点也会遇到额外的代价。由于当前的CUDA和OpenGL之间只能共享一维的缓存, 而不能共享二维纹理和三维纹理, 因此在我们目前实现的绘制算法流水线里, 在CUDA和OpenGL之间拷贝必须共享的数据所消耗的时间大概占总绘制时间的10%。我们采用CUDA来实现八叉树的建树操作, 从阴影贴图生成光子, 以及在体数据中跟踪光子传播; 我们采用OpenGL实现物体几何的体素化, 存储光子传播贡献的辐射亮度, 以及绘制从视点发出的跟踪光线得到的辐射亮度。由于这些操作交叉进行, 我们需要从OpenGL向CUDA拷贝阴影贴图、折射率的体数据以及消散系数的体数据。幸运的是, 我们可以在CUDA实现的光子传播操作后能在OpenGL实现的辐射存储中共享光子的线段序列, 这是因为这个线段序列所在的顶点缓存是一个一维的缓存。我们希望在将来CUDA和OpenGL之间能够共享使用二维纹理和三维纹理, 这样能够有效地加速我们的绘制算法, 以及其他所有需要在通用计算和图形绘制流水线之间共享二维纹理和三维纹理的算法。

6.4.3 八叉树的改进

我们的绘制算法流水线生成的焦散的效果对光子传播的步长非常敏感。而光子传播的步长, 取决于我们的八叉树建树算法中用到的阈值 ϵ 。但是对于某一些体数据, 我们发现难以选择阈值 ϵ 的取值, 以保证不但能够得到良好的绘制结果, 还能有效地限制光子传播的总步长。特别是存在这样的情况, 我们发现如果选择一个较小的阈值以保证足够好的绘制效果, 那么光子传播的步长就非常小, 从而无法达到交互级的绘制性能。如果略微增加阈值, 就会导致八叉树的大量节点合并并在同一个节点里, 从而导致光子传播的步长太大, 而无法得到良好的绘制效果。因此我们修改了原先八叉树建树的方法以设法从八叉树获得合理的光子传播的步长。

我们修改了我们八叉树的存储结构, 在每个体素中除了存储该体素所在叶结点的层数外, 还存储了在这个叶结点的最长的光子传播的步长 $\Delta s_{\max}$ 。我们也修改了前文6.3.4中介绍的自适应光子贴图算法: 我们设置的光子传播的步长 Δs 为 Δs_{octree} , 并且增加约束以保证不小于用户指定的最小传播步长 $\Delta s_{\min}$, 也不大于所在的八叉树叶结点的最长步长 $\Delta s_{\max}$ 。

我们在八叉树的建树步骤中为每个体素指定最长光子传播的步长。当一个体素所在的叶结点所覆盖区域内的折射率的最大差异小于 ϵ , 我们将这个体素的光子传播的最大步长设置为正无穷。如果折射率的最大差异比 ϵ 大, 但是比用户指定的第

二个阈值 ϵ' 小, 那么我们将这个体素的最大步长 $\Delta s_{\max}$ 设置为一个用户指定的限制步长。采用这种策略的目的, 是在折射率基本不变的区域内, 光子传播直接到达所在叶结点的边缘, 而在折射率变化较小的很大的空间区域内, 光子传播以合适的步长进行。在我们的具体实现中, 我们将第一个阈值 ϵ 设置为0.005, 将第二个阈值 ϵ' 设置为0.02, 对应的最长步长为8个体素的宽度。

6.5 实验结果和应用

为了测试我们的绘制方法, 我们将绘制结果和广泛运用的光线跟踪的绘制结果, 以及Ihrke^[43]提出的基于程函方程的绘制结果进行比较。在和光线跟踪绘制得到的参考图比较之后, 我们可以确定相比于基于程函方程的绘制结果, 我们的绘制结果更接近于参考图象。图6-6显示了两组绘制结果的对比。

我们的算法能得到更好的效果, 主要有两个原因。第一, 对每一个体素, 我们收集了所有经过的光子所携带的辐射亮度, 而基于程函方程的绘制方法只记录了辐射亮度最强的传输波阵面片所携带的辐射亮度, 因此, 许多亮度较低的波阵面片从多个方向汇聚到一个体素而产生的焦散效果就无法绘制得到。比如图6.6(c)和图6.6(e)中透明球体折射产生的焦散中心有一个亮点。第二, 我们可以在体数据中任意地跟踪光子传播的路径, 并且产生相应的焦散效果, 而基于程函方程的绘制方法中, 波阵面片的四个顶点如果在不同的时候经过聚焦的奇点, 那么形状怪异的波阵面片就会产生不可预测的错误的焦散效果。这些奇异的波阵面片可以部分被识别并剔除, 但是会丢失许多辐射传输的亮度, 而剩余的未被剔出的奇异波阵面片还会产生类此噪点的效果, 以及锯齿状的焦散边缘。当一个物体的折射率从1逐渐增加, 折射光线到达聚焦的奇点会越来越早, 那么这些奇异波阵面片造成的瑕疵会更加明显。与此相对, 光子贴图则不会因为物体折射率的增加产生类似的绘制瑕疵。

我们的绘制算法在改变光照和材质时, 相比于基于程函方程的绘制方法, 能得到更高的绘制性能。比如绘制如图6-6的场景的重光照效果时, 我们可以得到大约每秒8帧的更新速度, 而基于程函方程的绘制方法大约需要7秒才能更新一帧。我们的算法能达到交互级的绘制性能, 是因为在光子传播时采用了自适应的步长。这样我们能够减少光子传输的总的步数, 也就减少了为每步生成的需要光栅化的线段数量。此外, 我们还采用线段的光栅化代替大量的点来存储场景中的辐射传输。

我们算法的交互级的绘制性能使得许多新颖的应用得以实现。之前的许多折射物体的实时绘制算法只能交互改变绘制的视点, 而现在的我们的绘制算法还能同时改

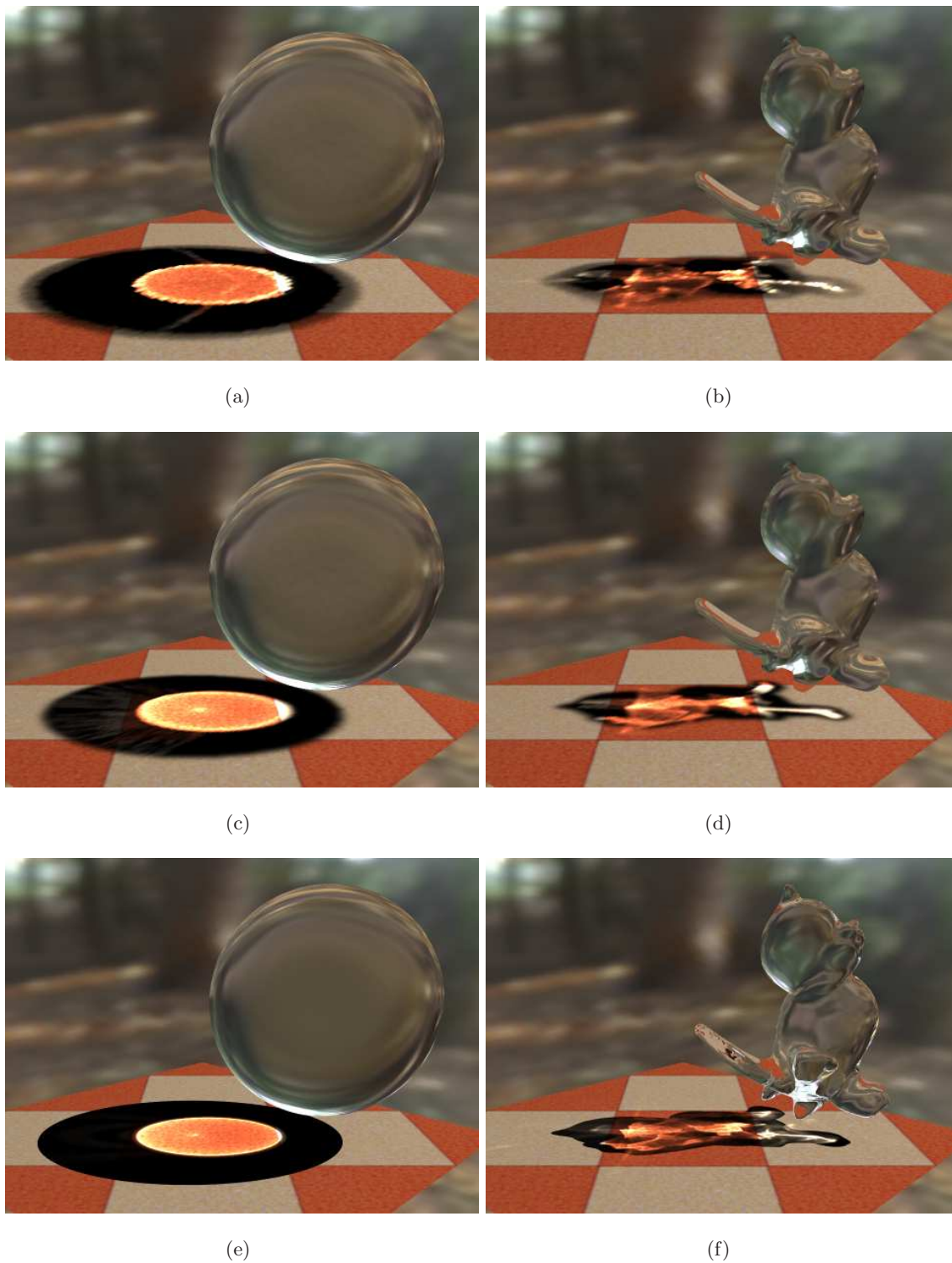


图 6-6 不同算法得到的绘制结果。(a) (b)基于程函方程的绘制方法；(c) (d)我们的绘制方法；(e) (f)光线跟踪得到的参考图像。

变光照和物体的材质，并达到交互级的绘制性能。以下我们介绍了我们算法的一些应用，他们主要的性能统计如表所示。

6.5.1 重光照

我们提出的自适应的非线性光子贴图算法使得我们的绘制算法流水线能得到交互级的绘制性能，即使它包含完整的光子传播和逐像素绘制的过程。我们实现的系统允许用户随意改变光源的位置和颜色，并可以同时看到焦散、消散、单次散射和阴影等效果。图6-7展示了不同光照条件下透明物体的绘制结果，包括参与材质产生的体焦散的绘制效果。

6.5.2 可变材质

我们的绘制算法流水线除了可以改变光照外，还可以改变物体的材质。包含在体数据中的材质一共有三个参数：折射率、消散系数和散射系数。折射率的变化决定了传播光路在场景中如何弯曲。消散系数决定了光在场景中传播时的衰减。散射系数决定了每个体素中存储的辐射亮度对视点的贡献。用户有许多方法来交互改变场景中物体的材质，我们接下来介绍其中的三种。

6.5.2.1 体喷涂(volume painting)

一种编辑体数据中的材质属性的方法是直接对体数据作喷涂操作。我们为用户提供多种“刷子”，每一个都是一种预计算得到的材质属性分布的式样。我们允许

表 6-1 绘制性能。每个实验结果的分辨率为 640×480 ，使用的显卡为 Nvidia GeForce 8800 Ultra，体数据的分辨率为 $128 \times 128 \times 128$ ，生成光子的阴影贴图的分辨率为 1024×1024 。

绘制图像	三角形面片数	生成光子的数量	光子传播总步数 (百万)	绘制性能 (fps)
图6.1(a)	—	185,000	11.5	2.00
图6.1(b)	—	61,000	3.5	3.54
图6.1(c)	5,000	143,000	9.5	2.52
图6-7	—	68,000	3.4	6.14
图6-8	—	41,000	2.0	7.24
图6-9	20,000	35,000	1.7	6.22
图6-10(a)	2,900	75,000	3.5	5.43
图6-10(b)	5,000	70,000	3.2	5.59

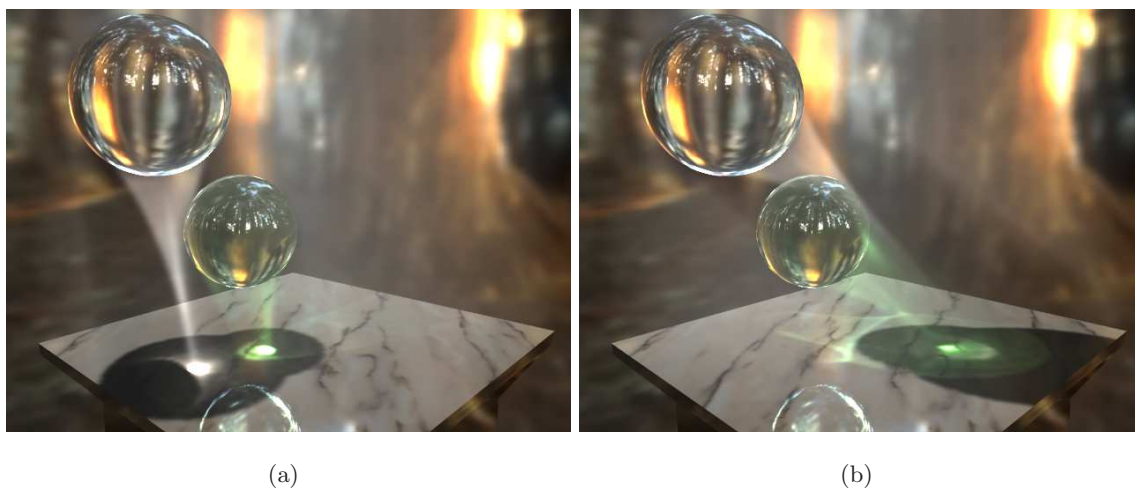


图 6-7 光照改变后产生的面焦散和体焦散的变化绘制结果

用户在整个体数据中喷涂，并对所用的“刷子”进行缩放、旋转和加权等修改。如图6-8中所示的就是一些用户交互喷涂得到的绘制结果。

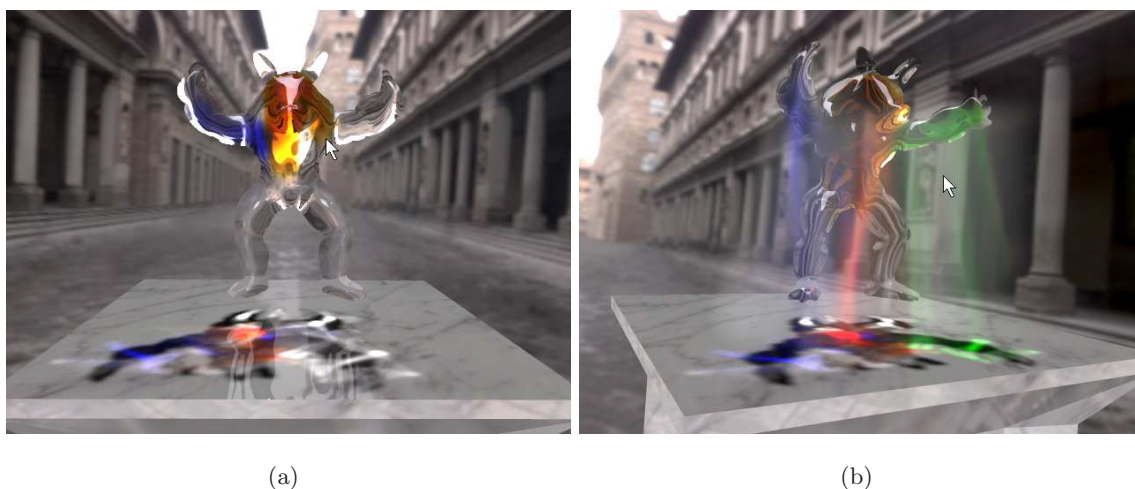


图 6-8 体喷涂的绘制结果：(a)改变物体内部的消散系数；(b)改变物体外部的散射系数。

6.5.2.2 建模和几何变形

虽然体喷涂非常便于我们修改体数据中参与材质的属性，但是非常不便于我们生成一个几何体，或者修改一个物体的几何。我们的绘制算法流水线包含体素化的步骤，能够将一个三角形面片表示的物体几何快速体素化为体数据，因此我们可以在我们的系统中集成许多交互级的几何建模方法，并且同时看到折射的绘制效果。

我们允许用户交互改变一个折射物体的形状，并同时看到这个几何变形后的场景的光照效果。由于包括体素化在内的绘制算法流水线的每个步骤都是在GPU上实现，因此我们在CPU上实现几何变形的运算操作。我们采用了Pauly^[68]的方法实现几何变形，用户通过交互改变用三角形面片表示的物体几何，然后采用我们的体素化方法转化为体数据，再进行八叉树的建树，光子传播以及最后的绘制得到最终的效果图象。如图6-9显示了一些用户交互改变物体几何的绘制效果。

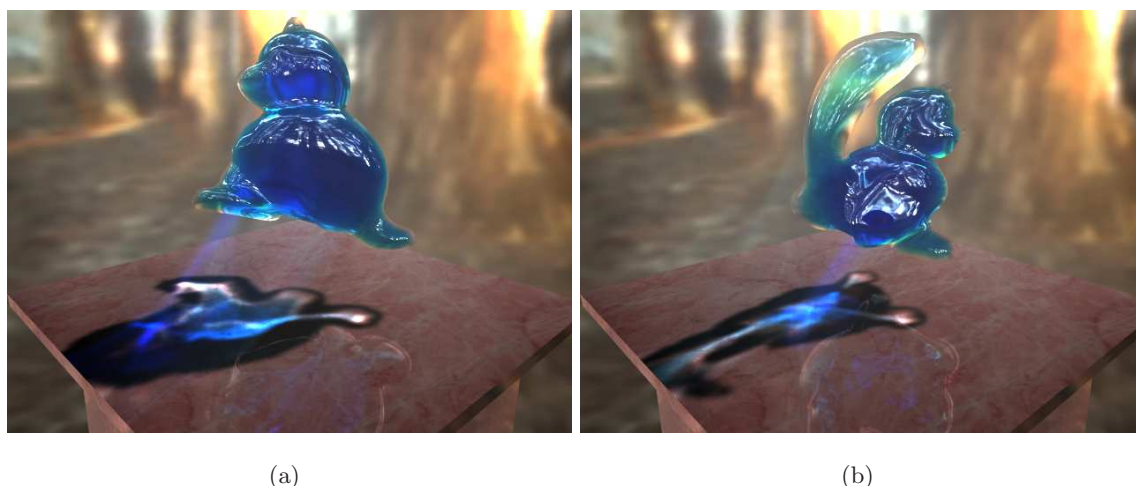


图 6-9 用户交互改变透明物体的几何形状后产生的面焦散和体焦散的变化。

基于手绘草图的建模(sketch-based modeling)近些年广受关注，因为这种建模方法非常直观而且方便。如图6-10所示，我们将Igarashi^[42]的方法集成到我们的系统，并以此进行几何建模，然后经过体素化得到体数据并且完成绘制算法后续步骤地运算操作。事实上，由于我们提供了一个通用的从三角形面片表示的物体几何到体数据最后完整绘制的整个流水线框架，任何现有的交互级建模算法都能作为我们的绘制算法流水线的输入，提供交互生成的物体几何。

6.5.2.3 仿真和动画

物体的材质可以通过对真实物理现象的仿真得到，也可以通过想象编辑的动画序列得到。固体和液体仿真的大量研究作为我们的绘制流水线提供了丰富的输入数据。如图6-1所示的是我们的系统绘制得到的效果，第一个输入数据是Mullen^[60]提供的，第二个数据是根据Fedkiw^[27]的方法计算得到的湍流混合而成，第三个数据是三角形面片表示的物体的动画序列。在这些实验中，我们的系统能为这些动态改变的物体材质绘制任意光照方向和视点方向下的全局光照效果。



图 6-10 使用基于手绘草图的建模方法得到的两个物体的绘制结果

6.6 结论

我们提出的绘制方法能够得到折射、单次散射和消散的绘制效果，并且在改变光照、视点和材质时能达到交互级的绘制性能。相比于之前的方法，我们有了显著的进步，因为以往的方法无法在改变光照和材质时达到交互级的绘制性能。我们的算法在GPU上实现，首先将三角形面片表示的物体几何转化为体数据，然后采用自适应的非线性光子贴图算法跟踪光子的传播，最后绘制效果图像。我们的方法没有使用CPU进行大量的运算，因此可以使用CPU实现交互级的建模或物理仿真算法得到输入数据。

尽管我们的方法和传统的体数据的光子贴图算法相类似，但是我们采用均匀分布的体素网格存储辐射亮度的分布，而不是采用稀疏Kd-tree的数据结构。可能有人认为这种体数据的表示方法难以扩展到更大规模的场景上，但是我们认为对于许多包含一个折射物体或者有限的空间区域内存在参与材质的场景而言，这样的个体数据的表示已经足够了。此外，这种体数据表示的方法能充分利用GPU的并行性在光子传播中在体数据中存储辐射传输，从而避免了在逐像素绘制中传统光子贴图算法必需的收集临近光子的运算操作的高昂代价。

我们的方法和基于程函方程的绘制方法也有相似的地方，但是我们在绘制性能上获得了极大地提高，这主要是因为我们采用了自适应的光子传播的步长。我们能够随意的改变光子传播的步长是由于我们跟踪的是单个光子的传播，而不是一个连续的波阵面。此外，基于程函方程的绘制方法在跟踪波正面传播时会由于聚焦奇点

而带来绘制瑕疵，而我们的方法能彻底避免这样的问题。

我们的绘制算法流水线需要同时采用OpenGL和CUDA来实现，因此我们希望这种通用计算和图形绘制混合的应用能得到更有力的支持。具体而言，我们希望硬件绘制流水线能够在三维纹理上进行光栅化操作，CUDA能够支持读写的原子操作，OpenGL和CUDA能够共享二维纹理和三维纹理。

第7章 总结与展望

真实感图形绘制的目标是在计算机的虚拟场景中重现真实世界中的各种光照效果。全局光照明算法相对于局部光照明算法，由于考虑了场景中物体对辐射传输的各种影响效果，因此更为符合真实世界的物理原理，能够产生更为逼真的绘制图像，因此在电影制作、室内设计等应用中，都需要采用全局光照明的算法生成效果图像。由于真实世界的场景中物体的作用非常复杂，因此全局光照明在模拟辐射传输时的时间和空间复杂度都非常高，用户对场景作任何改变后，都需要等待较长的时间才能得到修改结果的绘制效果。而在游戏、设计等诸多领域，用户需要对场景进行频繁的交互改动，同时需要立刻得到反馈，也就是说，用户在改变场景的同时就要得到相应的全局光照明的绘制。因此，实时或交互级的全局观光照明的绘制算法有广泛的应用需求。用户对一个绘制场景的修改主要包括视点、光照、材质和几何，而这些也就唯一决定了绘制的结果。该领域的研究人员首先致力于解决动态光照和视点的实时全局光照明的绘制，比如PRT的绘制框架，利用了光照和绘制效果的线性关系实现了静态场景、静态材质、动态视点、动态光照下的实时重光照。但是物体的材质和绘制效果不存在这种线性关系，因此类似于PRT的实时全局光照明绘制算法不能直接扩展为可变材质的实时全局光照明绘制算法。本文的工作致力于研究可变材质的实时全局光照明的绘制算法。换言之，本文提出的全局光照明的绘制算法能在用户改变绘制场景中物体材质的同时提供相应的全局光照明的绘制效果。我们的方法在电子游戏、计算机辅助设计等领域有广泛的应用前景。

7.1 全文工作的总结

本文的工作主要包含三部分：第4、5章的绘制场景中的物体采用反射材质，第6章的绘制场景中的物体为透明、散射材质；第4章的绘制算法必须固定光照和视点，而第5、6章的绘制算法没有这个约束。

第4章中，我们实现了静态场景、静态复杂光照、静态视点、可变反射材质的实时全局光照明绘制。我们的工作的意义，在于首先克服了辐射传输和材质的非线性关系。之前绝大多数实时全局光照明绘制算法，只能支持动态光源，原因是光照和辐射传输成线性关系。但是辐射传输和材质之间不存在这样的线性关系，所有我们的思路是如何将辐射传输表示为材质的函数，并且适于预计算和绘制，已达到足

够高的绘制性能。辐射传输和物体材质的非线性关系的原因，是光从光源出射后到达视点前可能经过不同的反射路径，也就经过不同物体不同次数的反射。因此我们的解决方案是辐射传输分离，也就是说我们将到达视点的辐射传输根据反射的路径进行分类。这样，每一条特定的反射路径的辐射传输就与其所经过的所有反射材质的乘积成正比。由此，我们将可变材质的这个非线性问题转化为一组线性问题的叠加。同时，我们用一组线性基表示BRDF空间。在预计算时，我们将这组线性基的所有可能的组合依次赋予场景中的物体，绘制计算相应的所有反射路径的辐射传输。在绘制时，我们允许用户修改物体的材质，然后将所有的材质投影的BRDF的基函数上，我们用材质的投影系数组合所有预计算的结果，就能实时得到相应的全局光照的绘制结果。对于空间静态可变材质，我们没有压缩预计算的数据，就可以得到实时的绘制性能。对于空间动态可变材质，我们用大量的空间静态可变材质小区域近似模拟一个空间动态可变材质物体，并且采用合理有效的方法压缩预计算得到的数据，从而能够达到交互级的绘制性能。

第5章中，我们实现了静态场景、动态环境光照、动态视点、可变反射材质的交互级全局光照绘制，并且在光照不变时达到实时的绘制性能。第4章的算法，最大的局限性在于无法支持动态光照和动态视点，这对于可变材质的实时全局光照绘制算法的应用前景有严重的影响。这是因为在许多虚拟场景设计的具体应用中，用户往往需要的是对光照、材质和视点同时进行修改编辑。因此第5章的工作，就具有非常重要的意义和广阔的应用前景。为了能够让我们的绘制框架支持动态光照、动态视点和可变材质，我们不但要克服PRT框架中辐射传输和物体材质互相耦合无法动态更新的弱点，还要避免像第4章中那样将所有的存储和运算资源都消耗在可变材质的数据空间上的情况。为此，我们采用了三种策略：光路切分，预计算传输张量，针对BRDF的镜面项分离和张量分解。光路切分指的是将辐射传输的路径分为两部分：局部入射辐射亮度和顶点着色。通过光路切分，辐射传输的数据空间就被有效地减小了。通过PTT的表示，局部入射辐射亮度就成为光照和材质的函数，其预计算结果能够在绘制时组合成不同光照和材质条件下的辐射传输。针对顶点着色部分，我们对BRDF空间采用镜面项分离和张量分解的方法，实现了动态的局部入射辐射亮度、动态的物体表面材质以及动态的视点方向的快速着色。

第6章中，我们实现了动态几何、动态光照、动态视点、可变的非均质的透明和散射材质的交互级全局光照绘制。除了第4章和第5章中的反射材质，透明和散射材质也是两种非常重要和常见的材质类型。和反射材质相比，透明和散射材质的

绘制有很大的不同。透明材质的绘制效果十分高频，细节丰富，而且在折射点不需要对半球面入射辐射亮度进行积分。散射材质的绘制的计算复杂度很高，因为光在传播过程中在所有位置都可能发生变化，而不只局限于物体表面。而非均质的透明和散射材质，相比于均质的材质，绘制复杂度更高，当然光照效果也更丰富。我们采用体数据表示一个绘制场景，并且提出了一个完整的绘制算法流水线：首先将三角形面片表示的物体几何体素化为体数据，其次为表示绘制场景的体数据建立一个八叉树的加速数据结构，然后在体数据的表面及物体表面产生光子，并且利用八叉树数据实现自适应传播步长的光子传播，最后逐像素绘制效果图像，能得到阴影、吸收、反射、折射、单次散射、面焦散、体焦散等多种光照效果。流水线的所有步骤都在GPU上实现，能够达到交互级的绘制性能。用户可以通过交互改变绘制场景中的物体几何、光照、材质和视点，而且不需要对此作任何预计算，只需要修改绘制流水线中相应步骤的输入即可。我们的算法，为用户提供了多种类型的交互手段，比如体喷涂、几何变形以及其它的交互级建模方法。

7.2 未来工作的展望

可变材质的实时和交互级全局光照明绘制算法的研究的可以在以下几个方面继续展开：

数据量的有效压缩

可变材质的实时和交互级全局光照的绘制的最大困难在于所需要采集的数据空间非常巨大。在我们第4、5章的内容中，我们有许多算法和工作都是围绕如何有效合理地采集和压缩数据，如果便于利用大数据量进行绘制。这部分的工作并没有达到极致，我们只是尝试了常用的压缩算法，选择了最为有效的策略，根据我们的特殊用途略作修正，并没有根据辐射传输，尤其是可变材质的辐射传输的特点施加先验知识、启发式规则或者设计特殊的压缩算法。而全局光照明问题中辐射传输的稀疏性和相关性是众所周知的，如何根据可变材质的辐射传输度身而制数据的表示形式和压缩算法，有效地提高压缩率和绘制效果，成为该类算法进一步提高有效性和实用性的关键所在。

统一的重光照算法框架

从提高算法实用性的角度而言，提供给用户的系统框架应该尽可能是一个统一的重光照算法框架，也就是说，用户可以自由的改变几何、光照、材质、视点等多

个因素。因此，虽然说文中第5章中基于顶点的绘制算法没有第4章中基于像素的绘制算法得到的结果精细，但是却更为实用，应用前景也更广。而第6章中的绘制算法还允许改变物体几何，因此实用性上更进一步，但是针对的是另一种材质的绘制场景，无法同时处理反射、折射、散射的可变材质。虽然各种算法各有所长，但是由于计算资源有限，难以在现有的硬件条件下集成多种算法，又或者算法之间本身互斥性很强，在算法流程和数据结构上难以统一，因此后续的工作可以多考虑算法的可集成性，从而更便于统一的重光照算法框架的设计和实现。

参考文献

- [1] 孙鑫, 周昆, 石教英. 空间动态可变材质的交互式全局光照照明绘制. *软件学报*, 19(7):1783–1793, 7 2007.
- [2] 孙鑫, 周昆, 石教英. 可变材质的实时全局光照照明绘制. *软件学报*, 19(4):1004–1015, 4 2008.
- [3] A. APPEL. Some techniques for shading machine renderings of solids. In *Proc. of Spring Joint Computer Conference*, pages 37–45, 1968.
- [4] Larry Aupperle and Pat Hanrahan. A hierarchical illumination algorithm for surfaces with glossy reflection. *Computer Graphics*, 27(Annual Conference Series):155–162, 1993.
- [5] A. Ben-Artzi, K. Egan, F Durand, and R. Ramamoorthi. A precomputed polynomial representation for interactive brdf editing with global illumination. *ACM Trans. Graph.*, 27(2):1–13, 2008.
- [6] Aner Ben-Artzi, Ryan Overbeck, and Ravi Ramamoorthi. Real-time brdf editing in complex lighting. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Papers*, pages 945–954, New York, NY, USA, 2006. ACM.
- [7] J. F. Blinn. Models of light reflection for computer synthesized pictures. In *Proc. of SIGGRAPH '77*, pages 192–198, 1977.
- [8] Max Born and Emil Wolf. *Principles of Optics (7th edition)*. Cambridge University Press, 1999.
- [9] Chris Buchalew and Donald Fussell. Illumination networks: fast realistic rendering with general reflectance functions. *SIGGRAPH Comput. Graph.*, 23(3):89–98, 1989.
- [10] Chris Wyman Charles. Interactive raytraced caustics.
- [11] Siddhartha Chatterjee, Guy E. Blelloch, and Marco Zagha. Scan Primitives for Vector Computers. In *Supercomputing 1990*, pages 666–675, 1990.
- [12] Petrik Clarberg, Wojciech Jarosz, Tomas Akenine-Möller, and Henrik Wann Jensen. Wavelet importance sampling: efficiently evaluating products of complex functions. volume 24, pages 1166–1175, New York, NY, USA, 2005. ACM.
- [13] M. F. Cohen and D. P. Greenberg. The hemi-cube: a radiosity solution for complex environment. In *Proc. of SIGGRAPH '85*, pages 31–40, 1985.

- [14] Michael F. Cohen, Shenchang Eric Chen, John R. Wallace, and Donald P. Greenberg. A progressive refinement approach to fast radiosity image generation. *SIGGRAPH Comput. Graph.*, 22(4):75–84, 1988.
- [15] R. L. Cook, T. Porter, and L. Carpenter. Distributed ray tracing. In *Proc. of SIGGRAPH '84*, pages 137–145, 1984.
- [16] R. L. Cook and K. E. Torrance. A reflectance model for computer graphics. In *SIGGRAPH '81*, pages 307–316, 1981.
- [17] Keenan Crane, Ignacio Llamas, and Sarah Tariq. Real-Time Simulation and Rendering of 3D Fluids. In Hubert Nguyen, editor, *GPU Gems 3*, chapter 30, pages 633–675. Addison-Wesley Professional, 2007.
- [18] Scott T Davis and Chris Wyman. Interactive refractions with total internal reflection. In *GI '07: Proceedings of Graphics Interface 2007*, pages 185–190, New York, NY, USA, 2007. ACM.
- [19] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.
- [20] Y. Dobashi, K. Kaneda, H. Nakatani, and H. Yamashita. A quick rendering method using basis functions for interactive lighting design. *Computer Graphics Forum*, 14(3):229–240, 1995.
- [21] J. Dorsey, J. Arvo, and Greenberg D. Interactive design of complex time dependent lighting. In *Proc. of SIGGRAPH '95*, pages 26–36, 1995.
- [22] J. Dorsey, F. Sillion, and Greenberg D. P. Design and simulation of opera lighting and projection effects. In *Proc. of SIGGRAPH '91*, pages 41–50, 1991.
- [23] G. Drettakis and F. X. Sillion. Interactive update of global illumination using a line-space hierarchy. In *Proc. of SIGGRAPH '97*, pages 57–64, 1997.
- [24] Frédo Durand, Nicolas Holzschuch, Cyril Soler, Eric Chan, and François X. Sillion. A frequency analysis of light transport. In *SIGGRAPH '05: ACM SIGGRAPH 2005 Papers*, pages 1115–1126, New York, NY, USA, 2005. ACM.
- [25] Manfred Ernst, Tomas Akenine-Möller, and Henrik Wann Jensen. Interactive rendering of caustics using interpolated warped volumes. In *GI '05: Proceedings of Graphics Interface 2005*, pages 87–96, School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, 2005. Canadian Human-Computer Communications Society.

- [26] S. Ershov, K. Kolchin, and K. Myszkowski. Rendering pearlescent appearance based on paint-composition modelling. In *Proc. of Eurographics '01*, pages 227–238, 2001.
- [27] Ronald Fedkiw, Jos Stam, and Henrik Wann Jensen. Visual simulation of smoke. In Eugene Fiume, editor, *SIGGRAPH 2001, Computer Graphics Proceedings*, pages 15–22. ACM Press / ACM SIGGRAPH, 2001.
- [28] R. Gershbein and P. M. Hanrahan. A fast relighting engine for interactive cinematic lighting design. In *Proc. of SIGGRAPH '00*, pages 353–358, 2000.
- [29] C. M. Goral, K. E. Torrance, Greenberg D. P., and B. Battaile. Modeling the interaction of light between diffuse surfaces. In *Proc. of SIGGRAPH '84*, pages 213–222, 1984.
- [30] S. J. Gortler, P. Schröder, M. F. Cohen, and P. Hanrahan. Wavelet radiosity. In *Proc. of SIGGRAPH-93: Computer Graphics*, pages 221–230, Anaheim, CA, 1993.
- [31] Paul Green, Jan Kautz, Wojciech Matusik, and Frédo Durand. View-dependent precomputed light transport using nonlinear gaussian function approximations. In *I3D '06: Proceedings of the 2006 symposium on Interactive 3D graphics and games*, pages 7–14, New York, NY, USA, 2006. ACM.
- [32] Eduard Gröller. Nonlinear ray tracing: visualizing strange worlds. *The Visual Computer*, 11(5):263–274, 1995.
- [33] Jinwei Gu, Chien-I Tu, Ravi Ramamoorthi, Peter Belhumeur, Wojciech Matusik, and Shree Nayar. Time-varying surface appearance: acquisition, modeling and rendering. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Papers*, pages 762–771, New York, NY, USA, 2006. ACM.
- [34] Diego Gutierrez, Adolfo Munoz, Oscar Anson, and Francisco J. Seron. Non-linear volume photon mapping. In Oliver Deussen, Alexander Keller, Kavita Bala, Philip Dutré, Dieter W. Fellner, and Stephen N. Spencer, editors, *Rendering Techniques*, pages 291–300. Eurographics Association, 2005.
- [35] Stephane Guy and Cyril Soler. Graphics gems revisited: fast and physically-based rendering of gemstones. *ACM Trans. Graph.*, 23(3):231–238, 2004.
- [36] Ziyad S. Hakura and John M. Snyder. Realistic reflections and refractions on graphics hardware with hybrid rendering and layered environment maps. In *Proceedings of the 12th Eurographics Workshop on Rendering Techniques*, pages 289–300, London, UK, 2001. Springer-Verlag.
- [37] Pat Hanrahan, David Salzman, and Larry Aupperle. A rapid hierarchical radiosity algorithm. *SIGGRAPH Comput. Graph.*, 25(4):197–206, 1991.

- [38] Miloš Hašan, Fabio Pellacini, and Kavita Bala. Direct-to-indirect transfer for cinematic re-lighting. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Papers*, pages 1089–1097, New York, NY, USA, 2006. ACM.
- [39] Miloš Hašan, Fabio Pellacini, and Kavita Bala. Matrix row-column sampling for the many-light problem. *ACM Trans. Graph.*, 26(3):26, 2007.
- [40] Xianyou Hou, Li-Yi Wei, Heung-Yeung Shum, and Baining Guo. Real-time multi-perspective rendering on graphics hardware. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Sketches*, page 79, New York, NY, USA, 2006. ACM.
- [41] Wei Hu and Kaihuai Qin. Interactive approximate rendering of reflections, refractions, and caustics. *IEEE Transactions on Visualization and Computer Graphics*, 13(1):46–57, 2007.
- [42] Takeo Igarashi, Satoshi Matsuoka, and Hidehiko Tanaka. Teddy: a sketching interface for 3d freeform design. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Courses*, page 11, New York, NY, USA, 2006. ACM.
- [43] Ivo Ihrke, Gernot Ziegler, Art Tevs, Christian Theobalt, Marcus Magnor, and Hans-Peter Seidel. Eikonal rendering: efficient light transport in refractive objects. *ACM Trans. Graph.*, 26(3):59, 2007.
- [44] D. S. Immel, M. F. Cohen, and D. P. Greenberg. A radiosity method for nondiffuse environments. In *Proc. of SIGGRAPH '86*, pages 133–142, 1986.
- [45] H. K. Jensen. *Realistic image synthesis using photon mapping*. A. K. Peters, Ltd., Natick, MA, USA, 2001.
- [46] Henrik Wann Jensen and Per H. Christensen. Efficient simulation of light transport in scenes with participating media using photon maps. In *Proc. of SIGGRAPH '98*, pages 311–320, 1998.
- [47] James T. Kajiya. The rendering equation. In *Proc. of SIGGRAPH '86*, pages 143–150, 1986.
- [48] Jan Kautz and Michael D. McCool. Interactive rendering with arbitrary brdfs using separable approximations. In *SIGGRAPH '99: ACM SIGGRAPH 99 Conference abstracts and applications*, page 253, New York, NY, USA, 1999. ACM.
- [49] Anders Wang Kristensen, Tomas Akenine-Möller, and Henrik Wann Jensen. Precomputed local radiance transfer for real-time lighting design. In *SIGGRAPH '05: ACM SIGGRAPH 2005 Papers*, pages 1208–1215, New York, NY, USA, 2005. ACM.

- [50] Jens Krüger, Kai Bürger, and Rüdiger Westermann. Interactive screen-space accurate photon tracing on GPUs. In *Rendering Techniques (Eurographics Symposium on Rendering - EGSR)*, pages 319–329, June 2006.
- [51] Lieven De Lathauwer, Bart De Moor, and Joos Vandewalle. On the best rank-1 and rank- $(r_1, r_2, \dots, r_n)$ approximation of higher-order tensors. *SIAM J. Matrix Anal. Appl.*, 21(4):1324–1342, 2000.
- [52] Jason Lawrence, Aner Ben-Artzi, Christopher DeCoro, Wojciech Matusik, Hanspeter Pfister, Ravi Ramamoorthi, and Szymon Rusinkiewicz. Inverse shade trees for non-parametric material representation and editing. *ACM Trans. Graph.*, 25(3):735–745, 2006.
- [53] M. E. Lee and R. A. Redner. Filtering: A note on the use of nonlinear filtering in computer graphics. *IEEE Computer Graphics and Applications*, 10(3):23–29, 1990.
- [54] X. Liu, P. Sloan, H.-Y. Shum, and J. Snyder. All-frequency precomputed radiance transfer for glossy objects. In *Proc. of Eurographics Symposium on Rendering '04*, pages 337–344, 2004.
- [55] W. Matusik, J. Pfister, M. Brand, and L. McMillan. A data-driven reflectance model. In *Proc. of SIGGRAPH '03*, pages 759–769, 2003.
- [56] M. D. McCool, J. Ang, and A. Ahmad. Homomorphic factorizations of brdfs for high-performance rendering. In *Proc. of SIGGRAPH '01*, pages 227–238, 2001.
- [57] C. H. Mei, J. Y. Shi, and F. L. Wu. Rendering with spherical radiance transport maps. In *Proc. of Eurographics '01*, pages 281–290, 2004.
- [58] Mark Meyer and John Anderson. Statistical acceleration for animated global illumination. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Papers*, pages 1075–1080, New York, NY, USA, 2006. ACM.
- [59] M. Minnaert. The reciprocity principle in lunar photometry. *Astrophysical Journal*, 93:403–410, May 1941.
- [60] Patrick Mullen, Alexander McKenzie, Yiyang Tong, and Mathieu Desbrun. A variational approach to eulerian geometry processing. *ACM Trans. Graph.*, 26(3):66, 2007.
- [61] R. Ng, R. Ramamoorthi, and P. Hanrahan. All-frequency shadows using non-linear wavelet lighting approximation. In *Proc. of SIGGRAPH '03*, pages 376–381, 2003.
- [62] R. Ng, R. Ramamoorthi, and P. Hanrahan. Triple product integrals for all-frequency relighting. In *Proc. of SIGGRAPH '04*, pages 477–487, 2004.

- [63] A. Ngan, F. Durand, and W. Matusik. Experimental analysis of brdf models. In *Proceedings of the Eurographics Symposium on Rendering*, pages 117–226, 2005.
- [64] A. Ngan, F. Durand, and W. Matusik. Image-driven navigation of analytical brdf models. In *Proceedings of the Eurographics Symposium on Rendering*, pages 399–408, 2006.
- [65] T. Nishita and E. Nakamae. Continuous tone representation of three-dimensional objects taking account of shadows and interreflection. In *Proc. of SIGGRAPH '85*, pages 23–30, 1985.
- [66] Nvidia Corporation. CUDA Programming Guide. <http://developer.nvidia.com/object/cuda.html>, 2007.
- [67] Manuel M. Oliveira and Maicon Brauwere. Real-time refraction through deformable objects. In *I3D '07: Proceedings of the 2007 symposium on Interactive 3D graphics and games*, pages 89–96, New York, NY, USA, 2007. ACM.
- [68] Mark Pauly, Richard Keiser, Leif P. Kobbelt, and Markus Gross. Shape modeling with point-sampled geometry. *ACM Trans. Graph.*, 22(3):641–650, 2003.
- [69] F. Pellacini, K. Vidimce, A. Lefohn, A. Mohr, M. Leone, and J. Warren. Lpics: a hybrid hardware-accelerated relighting engine for computer cinematography. In *Proc. of SIGGRAPH '05*, pages 464–470, 2005.
- [70] Fabio Pellacini and Jason Lawrence. Appwand: editing measured materials using appearance-driven optimization. *ACM Trans. Graph.*, 26(3):54, 2007.
- [71] B. T. Phong. Illumination for computer generated pictures. *Commun. ACM*, 18(6):311–317, 1975.
- [72] Voicu Popescu, Chunhui Mei, Jordan Dauble, and Elisha Sacks. Reflected-scene impostors for realistic reflections at interactive rates. *Computer Graphics Forum*, 25(3):313–322, September 2006.
- [73] Timothy J. Purcell, Craig Donner, Mike Cammarano, Henrik Wann Jensen, and Pat Hanrahan. Photon mapping on programmable graphics hardware. In *SIGGRAPH '05: ACM SIGGRAPH 2005 Courses*, page 258, New York, NY, USA, 2005. ACM.
- [74] Z. Ren, R. Wang, J. Snyder, K. Zhou, X. Liu, B. Sun, P. P. Sloan, H. J. Bao, Q. S. Peng, and B. N. Guo. Real-time soft shadows in dynamic scenes using spherical harmonic exponentiation. *Proc. of SIGGRAPH '06*, pages 977–986, 2006.

- [75] David Roger and Nicolas Holzschuch. Accurate specular reflections in real-time. *Computer Graphics Forum (Proceedings of Eurographics 2006)*, 25(3), sep 2006.
- [76] T. Saito and T Takahashi. Comprehensible rendering of 3-d shapes. In *Proc. of SIGGRAPH '90*, pages 197–206, 1990.
- [77] J. Semancik. Ray tracing non-linear spacewarps, 2002.
- [78] Francisco J. Serón, Diego Gutierrez, Guillermo Gutierrez, and Eva Cerezo. Implementation of a method of curved ray tracing for inhomogeneous atmospheres. *Computers & Graphics*, 29(1):95–108, 2005.
- [79] M. Z. Shao, Q. S. Peng, and Liang Y. D. A new radiosity approach by procedural refinement for realistic image synthesis. In *Proc. of SIGGRAPH '88*, pages 93–101, 1988.
- [80] Peter Shirley and R. Keith Morley. *Realistic Ray Tracing*. A. K. Peters, Ltd., Natick, MA, USA, 2003.
- [81] F. Sillion and C. Puech. A general two-pass method integrating specular and diffuse reflection. In *SIGGRAPH '89: Proceedings of the 16th annual conference on Computer graphics and interactive techniques*, pages 335–344, New York, NY, USA, 1989. ACM.
- [82] François Sillion, James Arvo, Stephen Westin, and Donald P. Greenberg. A global illumination solution for general reflectance distributions. In *SIGGRAPH'91 conference proceedings*, pages 187–196. ACM Press, 1991.
- [83] P. Sloan, J. Hall, J. Hart, and J. Snyder. Clustered principal components for precomputed radiance transfer. In *Proc. of SIGGRAPH '03*, pages 382–391, 2003.
- [84] P. Sloan, J. Kautz, and J. Snyder. Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. In *Proc. of SIGGRAPH '02*, pages 527–536, 2002.
- [85] Jos Stam and Eric Languénou. Ray tracing in non-constant media. In *Proceedings of the euro-graphics workshop on Rendering techniques '96*, pages 225–234, London, UK, 1996. Springer-Verlag.
- [86] Weifeng Sun and Amar Mukherjee. Generalized wavelet product integral for rendering dynamic glossy objects. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Papers*, pages 955–966, New York, NY, USA, 2006. ACM.
- [87] Xin Sun, Kun Zhou, Yanyun Chen, Stephen Lin, Jiaoying Shi, and Baining Guo. Interactive relighting with dynamic brdfs. *ACM Trans. Graph.*, 26(3):27, 2007.

- [88] Eric Tabellion and Arnauld Lamorlette. An approximate global illumination system for computer generated films. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Papers*, pages 469–476, New York, NY, USA, 2004. ACM.
- [89] Roy Troutman and Nelson L. Max. Radiosity algorithms using higher order finite element methods. In *SIGGRAPH '93: Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pages 209–212, New York, NY, USA, 1993. ACM.
- [90] Yu-Ting Tsai and Zen-Chung Shih. All-frequency precomputed radiance transfer using spherical radial basis functions and clustered tensor approximation. *ACM Trans. Graph.*, 25(3):967–976, 2006.
- [91] M. A. O. Vasilescu and D. Terzopoulos. Tensortextures: multilinear image-based rendering. In *Proc. of SIGGRAPH '04*, pages 336–342, 2004.
- [92] J. R. Wallace, K. A. Elmquist, and E. A. Haines. A ray tracing algorithm for progressive radiosity. *SIGGRAPH Comput. Graph.*, 23(3):315–324, 1989.
- [93] John R. Wallace, Michael F. Cohen, and Donald P. Greenberg. A two-pass solution to the rendering equation: A synthesis of ray tracing and radiosity methods. *SIGGRAPH Comput. Graph.*, 21(4):311–320, 1987.
- [94] Bruce Walter, Sebastian Fernandez, Adam Arbree, Kavita Bala, Michael Donikian, and Donald P. Greenberg. Lightcuts: a scalable approach to illumination. In *SIGGRAPH '05: ACM SIGGRAPH 2005 Papers*, pages 1098–1107, New York, NY, USA, 2005. ACM.
- [95] Michael Wand and Wolfgang Straßer. Real-time caustics. *Comput. Graph. Forum*, 22(3):611–620, 2003.
- [96] H. C. Wang, Q. Wu, L. Shi, Y. Z. Yu, and N. Ahuja. Out-of-core tensor approximation of multi-dimensional matrices of visual data. In *Proc. of SIGGRAPH '05*, pages 527–535, 2005.
- [97] R. Wang, J. Tran, and D. Luebke. All-frequency relighting of non-diffuse objects using separable BRDF approximation. In *Proc. of Eurographics Symposium on Rendering '04*, pages 345–354, 2004.
- [98] G. J. Ward. Measuring and modeling anisotropic reflection. In *Proc. of SIGGRAPH '92*, pages 265–272, 1992.
- [99] Li-Yi Wei, Baoquan Liu, Xu Yang, Ying-Qing Xu, Baining Guo, and Chongyang Ma. Nonlinear Beam Tracing on a GPU.

- [100] D. Weiskopf, T. Schafhitzel, and T. Ertl. GPU-Based Nonlinear Ray Tracing. *Computer Graphics Forum (Eurographics 2004)*, 23(3):625–633, 2004.
- [101] Daniel Weiskopf. Four-dimensional non-linear ray tracing as a visualization tool for gravitational physics. In *VISUALIZATION '00: Proceedings of the 11th IEEE Visualization 2000 Conference (VIS 2000)*, Washington, DC, USA, 2000. IEEE Computer Society.
- [102] Turner Whitted. An improved illumination model for shaded display. *Commun. ACM*, 23(6):343–349, 1980.
- [103] Chris Wyman. An approximate image-space approach for interactive refraction. *ACM Trans. Graph.*, 24(3):1050–1053, 2005.
- [104] Chris Wyman. Interactive image-space refraction of nearby geometry. In *GRAPHITE '05: Proceedings of the 3rd international conference on Computer graphics and interactive techniques in Australasia and South East Asia*, pages 205–211, New York, NY, USA, 2005. ACM.
- [105] Chris Wyman and Scott Davis. Interactive image-space techniques for approximating caustics. In *I3D '06: Proceedings of the 2006 symposium on Interactive 3D graphics and games*, pages 153–160, New York, NY, USA, 2006. ACM.
- [106] Chris Wyman, Charles Hansen, and Peter Shirley. Interactive caustics using local precomputed irradiance. In *PG '04: Proceedings of the Computer Graphics and Applications, 12th Pacific Conference on (PG'04)*, pages 143–151, Washington, DC, USA, 2004. IEEE Computer Society.
- [107] H. R. Zatz. Galerkin radiosity: A higher order solution method for global illumination. In *Proc. of SIGGRAPH-93: Computer Graphics*, pages 213–220, Anaheim, CA, 1993.
- [108] K. Zhou, Y. Hu, S. Lin, B. Guo, and H. Shum. Precomputed shadow fields for dynamic scenes. In *Proc. of SIGGRAPH '05*, pages 1196–1201, 2005.
- [109] Gernot Ziegler, Rouslan Dimitrov, Christian Theobalt, and Hans-Peter Seidel. Real-Time Quadtree Analysis Using HistoPyramids. In *Real-Time Image Processing 2007*, page 64960L, 2007.

攻读博士学位期间主要的研究成果

[1] Xin Sun, Kun Zhou, Eric Stollnitz, Jiaoying Shi, Baining Guo. Interactive Relighting of Dynamic Refractive Objects. ACM SIGGRAPH '08, 2008. SCI, EI 收录

[2] Xin Sun, Kun Zhou, Yanyun Chen, Stephen Lin, Jiaoying Shi, Baining Guo. Interactive relighting with dynamic BRDFs. ACM SIGGRAPH '07, 2007. SCI, EI 收录

[3] 孙鑫, 周昆, 石教英. 空间动态可变材质的交互式全局光照明绘制, 软件学报(EI期刊), 第19卷, 第7期, P.1783–1793, 2008年7月

[4] 孙鑫, 周昆, 石教英. 可变材质的实时全局光照明绘制, 软件学报(EI期刊), 第19卷, 第4期, P.1004–1015, 2008年4月