

中图分类号：TP391.41

论文编号：10006BY1206148

北京航空航天大学
博士学位论文

复杂场景近似表达
及绘制方法研究

作者姓名 谢乃闻

学科专业 计算机应用技术

指导教师 赵沁平 教授

王莉莉 教授

培养学院 计算机学院

Researches on Approximate Expressions for Complex Scenes and Rendering

A Dissertation Submitted for the Degree of Doctor of Philosophy

Candidate: Xie Naiwen

Supervisor: Prof. Zhao Qinpeng

Prof. Wang Lili

School of Computer Science & Engineering

Beihang University, Beijing, China

中图分类号：TP391.41

论文编号：10006BY1206148

博 士 学 位 论 文

复杂场景近似表达及绘制方法研究

作者姓名	谢乃闻	申请学位级别	博士
指导教师姓名	赵沁平	职 称	教授
	王莉莉		教授
学科专业	计算机应用技术	研究方向	三维图形绘制
学习时间自	2012 年 月 日	起至	年 月 日止
论文提交日期	年 月 日	论文答辩日期	年 月 日
学位授予单位	北京航空航天大学	学位授予日期	年 月 日

关于学位论文的独创性声明

本人郑重声明：所呈交的论文是本人在指导教师指导下独立进行研究工作所取得的成果，论文中有关资料和数据是实事求是的。尽我所知，除文中已经加以标注和致谢外，本论文不包含其他人已经发表或撰写过的研究成果，也不包含本人或他人为获得北京航空航天大学或其它教育机构的学位或学历证书而使用过的材料。与我一同工作的同志对研究所做的任何贡献均已在论文中作出了明确的说明。

若有不实之处，本人愿意承担相关法律责任。

学位论文作者签名：_____ 日期：_____ 年 _____ 月 _____ 日

学位论文使用授权书

本人完全同意北京航空航天大学有权使用本学位论文(包括但不限于其印刷版和电子版)，使用方式包括但不限于：保留学位论文，按规定向国家有关部门(机构)送交学位论文，以学术交流为目的赠送和交换学位论文，允许学位论文被查阅、借阅和复印，将学位论文的全部或部分内容编入有关数据库进行检索，采用影印、缩印或其他复制手段保存学位论文。

保密学位论文在解密后的使用授权同上。

学位论文作者签名：_____ 日期：_____ 年 _____ 月 _____ 日

指导教师签名：_____ 日期：_____ 年 _____ 月 _____ 日

摘 要

复杂场景的实时绘制在电子游戏、虚拟现实、电影特效、场景可视化、立体仿真等领域有着广泛的应用。深度图像作为复杂场景的简要近似表达,以其与场景复杂度无关的存储特性、直观的表达结果和支持 GPU 快速绘制的硬件特性在计算机图形学领域得到了广泛的应用。但是传统的深度图像存在采样点冗余、采样范围受限和采样内容单调的缺点,制约了对复杂三维场景的表达能力、降低了图形学特效的绘制效果和效率,限制了深度图像的应用范围。本文对深度图像进行了深入的研究,通过改变深度图像的采样方式和重建方法提高了深度图像对场景的表达能力;提出按需生成深度图像的相机安排策略,扩大深度图像采样内容;提出广义深度图像来扩展深度图像的表达内容。本文的具体研究内容有,基于非冗余绘制的复杂场景离散采样方法,适用于非冗余深度图像的场景重建和软阴影绘制方法,基于聚类深度图像的反射特效绘制方法和基于帧间时间相关性的反射复用方法。本文取得的主要创新结果包括四个部分:

(1) 基于非冗余绘制的复杂场景离散采样方法。

用深度图像组对场景进行离散采样重建出场景的近似表达是处理复杂场景的重要方法。但是传统的深度图像组中不同图像之间存在很高的采样冗余性,这限制了深度图像采样表达的有效性。本文提出一种基于非冗余绘制的深度图像绘制采样方法,该方法能够在深度图像生成时检测并避免捕获到冗余采样点,捕获到传统采样方法不能捕获到的非冗余像素采样点。本文还给出了视相关和视无关两种场景采样方法度量方式的定义,并且展示了非冗余绘制采样方法,相比于传统深度图像绘制采样方法和深度剥离绘制采样方法,在这两种测量尺度上能取得更好的场景采样结果。

(2) 适用于非冗余深度图像的场景重建和软阴影绘制方法。

大部分的场景采样方法需要从场景的离散表达重建出整个三维场景。相比于传统深度图像采样方式,非冗余绘制采样方式由于避免了连续表面上冗余采样点的绘制,在生成图像中会有更多的场景碎片,这给场景重建带来了挑战。本文针对非冗余绘制的碎片化特性,提出一种适合于非冗余深度图像的几何重建方法。本文将非冗余绘制算法及其几何重建方法应用到软阴影的绘制中,并且验证了在此应用场景下,相对于其他场景采样方法如传统深度图像绘制采样方法或者是深度剥离绘制采样方法,非冗余深度图像方法有更好的场景重建效果和软阴影绘制效果。

（3）提出一种基于聚类深度图像的镜面反射和光泽反射特效绘制方法。

反射场景指包含大量表面带有反射属性对象的三维场景，反射属性包括镜面反射和光泽反射（glossy）属性。目前反射的绘制方法可以分为两大类：光线跟踪方法和环境光贴图方法。前一类方法需要进行大量的三角形与光线求交计算，在场景复杂的情况下效率低下；后一类方法假设反射光线是由场景中心点发出，通过反射光线的方向在环境贴图上索引被反射像素信息。尽管这种方式易于计算，但是在大多数反射场景中反射光线是由场景中心点发出的这一条件很难满足，尤其是当反射物体离场景中心较远时反射绘制效果失真严重。本文提出一种基于屏幕反射像素聚类生成深度图像的反射场景绘制方法。该方法首先根据屏幕空间反射像素的法线信息生成屏幕空间反射像素聚类，然后根据屏幕空间聚类像素的统计信息生成适合反射绘制的聚类相机模型，再使用聚类相机生成的深度图像近似表达此聚类会反射到的场景，最后将反射光线投影到聚类相机生成的深度图像上求交计算得到物理真实的反射效果绘制。本文提出的方法相比于光线跟踪方法，能够在不明显损失绘制质量的情况下效率提升十倍以上；此外，本文的方法不需要进行预处理，能够支持全动态场景，场景中模型可移动和变形，模型材质亦可动态变化。

（4）基于帧间时间相关性的反射复用方法。

应用时间相关性（Temporal Coherence）将已经计算好带有光照信息的深度图像映射到新的相机空间下是一种常见的复杂场景快速绘制方法，这是因为通常复用已经计算好的光照信息比重新进行光照计算开销要少。常用的基于投影的时间相关性复用方法是假设像素的光照特性在不同相机视角下基本保持不变（视无关）的。但是这和反射光会根据入射光线的变化而发生变化（视相关）的特性不符。本文首先通过扩展深度图像定义使得其不仅能表示当前相机观察光线采样到的场景，还能表达反射光线所采样的被反射场景信息。然后本文提出一种反射像素在不同视点下的映射方法，应用此映射能将关键帧中的扩展深度图像中可用反射光照值投影到新视点下。基于帧间时间相关性的反射复用方法通过关键帧中的反射颜色复用，减少了反射绘制的总开销，提高了绘制效率。

关键词：实时绘制，反射绘制，深度图像，相机构造，非冗余绘制，时间相关性，屏幕聚类，光线跟踪

Abstract

Real-time rendering in complex scenes has been widely used in the fields of computer games, virtual reality, film digital effects, scene visualization and 3D simulation. A powerful approach to approximate complex scenes is to sample the scene with depth images. It has the advantages of independence of the scene's complexity, intuitively clear approximation and GPU-friendly computing property. However, the conventional depth image is restricted by its fixed projection mode, a high level of redundancy and commonly using view-independent illumination. This paper hinges upon depth-texture based techniques, aiming at the camera model that constructs the depth image and proposes cluster-based camera model to capture reflective scene, non-redundant depth image to capture the scene more efficiently, and a reflection reprojection method to reuse the already computed reflective pixel on depth image. The contributions of this paper mainly lie in:

(1) A non-redundant sampling rendering method used to capture the scene more efficiently

A set of depth image is commonly used as a powerful approach to approximate complex scene. However, conventional depth image generated from nearby viewport has a high level of redundancy, which restricts its ability to approximate the scene. We introduce a non-redundant rendering method. This method can be used to avoid rendering redundant scene before the depth image is computed by detecting and avoiding redundant samples. We prove that our method, compared to conventional depth image and depth peeling method, improves scene approximation ability due to several view-independent and view-dependent metrics.

(2) A reconstruction and a soft shadow rendering method suitable for non-redundant rendering images

Non-redundant rendering sampling images have a higher level of fragmentation, because replacing the redundant samples with new samples creates additional depth discontinuities. The fragmentation of non-redundant images is challenging for the common reconstruction approaches. We present a novel reconstruction method that is suitable for the higher degree of fragmentation characteristic to the non-redundant image. This method provides a better

geometry approximation at surface edges, while the reconstruction of a continuous surface is watertight and non-redundant, and isolated samples contribute to the reconstruction. We demonstrate quality advantages over prior-art scene discretizations such as conventional images and depth peeling in the application of soft shadow rendering.

(3) A cluster-depth-buffer based method suitable for real-time glossy and specular reflection rendering

There are two mainly used methods in reflection rendering: ray-tracing method and environment mapping method. The former method requires lots of ray-triangle intersection. The latter method is based on a fake assumption that reflected ray originates from the center of the scene. We propose a screen-space cluster method. The coherence of reflected rays is leveraged through clustering, a depth image is used to represent the geometry reflected by a cluster, and the color samples captured by the reflected rays of a cluster are computed by intersection with the depth image. We achieve quality specular and glossy reflections at interactive rates in fully dynamic scenes.

(4) A reflection rendering method using temporal coherence by reprojection already-computed depth image.

Temporal coherence is commonly exploited for rendering high-quality images at low cost by mapping already rendered depth image into a novel view. However, conventional reprojection methods using temporal coherence assume pixel value remain almost unchanged from a viewpoint to another, which is not suitable for reflection rendering. We present a novel approach to reuse reflections from a nearby depth image. Our method can save most of overall workloads of reflection computation by reusing the information from the already computed depth image, which makes rendering efficiently.

Key words: Real-time rendering, Reflection rendering, Depth image, Camera model, Non-redundant rendering, Temporal coherence, Screen-space clustering, Ray-tracing

目 录

第一章 绪论	1
1.1 研究背景与意义	1
1.2 相关研究综述	2
1.2.1 场景实时绘制方法	2
1.2.2 反射绘制方法	11
1.2.3 基于时间连续性的绘制方法	13
1.3 目前存在问题	16
1.4 论文内容安排	17
1.4.1 论文内容	17
1.4.2 论文组织与安排	19
第二章 基于非冗余绘制的场景采样方法	21
2.1 引言	21
2.2 非冗余绘制	24
2.2.1 冗余定义	24
2.2.2 非冗余绘制定义	25
2.3 实验结果和讨论	28
2.3.1 算法实现	28
2.3.2 采样有效性度量	29
2.3.3 非冗余绘制采样质量	30
2.3.4 算法讨论和局限性分析	35
2.4 小结	35
第三章 适用于非冗余深度图像的场景重建和软阴影绘制方法	37
3.1 引言	37
3.2 非冗余绘制的场景重建	38
3.3 在软阴影上的应用	41
3.4 试验结果和讨论	42
3.4.1 重建质量	42
3.4.2 阴影生成质量	43
3.4.3 阴影生成效率	47
3.4.4 算法讨论和局限性分析	48

3.5 小结.....	49
第四章 基于像素聚类深度图像的反射绘制方法.....	50
4.1 引言.....	50
4.2 算法整体流程.....	52
4.3 反射光线聚类.....	53
4.4 聚类相机的构建.....	56
4.5 反射光线与深度图像求交.....	59
4.6 光泽反射绘制.....	60
4.7 实验结果及分析.....	61
4.7.1 反射质量	61
4.7.2 时间效率分析	66
4.7.3 算法实现	69
4.7.4 算法讨论和局限性分析	70
4.8 小结.....	72
第五章 基于反射深度图像时间相关性的反射绘制复用方法.....	73
5.1 引言.....	73
5.2 算法整体流程.....	75
5.3 反射深度图像的生成.....	76
5.4 基于时间相关性的反射投影.....	76
5.4.1 最佳反射像素匹配	78
5.4.2 反射偏移	81
5.4.3 生成图像合成	82
5.4.4 关键帧选取策略	83
5.5 实验结果和分析.....	83
5.5.1 算法实现	83
5.5.2 反射生成质量	84
5.5.3 效率分析	87
5.5.4 算法讨论和局限性分析	88
5.6 小结.....	90
结论与展望	91
结论.....	91

进一步工作	92
参考文献.....	95
攻读博士学位期间取得的研究成果.....	105
致 谢	107

图 目

图 1	场景绘制方法按照几何体作为场景表示权重排序图.....	3
图 2	三维场景（左）和生成的二维公告板（右）示意图 ^[35]	5
图 3	屏幕空间下深度图像示意图	6
图 4	左：遮挡相机场景采样效果图，右：原始采样图像 ^[92]	8
图 5	使用两个平面坐标作为光场的四维参数化表示示意图.....	10
图 6	反向重投影操作示意图 ^[87]	14
图 7	研究内容整体结构图	18
图 8	非冗余绘制采样鸟巢场景视点设置图.....	23
图 9	鸟巢场景的传统绘制采样，非冗余绘制采样和深度剥离方法的对比图.....	23
图 10	冗余定义示意图	25
图 11	使用传统采样方法（左），本章方法(中)，深度剥离方法（右），采样简单场景 2D 示意图.....	27
图 12	图 11 中采样的图像和中间视点的重建图像展示图.....	28
图 13	冗余采样点占比随图像分辨率变化示意图.....	34
图 14	遮挡采样点占比随图像分辨率变化示意图.....	34
图 15	有用采样点占比随图像分辨率变化示意图.....	34
图 16	使用三种采样方法和原始图像生成的软阴影绘制图.....	38
图 17	非冗余图像的混合重建方法	41
图 18	视点在面光源中心下本章方法重建结果（左）和用真实场景绘制的图像（右）	43
图 19	视点在面光源区域外本章方法重建结果（左）和用真实场景绘制的图像（右）	43
图 20	阴影图绘制时间随场景复杂度的变化情况.....	48
图 21	不同光照反射程度下的光泽反射绘制效果图.....	52
图 22	场景反射物体可视化图	55
图 23	可视化表 12 和图 21 中屏幕像素聚类示意图.....	56
图 24	聚类相机拟合平面和远近裁剪面示意图.....	57
图 25	聚类相机视点 e_0 的构建示意图.....	58
图 26	（上）表 12 中三张本章方法生成图和光线跟踪误差图。（下）放大 50 倍后的误差图.....	62
图 27	表 12 中花瓶场景遮挡误差像素（红色）会随着聚类数目增加会减.....	63
图 28	表 12 中使用环境贴图的像素（红色高亮）	63
图 29	（左）本算法生成的反射和（右）比当前分辨率低 3×3 倍的聚类深度图像生成的反射.....	64
图 30	忽略漫反射成分的低光泽程度（左）和高光泽程度（右）本文反射方法输出图像.....	65
图 31	混合漫反射成分的低光泽程度（左）和高光泽程度（右）本文反射方法输出图像.....	65
图 32	混合漫反射成分的低光泽程度（左）和高光泽程度（右）光线跟踪方法输出图像.....	65
图 33	图 30 中反射像素使用 49 根反射光线生成的反射图.....	66
图 34	直接使用深度投影方法（左），本章方法（中）和光线跟踪方法（右）生成的反射绘制图.....	74
图 35	反射像素随视点移动示意图	75
图 36	基于时间相关性的反射投影步骤图.....	78
图 37	计算 s 关于 s_n 匹配距离示意图	80
图 38	像素映射方式示意图	82
图 39	使用深度投影方法（左），本章方法（中）和光线跟踪方法（右）在起居室和盥洗室生成的反射绘制图	85

图 40	关键帧图像（上）和投影生成图像（下）	86
图 41	图 40 中对应输出图像与光线跟踪对比误差图（上）与放大 60 倍误差图（下）	86
图 42	遮挡误差示意图，遮挡误差是由相交到墙但没有相交到后面柱子的反射光线所致	87

表 目

表 1	基于图像的绘制不同技术对照表	11
表 2	视无关度量表（像素冗余定义方式）	30
表 3	遮挡采样点（绿色+黄色）和有用的遮挡采样点（黄色）可视化图表.....	31
表 4	视相关度量表（像素冗余定义方式）	32
表 5	视无关度量表(可见性冗余定义方式).....	33
表 6	视相关度量表(可见性冗余定义方式).....	33
表 7	三种绘制采样方式的时间效率	35
表 8	使用原始图像和三种绘制采样方式生成的阴影图表.....	44
表 9	不同分辨率下三种采样方式的软阴影误差表格.....	45
表 10	软阴影误差随着光源尺寸的变化情况表.....	46
表 11	软阴影绘制时间表.....	47
表 12	用环境光贴图(左)， 光线跟踪 (中)， 和本章方法 (右)生成的反射效果图表.....	51
表 13	本章方法与 Optix 的效率对比	66
表 14	章节 4.2 中各个步骤的耗时图（毫秒）	67
表 15	不同聚类数目下的平均帧率	67
表 16	不同生成图像分辨率下的平均帧率.....	67
表 17	不同聚类生成图像分辨率下的平均帧率.....	67
表 18	漫反射三角形数目不同规模下生成图像分辨率下的平均帧率.....	67
表 19	表 13 所用四条路径中的聚类数表.....	68
表 20	图 30 和图 32 中光泽反射绘制的时间效率.....	69
表 21	表 13 所用四条路径中平面反射物像素与反射像素数目比值表.....	70
表 22	本章方法和 Optix 的效率对比表	88
表 23	算法不同步骤的平均耗时表	88

第一章 绪论

本章首先介绍了论文的研究背景与意义，其次介绍了论文所涉及的关键技术的研究现状，并对相关技术进行了较为全面的分析。然后介绍了目前深度图像作为场景表示存在的问题。最后阐述了复杂场景近似表达及绘制方法研究的核心思想，并对本文的研究内容以及文章的组织结构进行了介绍。

1.1 研究背景与意义

虚拟现实（Virtual Reality）^[1-7]是一种使用交互式硬件和软件创建并由身体感官体验的三维 360 度环境的逼真和沉浸式模拟手段。它通过虚拟现实设备营造出拥有逼真图像，声音或者其他能代替真实环境感知体验的人工环境。使用虚拟现实设备的体验人员需要无卡顿地“环顾”模拟的人工环境，这要求虚拟现实设备能够进行复杂场景的实时绘制。

复杂场景的实时绘制是虚拟现实的基础技术^[8-9]。场景的实时绘制是指绘制单元需要足够快地生成输出图像序列使得观察人员产生输出图像帧是连续不断地出现的感觉。实时绘制通常使用计算机 GPU 进行绘制加速。实时绘制最早的应用领域是三维场景仿真。近年来，随着计算机软硬件技术的发展和计算机设备的普及，实时绘制技术已经深入到人们的生产生活当中，在包括计算机辅助设计、场景漫游、远程场景可视化、医疗训练、虚拟仿真和电子游戏^[59-60,89-91]等众多领域得到了广泛应用。

随着计算机图形学理论的工业化应用，特别是三维扫描技术精度的提升和虚拟场景建模技术的不断发展，三维场景模型的场景复杂度也随之增高、表现细节也不断增强。复杂场景在带来与真实场景物体相似的视觉感知特性的同时增大了场景绘制的开销，降低了人机交互的响应速度。当绘制场景的几何体尺寸小于屏幕上的一个像素时直接使用几何体进行绘制就会引入不必要的绘制开销。为了提高绘制效率，常常对场景进行预处理操作生成适合实时绘制的数据结构。但是这要求场景必须是静止的，不存在动态可变形几何体，这导致了大多数预处理算法不适用与动态复杂场景的实时绘制。另外复杂场景中的时变材质属性和复杂光照条件因其多样性也增加了复杂场景实时绘制的难度。

为了实时、快速、有效地进行动态复杂场景绘制，使用场景近似表达而不是直接真实场景几何体进行绘制的方法得到了广泛的应用。深度图像^[10]，即图像中每个像素在记录采样点的着色颜色值之外额外存储了采样点深度的图像，因为它易于计算和硬件友好

的特点，常被作为观察相机视点下采样到的几何场景的近似表达。深度图像在很多绘制算法中不仅可以作为最终绘制输出显示结果，还可以作为数据储备的一种中间媒质，供算法进一步计算使用。随着人们对计算机图形绘制的逼真性的不断提高，越来越多基于物理理论的绘制方法用于场景绘制算法中，这些方法通常比早期只提供视觉相似性而缺乏物理根据的方法的计算更为复杂，处理步骤更多。为了将深度图像场景替代方法应用在基于物理的绘制计算中，人们对深度图像的场景表达能力和基于深度图像的绘制结果逼真度提出了更高的要求。因此，如何进行深度图像对场景的充分采样和有效表达是我们需要研究的科学问题。此外，随着网络时代的到来，基于主机/客户端形式的三维场景远程绘制方式成为场景绘制的新模式。这种方式通常在计算能力更强的主机端使用复杂场景模型计算出包含颜色光照信息的深度图像序列组传输到轻量级的客户端（如笔记本电脑、平板电脑和智能手机）上进行终端显示变的日益流行。这种基于深度图像的绘制方式因为其紧凑的数据结构和有效的场景表达能力，使得远程场景复杂光照的绘制成为可能，具有广阔的应用前景。

本课题来源于国家自然科学基金项目“时变复杂场景的图像空间光照算法研究”。

1.2 相关研究综述

本文所涉及的关键技术包括：场景绘制方法、反射绘制以及基于时间连续性的绘制方式等。下面对本文研究内容所涉及的相关国内外研究现状进行分析。

1.2.1 场景实时绘制方法

三维场景实时生成输出图像有两种截然不同的解决思路，它们分别是基于几何绘制（Geomtry-Based Rendering）和基于图像绘制（Image-Based Rendering）。目前很多高效绘制方法的研究工作结合使用了这两种绘制思路。图 1 给出了场景绘制方法按照几何体作为场景表示重要度权重的排序图。绘制方法按照最左边完全由几何体对场景进行建模的绘制方法思想到最右边完全由图像对场景进行建模的绘制方法思想排序。在两种绘制思想之间是一系列结合应用两个思想的方法类别，基于几何绘制或者基于图像绘制贡献的权重决定了他们排列的顺序。

光线跟踪

光线跟踪是一种基于光学物理模型生成输出图像的绘制技术。在光线跟踪技术里面三维场景通常由显式的经典几何图元构成，比如三角形网格。光线跟踪首先通过跟踪图

像平面上光线穿过像素的路径追踪到光线与场景对象的相交点，再通过光线与材质的交互效果来生成输出图像。光线跟踪技术能产生高度真实感的图像，但相比于光栅化绘制技术，光线跟踪的时间开销更高，因此光线跟踪技术更适用于不强调实时性能的电影及电视特效的生成，而不适合于场景的实时绘制。

光线追踪计算耗时的原因来源于光线与场景网格复杂的线面求交计算，对其优化的加速策略主要思想是通过避免光线与不相干场景的相交计算来减少无效的求交计算。光线跟踪的加速策略包括层次细节方法^[11]、KD-trees 空间划分结构加速方法^[12]、层级包围体(BVH) 物体划分结构加速^[13] 和 GPU 端的并行加速方法^[14-16]。

光线跟踪可以配合 Monte Carlo 采样方法和物体表面反射模型（Bidirectional Reflectance Distribution Function, BRDF）通过在采样点按照 BRDF 概率分布生成的大量采样光线生成基于物理的高质量绘制图像。光线跟踪的另一个研究方向是减少基于物理绘制生成图像的方差，研究成果包括光线束跟踪^[16]、圆锥追踪^[18-19]、线段追踪^[23]和采样模式研究^[20-22]。

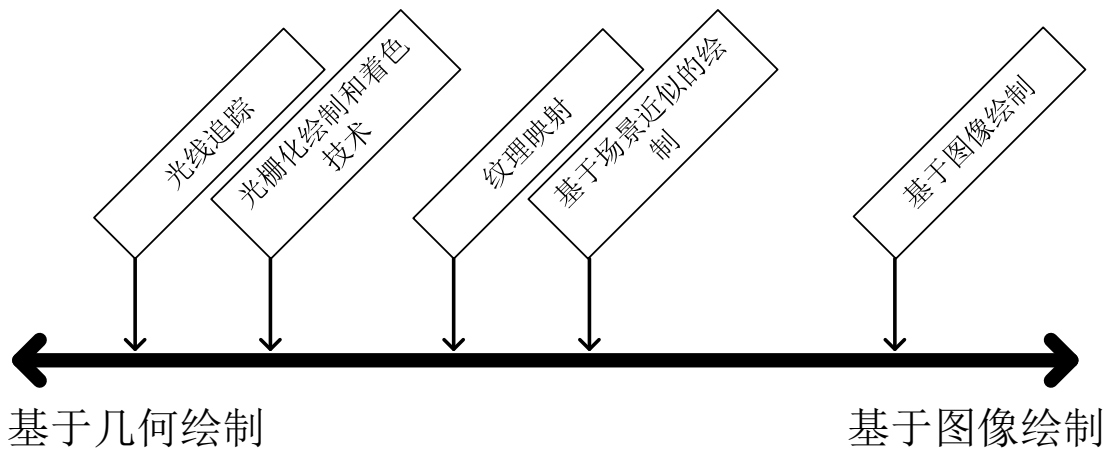


图 1 场景绘制方法按照几何体作为场景表示权重排序图

光栅化绘制和着色技术

光栅化算法是一种将场景图元转换为像素的技术。它同光线跟踪算法一样使用场景几何体生成输出图像。场景通常由三角形或者三角形网格组成。光栅化算法首先把组成图元的三维顶点流投影到二维空间上，然后对二维空间上的映射图元区域进行着色操作生成输出图像。

二维投影图元的着色操作是为了计算图元的近似光照。随着计算机图形学理论的发展，实时光照能支持的着色模型也变的越来越复杂。Gouraud 着色^[24]是一种简单的由图

元顶点处局部光照值线性内插值生成图元投影区域颜色值的方法。因为 Gouraud 着色绘制的着色信息能在图元的边界上连续变化，所以它生成的图元颜色能平缓过度。尽管这种方法提出的时间较早，Gouraud 着色仍然被当今的绘制流水线采用。Phong 提出一种模拟高光材质的更准确的着色方法^[25]。他同时提出使用法线而不是颜色进行图元的内插值，使光照模拟更加准确。Blinn^[27]采用 Torrance-Sparrow 的场景微表面模型^[26]来描述光线与实际粗糙平面的交互效果。与 Phong 着色方法相比，这种着色方法通过微表面分布函数模拟物体表面的粗糙程度，能生成更接近于物理真实的光照图像。所有这些着色方法都是局部着色方法，因为他们没有考虑场景中像反射一类的全局光照信息。

光栅化算法和局部着色模型因为能很容易地和局部光照模型及深度缓存算法集成在 GPU 硬件上，所以在实际应用中，光栅化算法和局部着色模型是当前最流行的实时三维计算机图形学绘制算法。

纹理映射

为了增加基于几何体绘制方式生成图像的真实感，研究人员提出纹理映射方法旨在通过加入基于图像的信息来增强生成图像的着色质量。纹理映射方法可以定义为二维图像对三维表面的映射方法。使用纹理映射的绘制可以在只增加相对较少的计算开销情况下极大地丰富输出图像的细节。

纹理映射在实时绘制上有很多应用，他们的主要区别在于映射到场景表面上的不同属性。Catmull^[28]最早将图像色彩（颜色值）映射到图形表面。他最早发现通过建立图形表面的二元变量和图像之间的联系就能进行图像颜色的纹理映射。他同时指出了纹理映射有反走样的问题。Blinn^[29]提出在着色计算之前使用纹理映射控制表面法线方向的扰动，这种方法后来称为凹凸贴图技术，它能绘制出具有几何真实感的场景褶皱而不需要改变原始场景几何体。Gardner^[30]将透明值（alpha 值）映射到表面上来做云的视觉仿真。由于位置属性也是场景表面的一种属性，Cook^[34]提出一种映射位置偏移量的纹理映射方法来改变图元的位置。

纹理映射的另外一种应用方式是首先由 Blinn^[31]提出的环境光映射贴图。与纹理映射相比，环境光贴图方法不是将图像映射到场景中的某一个物体表面而是映射到一个以场景中心为球心拥有无限半径的球或者立方体上。环境光贴图方法使用表面的法线或者光线的方向进行索引，这两种索引方式分别作为漫反射环境光贴图^[32]和镜面反射环境光贴图^[31]的基本思想。在这两种方法中，环境光贴图方法分别作为镜面反射中光线跟踪和漫反射绘制中全局光照方法的简要近似。环境光贴图方法也可映射透明度^[33]，这种方式

被使用在折射环境光贴图方法中。因为环境光贴图方法对于某一采样点的着色不依赖于光源的数量, 所以环境光贴图方法更适合于复杂环境光条件下的光照模拟。

基于场景近似的绘制

大多数不管是人工还是自然生成的场景, 都趋向于使用高度细节的物体来提高建模的准确程度。当绘制场景足够复杂以致于几何体尺寸小于屏幕上的一个像素时直接使用几何体进行绘制就会引入不必要的绘制开销。因此为了实时、快速、有效地进行场景绘制, 使用图像作为场景近似表达而不是直接使用真实场景几何体进行绘制的方法得到了广泛的应用。图像作为场景替代方法和纹理映射方法有相似性, 他们都是在基于几何绘制方式之上额外使用基于图像的绘制。不同点在于纹理映射使用图像提高着色质量, 而图像作为场景替代的绘制方法是用图像表示场景的几何体与材质。

公告板技术^[35-36]是当今图像场景替代方法中一种简单并被广泛应用的技术。公告板技术通过将二维图像纹理映射到透明四边形上使得该四边形所在区域看起来有几何细节。公告板技术的关键是在不超过用户指定帧率情况下如何指定哪些公告板需要绘制使得生成图像质量最大化。映射的二维图像纹理可以预计算生成也可以计算时生成^[35]。公告板技术在视点较远且在特定理想角度观察下时能提供很好场景的近似效果。当视点从理想角度偏离时使用公告板技术绘制的图像质量就会下降。当视觉误差增加到给定阈值范围以上后原先使用的公告板就需要由一个更准确的公告板替代。图 2 展示了生成公告板的三维场景和在生成角度以外视角观察下的公告板成像。



图 2 三维场景(左)和生成的二维公告板(右)示意图^[35]

相比于公告板使用二维颜色图像作为场景近似, 深度图像^[10]技术通过额外写入视点下采样场景的深度信息大幅提高了场景建模的有效性。深度图像方法延续了偏移贴图^[34]通过纹理像素表示几何位置的思想, 通过在图像中每个像素的 A 通道额外记录场景点到观察视点深度来近似表示视点观察下场景(如图 3 所示)。而场景的深度可以通过观察相机的参数转换成深度图像场景采样点的三维空间坐标。

深度图像作为场景近似绘制技术的应用有阴影图算法和屏幕空间绘制算法。

阴影图方法^[37]是一种使用深度图像表示场景遮挡的最常用的实时阴影绘制方法。它的思想是首先在光源空间下绘制出一张深度图像（阴影图），这个图像记录了阴影光线与场景的最近交点。然后在当前相机绘制场景时候，每个场景绘制片断都需要投影到光源空间下，在光源空间下进行变换后的深度值与阴影图上的对应索引深度大小比较，如果变换后深度离光源近，该场景采样片断是照亮的，反之则在阴影里面。

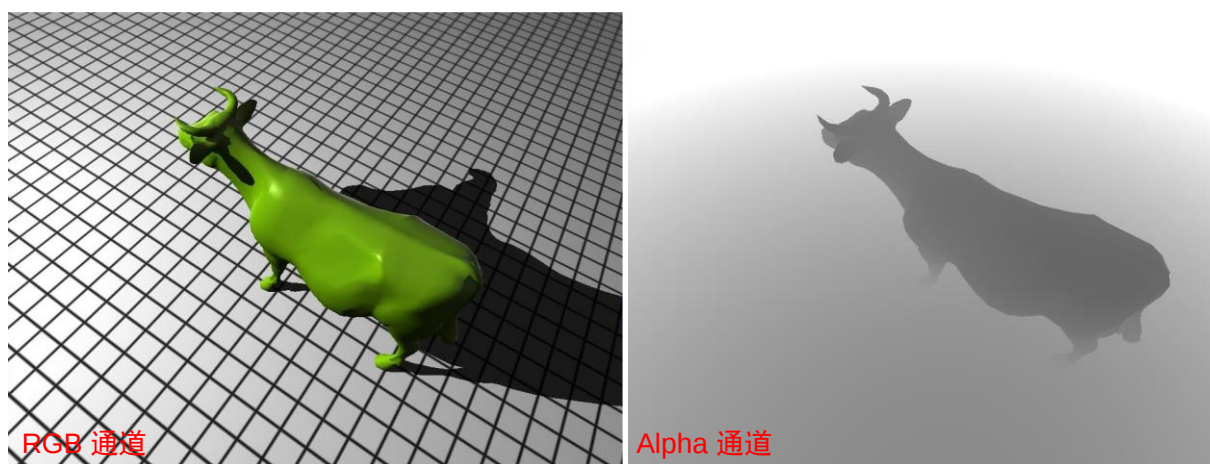


图 3 屏幕空间下深度图像示意图

阴影图方法有几个缺陷限制了它的应用。第一个缺陷是阴影图捕获场景视角有限，阴影图只能捕获固定视角下的遮挡物场景。对于位于场景中央的点光源则需要全景视图下表达整个场景对点光源的可见性，而常规阴影图方法只能捕获相机视景体下的场景。为了解决这个问题，可以对每个点光源使用六个阴影图形成立方体阴影贴图近似整个场景，但是这带来绘制效率上的损失。Brabec 等人^[41]提出将场景划分成两个半球，然后使用非线性的椭圆映射方式分别映射到两张深度图像上。但是这种方法的难点在于如何使用图像硬件的光栅化算法对椭圆映射算法后的图形（直线映射成弧）进行光栅化。Gascuel 等人^[42]通过对场景的三角形进行了充分的曲面细分，使得椭圆映射算法后的阴影图可以直接由图像绘制硬件生成。

第二个缺陷是由于阴影图采样频率和生成图像采样点投影到阴影图像上的采样频率不一致导致的走样问题。Reeves 等人^[38]提出一种将着色点投影到阴影图上后在投影区域范围内随机产生采样点进行可见性判断，最后对区域内可见性判断布尔值进行过滤的方法减少阴影的走样问题。Yang 等人^[40]提出一种使用方差的方法减少阴影图像方差。Dimitrov 和 NVIDIA 公司^[39]提出用层级化阴影图（Cascaded Shadow Map CSM）来处理远处场景的可见性判断走样问题。通过对场景进行分层并为每层场景建立适合分辨率尺寸的阴影图能有效增加室外场景如地形场景和森林场景的阴影绘制质量。

深度图像作为场景近似的绘制技术的另一个重要应用是屏幕空间绘制算法。屏幕空间绘制算法利用了深度图像与场景复杂度无关的特性，将着色过程和几何体处理过程分离来加速着色的绘制。这种方法由两遍绘制构成，第一遍绘制在观察相机视点下绘制场景，将场景的几何信息记录在被称作几何缓存的深度图像中而不进行着色操作。第二遍绘制进行着色操作，这个阶段将着色函数应用在缓存上已经保存好的几何信息上然后将着色结果作为输出生成图像。屏幕空间绘制算法通过只对视点可见的像素采样点进行着色操作节省了最费时的光照着色计算。

由于在深度缓存中可以访问整个缓存上的深度和其他像素的信息，研究人员提出多种屏幕空间内全局光照算法。Filion 和 McNaughton^[43]通过着色点周围的区域随机采样生成屏幕空间全局着色。Soler 等人^[44]通过首先在深度图像上执行 mipmap 操作，然后在每一层图像进行全局光绘制来生成全局光照图像。Shishkovtsov^[45]在 S.T.A.L.K.E.R 引擎开发时通过把屏幕空间绘制扩展到漫反射表面上实现全局光照明绘制。除开屏幕空间绘制算法在全局光照上的应用，屏幕空间算法还被应用在模拟光线跟踪^[106]、次表面散射^[46]、运动模糊^[47]、景深效果绘制^[43]和水面效果绘制^[48]上。

屏幕空间绘制算法的一个缺陷在于深度图像只会捕获到离相机最近的采样点而不能处理第一层采样点后面的场景，这样屏幕空间绘制算法就不能处理涉及到视点下被遮挡场景的绘制，比如透明绘制。深度剥离（Depth Peeling）算法^[50]使用多遍绘制模式，每一遍采样比上一遍更深的场景，来解决只能采样到视点下最近场景的问题。这种方式可以采样到包含第一层采样场景之后的多层几何场景信息。深度剥离方法在透明绘制、复杂物体绘制，例如树的绘制^[49]，有广泛的应用。深度剥离方法因为只需要传入已生成的深度图像进行深度比较来剥离冗余采样点的绘制所以绘制速度很快。双向深度剥离方法^[50]充分利用 GPU 的深度最小最大比较操作来在每一遍的绘制之中同时获得深度的最远和最近采样片断，从而提高了剥离的效率。多层剥离^[51-52]技术依赖 GPU 的多绘制对象（Multiple Render Targets (MRTs)）进一步地提高每遍绘制的剥离效率。

分层的深度图像也可以解决只能采样到视点下最近场景的问题。这种方法包括分层深度图像（Layered depth image, LDI^[53]）和分层深度图像树（LDI trees^[54]）他们可以在不同尺度上提供场景的多层采样，从而支持场景几何细节的多层级模拟。层级化深度图像的创建过程是首先绘制出多张深度图像，然后通过构建出不同深度图像之间的联系来去除冗余采样点。这种方法的缺点主要有三个，第一是创建过程比较耗时，不适合需

要每一帧都实时构建场景近似表达的动态场景；第二是层级化深度图像存储的采样点集合仅仅是常规深度图像绘制得到的采样点的并集，而没有去产生新的采样点。如果要得到完备的采样点集就需要大量的构建深度图像；第三是采样点的邻接信息很难从分层深度图像中获取到，这样加剧了场景重建的难度。

屏幕空间绘制算法的另一个缺陷跟阴影图算法一样来源于深度图像捕获场景视角有限。在通常的图形学绘制中使用传统的小孔相机模型进行三维场景向二维图像投影转化。针孔相机模型构造简单，适合于 GPU 绘制流水线进行绘制，仅需要一次矩阵乘法就可完成三维场景在二维图像上的投影操作。但是小孔相机模型采样范围有限，采样点局限在可视范围内，这导致了某些着色方法如全局光照绘制中因为只是用屏幕空间场景来模拟全局场景而在绘制效果上产生缺陷。

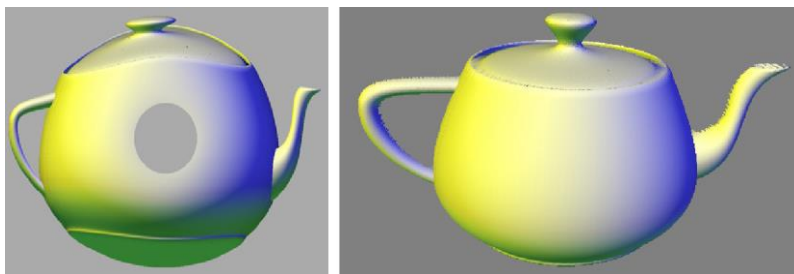


图 4 左：遮挡相机场景采样效果图，右：原始采样图像^[92]

Mei 等人^[92]提出遮挡相机概念（Occlusion Camera），通过对模型深度不连续区域部分遮挡的光线进行扭曲，从而实现对物体轮廓后面遮挡区域的场景采样，提高了对场景的表达能力，但是这种方法使用一种非直观的投影扭曲方式，不能处理复杂场景。图 4 展示了遮挡相机采样的效果图，通过对茶壶边缘处空间扭曲投影采样到原始图像捕获不到的茶壶底部表面信息。Popescu 等人^[93]提出一种可降低复杂场景非遮挡性错误的深度不连续遮挡相机（Depth Discontinuity Occlusion Camera, DDOC），该相机同样地通过使用由图像空间的深度不连续点生成的扭曲图，对观察光线进行扭曲投影，生成能采样到更多遮挡区域的扭曲图像，并将此技术应用于浮雕贴图绘制和软影计算绘制中。该算法以额外的扭曲图的绘制生成为代价获得采样光线的扭曲采样，但是增加的采样区域范围不能完全覆盖整个场景。Rosen 等人^[94]引入一种双极遮挡相机模型（Epipolar Occlusion Camera, EOC），通过对每个横向像素行中的采样深度不连续点生成额外采样相机观察遮挡局域，然后用像素的平移和新采样像素点的插入构成新的场景采样图像。该方法可以控制需要额外采样的场景区域范围，而且生成过程与场景复杂度无关，但双极遮挡相机方法支持的场景需要人为划分遮挡物和背景物。

在此基础上, Cui 等人^[95]提出了曲线相机计算模型, 将多个视点无缝集成在一个相机中, 以及光线在多个相机空间中的平滑过渡。为了消除传统针孔相机模型要求对光线进行均匀采样的限制, Popescu 等在^[96]中提出了通用针孔相机计算模型, 对光线可以根据用户兴趣或绘制效果的需求进行非均匀采样, 并通过映射函数恢复出正常的规则像素点阵图像。

基于图像的绘制

之前介绍的图像作为场景近似表达的绘制方法中, 图像是作为限制在某一相机视点下或者某些物体的场景中间近似表达。在基于图像的绘制方式技术中, 场景完全由图像进行建模。基于图像的绘制方式与用图像作为场景近似有相似点, 但是之前描述的方法都以几何为中心的绘制, 基于图像绘制是以图像为中心, 也就是说, 基于图像绘制完全不像之前所述方法那样依赖几何信息。所以这种方法能完全体现出使用图像的优点: 绘制方式完全跟场景复杂度无关和绘制质量完全取决于输入图像的分辨率。但是, 基于图像的绘制方式大体受限于静态场景的静态光照。而且, 在有些基于图像的绘制之中, 相机的移动被限制在一定范围之内。

Adelson 和 Bergen^[56]提出的一种可以准确描述基于图像绘制问题的全光函数 (plenoptic function)。全光函数是一种参数化表示空间任意一点 (V_x, V_y, V_z) 、任意角度 (θ, φ) 、任意时间 t 所看到的任意波长 (λ) 的光线强度的函数。完整的全光函数是一个 7 维表示函数, 可以由公式 1 进行表达。

$$P_7 = P(V_x, V_y, V_z, \theta, \varphi, t, \lambda) \quad 1$$

基于图像的绘制可以看作是一系列从离散图像样本 (完全或不完整) 构建全光函数的连续表示的技术。如何从全光函数进行采样以及如何从离散图像样本重建连续函数是基于图像的绘制的重要的研究课题。

通过对全光函数施加限制可以降低函数的变量维度。McMillan 和 Bishop^[58]通过丢弃两个变量, 即时间 (因此是静态环境) 和光波长 (因此是固定的照明条件) 引入五维全光模型 (plenoptic model) 概念, 可以由公式 2 表示。

$$P_5 = P(V_x, V_y, V_z, \theta, \varphi) \quad 2$$

Levoy^[55]和 Gortler^[57]使用一种有向直线参数化方式把五维空间表示缩减到四维自由空间, 自由空间是指中间没有遮挡的空间。他们提出使用两个任意平面与有向直线段的

交点来参数化表示四维光场模型。图 5 展示了四维光场的参数化过程。有向直线段同第一个平面的交点坐标为 (u,v) ，第二个平面的交点是 (s,t) 。给定有向线段可以由 uv 平面上的交点和 st 平面上的交点连线定义的四维变量组 (u,v,s,t) 表示。四维光场空间可以由式 3 给出。

$$P_4 = P(u, v, s, t) \quad 3$$

Levoy^[55]同时描述了四维光场生成观察视角下的输出图像的采样方式。上述方法的光场数据是由二维相机阵列在虚拟场景绘制而成，或者是在真实场景拍照而成。基于图像的绘制方法关键在于足够高的采样率，在^[55]中描述了一种压缩光场数据的方法，使用这个方法生成的光场存储容量是原有方式的百分之一并只带来很小的可靠性损失。用两个平面交点对光场进行参数化是四维光场的一种参数化方法。其他的光场参数化方法包括球光场^[101]、各项同性光场^[102]和均匀采样光场^[103]。Sloan 等人^[103]和 Katayama 等人^[104]通过限制相机在一条直线上提出更严格的光场表示模型。

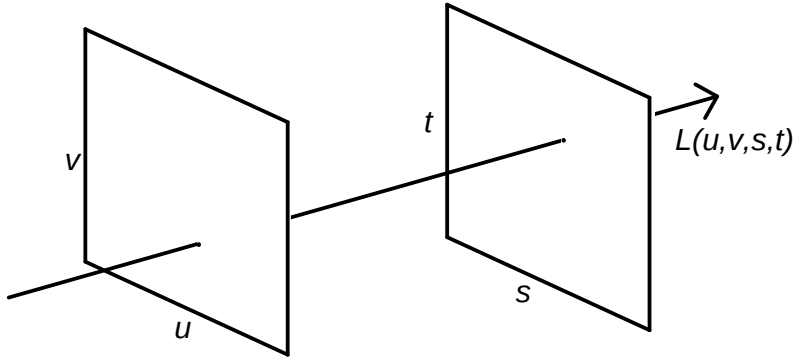


图 5 使用两个平面坐标作为光场的四维参数化表示示意图

最简单的全光函数模式是观察点固定的二维全景图环境方法^[98-99]（环境光贴图方法），即 $P_2 = P(\theta, \varphi)$ 。在固定视角下生成的常规图像可以看作是视角受限的非完全全景图。

显然，视点限制得越多，全光函数的表达越简单。如果让视点是自由变化，则需要 5 维全光模型，如果视点只在两个平面的自由空间中，可以由四维光场来描述。如果视点固定，可以由环境光贴图来表示。一种叫做同心拼图^[100]的方法可以作为全光函数的三维表示。它的视点范围被限制在平面圆内，通过每个圆的不同位置拍摄的图像的窄缝来合成同心拼图。同心拼图以 3 个参数表示输入图像的所有光线：半径，旋转角度和垂线距离。它通过有效地组合捕获图像的窄缝来绘制新视图。虽然在生成图像中存在垂直失真，但是它们可以通过深度校正来缓解。同心贴图具有良好的空间和计算效率。与

光场方法相比，同心贴图所需文件大小要小得多。

归而总之，7 维全光函数的不同降维表示方法如表 1 所示：

表 1 基于图像的绘制不同技术对照表

名称	维度	视点范围
全光函数	7	自由
全光模型	5	自由
光场相机	4	三维包围盒内
同心拼图	3	二维圆圈内
全景图	2	固定点

使用基于图像绘制方式的好处是避免了场景的重建。但是，这种好处带来的开销是需要使用大量的图像对场景密集采样，这样生成高质量的输出图像就需要处理和传输大量的图像。换言之，如果捕获场景的数目比较少，由于欠采样导致的内插值缺陷就会非常明显而影响到整个生成图像的绘制质量。

1.2.2 反射绘制方法

反射材质属于场景表面材质的一部分。当物体表面材质足够光滑时，镜面材质可以反射呈现出被反射场景物体。当表面材质光洁度下降时，可以产生光泽反射效果，即距离反射对象近的物体的反射清晰，距离远的物体的反射模糊。反射物体的反射成像与观察相机的位置所决定的入射光线朝向有关，不同观察位置发出的不同观察光线会产生不同的反射成像。入射光线和相交点法线按照反射定律决定出射的反射光线朝向，所以反射光线也被称为二阶光线。反射效果影响场景绘制的真实感体验，我们把反射的绘制分为如下大类。

光线跟踪

光线跟踪方法能支持镜面和光泽反射绘制，但是速度很慢。速度慢的原因之一来源于每个像素需要足够多的反射光线进行反射反走样操作。当一个反射像素反射角较大时，需要大量的光线对被反射角度进行采样，增加了光线跟踪的计算量。原因之二来源于光泽反射表面。镜面反射产生的反射光线只需要很少量光线就能足够采样。但是光泽反射的表面产生的是非连贯的反射光线，每个像素都需要大量的采样光线。原因之三来源于动态场景。在动态场景中，光线跟踪的用于加速的数据结构需要每帧进行更新，这

是个耗时的操作。

基于反射近似投影的光栅化绘制

反射近似投影方法首先将被反射场景投影在给定反射物表面,然后使用光栅化绘制流水线在反射物体上生成反射效果,这种方法的绘制帧率很高。基于爆炸图的方法^[70]通过对被反射物体的变形和转换投影到成像平面上绘制出反射图像。*Estalella* 等人^[71]将反射物的顶点和法线绘制到纹理,然后并行式地通过片断着色器局部搜索寻找被反射物体的近似投影点。用被反射物近似投影成像生成最后输出图像有三个问题,首先需要在反射绘制时明确反射物和可能的被反射物体;其次,如果场景包含许多反射物,则需要对每个反射物体生成被反射物的投影成像。最后,还需要解决被反射物在成像上的遮挡问题。基于采样点的相机^[72]能解决上述问题。它是一种构建在场景划分 **BSP** 树上的一组采样相机集合,它能为曲面绘制出准确的反射,但是复杂度会随着场景中反射对象的数目急剧增加。

基于场景近似的反射绘制

基于被场景近似的反射绘制的主要思想是根据反射物表面的 **BRDF** 计算出反射光线,然后用图像模拟被反射物体场景。这类方法包括立方体贴图,环境光贴图^[65-66]。*McCool* 引入环境光贴图的高光过滤优化策略来绘制有光泽 **BRDF** 反射模型的物体。*Green* 等人^[65]使用 **BRDF** 的非线性近似作为各向同性高斯函数的加权和,并且通过预计算自遮挡来实现准确和有效的环境光绘制。*Kalaset* 等人^[66]通过给环境光贴图添加深度来计算反射光线或者折射光线的交点。*Popescu* 等人^[67]提出了一种使用深度图而不是实际几何构造图像的方法来加速反射计算。*McGuire* 和 *Mara*^[68]提出了一种用屏幕空间深度图像模拟被反射场景的 GPU 反射计算方法,在此基础上 *Vardis* 等人^[69]使用多视点下的多层深度图像来计算更精确的反射效果。但是以上场景近似方法存在过度近似或欠近似被反射场景的情况。基于场景近似的反射绘制方法的主要难点是如何设计既能够按需捕获所有被反射物体又能够方便反射的计算的场景近似策略。

基于图像的绘制

基于图像的绘制方法通过预计算和预存储的图像采样点数据库来计算反射光线。光场方法^[55,57]通过存储不同视点下的反射,然后由新视点在采样图片中的索引和内插值合成反射图像。*Heidrich*^[61]通过在光场中存储光线到光线的映射使用户能改变反射体和折射体的位置。*Taguchi*^[62]等人提出一种轴向环绕反射物体建立相机组由外向内对光场进行参数化的方法。*Yu*^[63]等人结合环境光贴图方法和光场方法提出一种环境贴图光场方法

在反射物体周围由内向外进行光场参数化。

基于图像的绘制方式不适合做镜面反射的绘制。因为镜面反射会随着视点的微小变化而变化，这导致了需要存储的采样集合数目过大。其次基于图像的绘制不适合动态场景，因为动态场景需要随时对存储的采样库进行更新，而更新操作开销太大。

1.2.3 基于时间连续性的绘制方法

当今社会对高质量图片的生成有很高的效率要求。传统的提高绘制效率的方法是单独提升每个运算单元的运算时间，但是，当帧率很高的时候，帧与帧之间的差异会很小，可见三角形在一段时间内依然可见而且生成图像采样点的属性（比如颜色）趋向于保持不变。这种时间连续性就给效率优化提供了另一种解决思路：可以通过复用参考帧的数据（中间结果，或者最终颜色信息）来加速后续输出图像的生成。本节由帧间时间连续性的映射方法和基于时间连续性的绘制应用组成。

1. 帧间时间连续性的映射方法

(1). 反向重投影方法

一种简单的使用时间相关性的情况是使用已绘制好的关键帧数据去产生新的输出图像，对于新的输出图像中的每个像素，我们都可以追踪到它在关键帧中的映射位置来判断它在关键帧中是否可见。如果关键帧中映射位置的像素能够被复用，映射点的值就直接被复用在当前像素上而不需要重新进行复杂的计算；否则就必须重新计算值。这个方法被叫做三维图像变换或者反向重投影 (Reverse Reprojection Cache)，最早被 Scherzer 等人^[73]和 Nehab 等人^[74]分别独立提出，并且被用作大部分基于时间连续性绘制方法的框架。

定义在时间 t 时生成的关键帧图像 f_t ， f_t 记录该帧屏幕空间下可见像素。除图像 f_t 之外，我们还记录一个伴生深度缓存 d_t 记录屏幕空间下的深度值。我们用 $f_t(p)$ 和 $d_t(p)$ 表示屏幕像素 p 点在关键帧图像 f_t 和伴生深度缓存 d_t 索引值。对于时间 t 时生成图像帧中坐标为 (x, y) 的像素点 p ， $p=(x, y)$ ，可以计算出产生像素 p 的三维场景几何体在关键帧中剪切空间的位置值，用 $(x', y', z') = \pi_{tk}(p)$ 表示。这里用反投影算子 $\pi_{tk}(p)$ 表示把像素点 p 映射到在时间点 tk 生成的关键帧图像对应相机剪切空间位置。注意到使用这个操作我们同时可以得到帧 tk 中的对应场景几何体深度 z' 。这个深度可以用来检测当前像素 p 在关键帧是否可见。如果投影点的深度 z' 和 $d_{tk}(x', y')$ （在一定阈值之内）相等，我们认为当前像素 p 和关键帧中投影像素 $f_{tk}(x', y')$ 是由同一个场景点生成。在这种情况下

下投影像素索引值可以被复用。否则这两个像素就没有时间相关性，这种情况下用 $\pi_{tk}(p) = \varnothing$ 表示，这意味着关键帧缓存未命中。使用物体编号相等作可见性判断能加强缓存命中的准确率。图 6 展示了反射重投影操作。着色信息和像素深度保存在关键帧屏幕空间的深度图像上（左）。对于时间 t 时屏幕空间内每一个像素 p ，反射重投影操作用来寻找时间 tk 时屏幕空间的对应位置 $\pi_{tk}(p)$ ，同时重计算出深度 z' 与储存深度进行比较。如果比较结果一致，就认为找到索引点（ p_2 ，及绿色区域）。否则就认为关键帧缓存未命中（ p_1 ，及红色区域）。

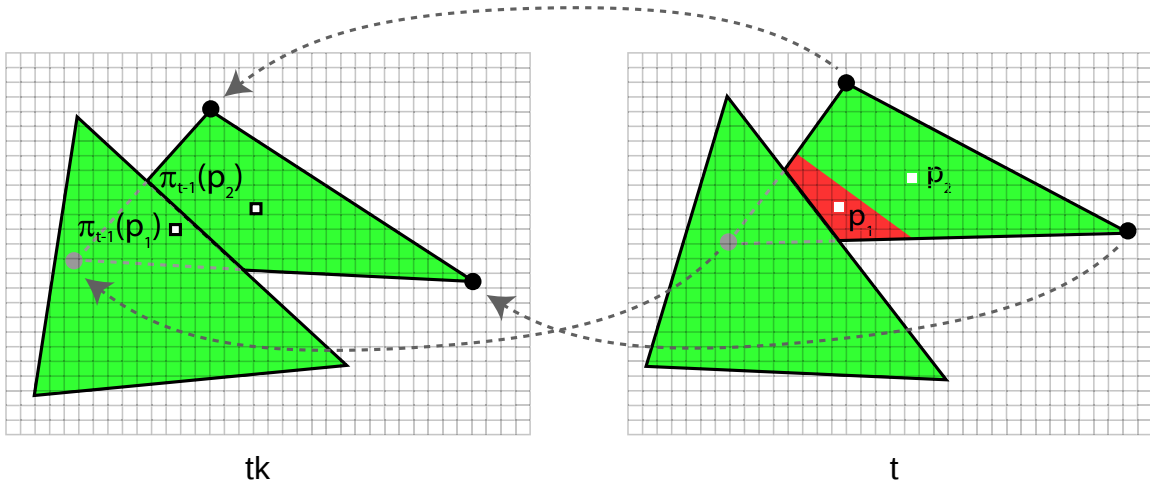


图 6 反向重投影操作示意图^[87]

(2)前向投影方法:

另一种生成输出图像的方式是前向投影方法，与反向重投影方法中以目标帧的每个像素作为计算单元不同，前向投影方法直接通过对缓存进行操作把关键帧中的每个像素映射到新的位置。这有一个好处是不需要新的输出帧中处理场景几何体。但是，他需要对关键帧中每个像素产生一个前向投影矢量（或者叫做偏移矢量），这相当于 $\pi_{tk}(p)$ 的反向映射。

但是对每个像素进行前向投影方法在当前显示硬件下比较困难而且费时。这是因为它涉及到一个溅射操作，这在当前硬件下支持不是特别充分。Didyk 等人^[76]提出一个回避这个问题的方法：他们提出一种在当前硬件能充分支持的图像扭曲方法。图像扭曲方法假定像素的移动是分段线性化的，这样可以由运动矢量场和一个稀疏网格模拟关键帧进行前向投影。他们之后提出一种基于自适应网络的改进方法^[77]。Yu 等人^[78]提出利用 GPU 的并行处理能力（可由 CUDA 或者是 DirectX 11 的计算着色器实现）进行溅射操作的前向投影方法。Yang 等人^[79]提出一种使用当前硬件像素着色器，使用像素收集操

作而不是溅射操作的基于图像的前向投影算法。

2. 基于帧间时间连续性的绘制应用

(1) 阴影绘制

阴影图算法是最常用的实时阴影绘制算法。阴影图算法中的一个问题是由于阴影图采样率有限导致的阴影走样问题。Scherzer 等人^[74]中为了解决这个问题，通过每帧对投影机进行不同程度的微小偏移生成有不同偏移的阴影图，然后再通过结合连续多帧阴影图的阴影比较结果来进行阴影反走样，生成更高质量的阴影结果。

阴影图算法的另一个问题是生成软阴影质量不高。光源采样算法^[81]通过为每个面光源离散采样点（每个采样点在光源上不同位置）产生一个阴影图，然后为每个像素计算平均阴影图比较结果为软阴影强度。给定像素 p ， n 个阴影图产生的软阴影值可以由式 4 给出：

$$\Psi(p) = \frac{1}{n} \sum_{i=1}^n s_i(p) \quad 4$$

这个方法的缺陷是需要足够多的采样点才能产生够软的阴影。基于时间连续性算法的目标是把空间上的采样转换到时间上，比如一种可行方法是把软阴影在空间上由不同随机光源采样点进行积分计算的方式改变为空间上为单个随机光源采样点进行时间维度上的积分。这种方法对对称光源来说非常容易实现^[82-83]。面光源采样可以通过每帧进行一次阴影图判断来迭代生成^[85]。基于帧间时间连续性软阴影生成公式由式 5 给出：

$$\begin{aligned} \Psi(p) &= \frac{s(p) + \sum (p)}{n(p) + 1} \\ \sum (p) &= \sum_{i=1}^{n(p)} s_i(p) \end{aligned} \quad 5$$

这里 $s(p)$ 表示 p 像素的硬阴影(单个光源采样点阴影图)查询结果， $n(p)$ 是参考帧中的累计比较的阴影图总数。注意到 n 对每个像素是独立的，它取决于这像素在多少帧中可见。这个方法可以通过把 $n(p)$ 和 $\sum(p)$ 值存储在缓存中很直观地得到软阴影时间维度积分结果。

(2) 屏幕空间环境光遮挡绘制

环境光遮挡^[84]是一种简易有效的透过度量采样点的空间半球有多少比例被遮挡模拟采样点全局光照的方法。屏幕空间环境光遮挡(SSAO)使用深度图像作为场景几何的近似。屏幕空间环境光遮挡可以写作是一系列采样点 s_i 的遮挡贡献 C 的平均值。

$$SSAO_n(p) = \frac{1}{n} \sum_{i=1}^n C(p, s_i)$$

$$C(p, s_i) = V(p, s_i) \max(\cos(s_i - p), 0)$$

其中 s_i 是 p 周围的采样点, $V(p, s_i)$ 是由 s_i 深度解析出来的二值可见性函数。

Mattausch 等人^[85]提出的帧间时间相关性 SSAO 计算方法。这种方法通过每帧指定若干采样点来提升 SSAO 的准确性和绘制质量。该算法使用时间维度上的积分方法。在时间帧 t , 新的 k 个 SSAO 采样点贡献为 C_t :

$$C_t(p) = \frac{1}{k} \sum_{i=j_t(p)+1}^{j_t(p)+k} C(p, s_i)$$

这里 $j_t(p)$ 是已经使用过的采样点数目。新的贡献和之前的累计值混合计算新的 SSAO 值:

$$SSAO_t(p) = \frac{w_{t-1}(p-1)SSAO_{t-1}(p-1) + kC_t(p)}{w_{t-1}(p-1) + k}$$

$$w_t(p) = \min(w_{t-1}(p-1) + k, w_{\max})$$

权重 w_{t-1} 表示有多少采样点已经参与了贡献, 直到达到 $w_{\max}$ 。基于帧间时间相关性的 SSAO 方法能很快达到收敛, $w_{\max}$ 作为预设定值控制着刷新频率, 同时保证旧的贡献值会随着时间的逐步衰减。

上述基于帧间时间连续性的绘制特效集中在阴影, 全局光照等值不会随视点明显变化(视无关)的特效绘制上。采样点值不依赖于视点变化使得关键帧深度图像中的中间信息可以在不同的视点下通过映射进行复用。但是, 基于时间连续性的绘制不适合生成含有生成镜面二阶光线(比如镜面反射光线, 折射光线, 焦散光线)材质的场景。其原因是二阶光线绘制效果是视相关, 当前帧像素所需要的二阶光线信息在关键帧深度图像中没有存储, 或者很难找到对应索引点, 阻碍了帧间时间连续性复用方法在二阶光线特效上的应用。

1.3 目前存在问题

由上述相关研究综述中, 我们可以概括到, 深度图像作为场景近似表示方法因为它易于计算和硬件友好的特点在场景实时绘制下得到广泛的应用。但是, 它的局限性和应用难点限制了它的使用效率和应用范围。深度图像作为场景近似的局限性和应用难点主要体现在以下三个方面:

1. 深度图像采样点有冗余：深度图像能够模拟出视点下的可见场景，但是常规的深度图像由于是在单视点进行采样收集的，所以通常不足以保证能完整采样到绘制特效所需的场景。常用的解决方法是使用一组不同视点下的深度图像增加采样点集合，但是使用这种方法采集的一组深度图像采样点之间有很高的冗余性，限制了采样点集对于完整表示场景的有效性。

2. 深度图像难以按需近似表示被反射场景。反射的物理特性使得当前视点下场景的被反射物体全局性地分布在场景中，而且被反射物体可能只是部分被反射光线捕捉到。因此，深度图像作为反射场景近似表达的主要难点是如何利用反射光线在平滑表面上的连贯性设计深度图像对反射场景的近似表达策略，使得深度图像能按需捕获所有被反射物体的同时又方便反射的计算。

3. 基于时间相关性的深度图像复用算法难以复用高阶光线绘制效果。基于时间相关性的深度图像复用算法大多是基于关键帧中深度图像能捕获场景光照视无关成分，和这些视无关部分几乎不随视点变化而改变的特点。通过复用已经计算好的深度图像上的视无关光照信息来节省光照计算开销。但是由于高阶绘制效果的视相关特性，直接应用基于深度的投影会带来明显的错误，这限制了关键帧深度图像使用时间相关性算法的应用范围。

针对目前深度图像作为场景表达所面临的以上难点问题，通过分析国内外研究现状和未来发展趋势，本文主要围绕对深度图像采样的扩展和应用展开研究。

1.4 论文内容安排

1.4.1 论文内容

为了使深度图像能更有效地对场景进行采样、重建出采样场景的几何特征、按需采样反射绘制所需的被反射场景，并对反射绘制进行基于时间相关性的复用加速，本文的研究思路是：首先针对深度图像组图像之间的冗余采样限制，提出基于非冗余绘制的场景采样方法，旨在提高图像采样像素点的采样有效性，增加深度图像对场景的表达能力；然后针对非冗余绘制带来的高碎片化特性，提出适合于非冗余深度图像的重建方法，并应用此重建方法在软阴影绘制中，完成输出图像像素阴影光线对面光源的可见性计算；在反射绘制中，使用聚类相机生成深度图像按需近似表达反射光线的相交采样场景，从而在实时情况下生成高质量镜面和光泽反射特效；最后为了减少耗时的反射绘制工作开

销，提出反射在不同反射深度图像之间的投影映射关系，借助反射的投影加速反射的绘制，完成基于时间相关性的反射复用方法研究。

图 7 展示了本文关于优化深度图像表示复杂三维场景和相关绘制特效研究的主要流程以及所涉及的相关研究内容。其中，第一个研究内容（即非冗余绘制场景采样方法）是为了提高深度图像相机观察光线采样有效性，第二个研究内容（非冗余深度图像的场景重建和软阴影绘制方法）是为了解决像素阴影光线对面光源可见性的解析。这两个研究内容的处理光线都由相机参数决定，属于一阶光线范畴。第三个研究内容（像素聚类深度图像的反射绘制方法）使用聚类深度图像来近似表达当前屏幕空间采样几何体所反射到的场景，第四个研究内容（基于反射深度图像时间相关性的反射绘制复用方法）使用扩大化定义的深度图像概念来表达当前相机的观察光线采样场景和观察场景所反射的场景表达。第三和第四个研究内容的处理光线是由相机发出的观察光线根据采样场景物体表面决定的反射光线，归为二阶光线范畴。

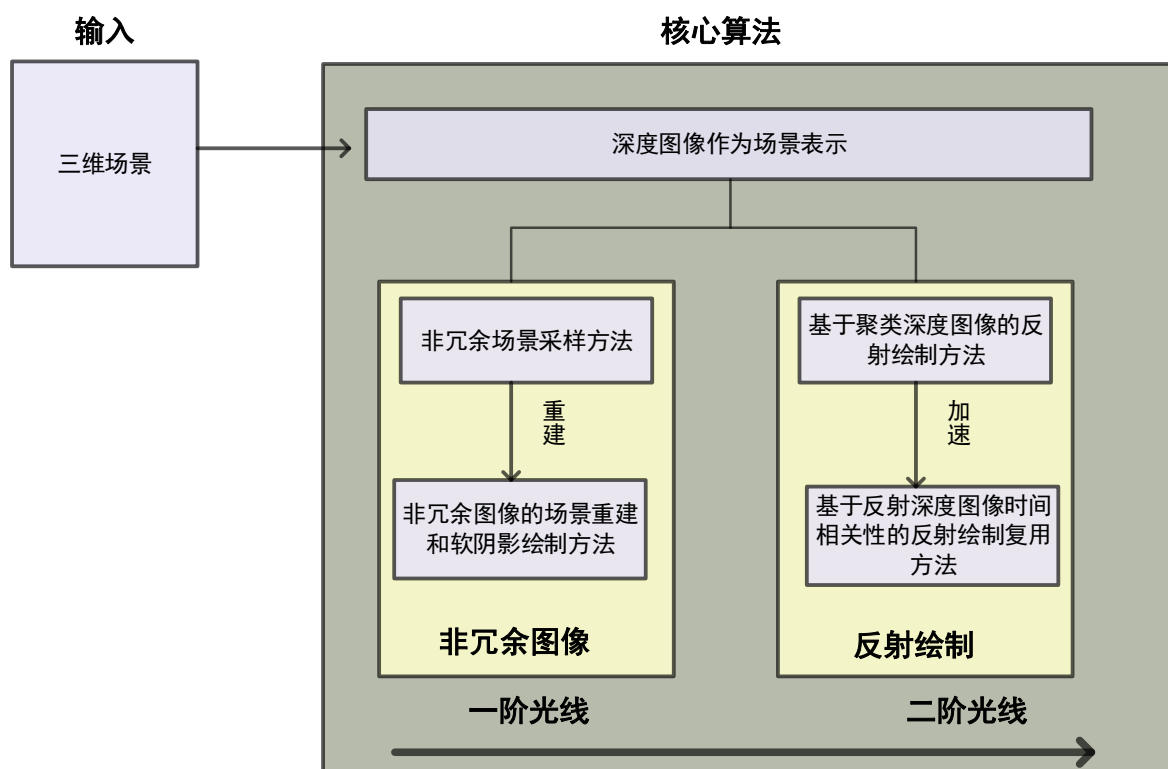


图 7 研究内容整体结构图

本文题目“复杂场景近似表达及绘制方法研究”。本文关注的复杂场景是指拥有多种材质，比如漫反射材质、纹理贴图材质、镜面反射材质、光滑反射材质等，拥有可变更光源，而且场景几何体可以动态变换和扭曲的场景。场景由三角形网格进行建模，数量最高达到数百万三角形。

1.4.2 论文组织与安排

论文共分五章，除了本章绪论以外，其他四章分别介绍了论文在深度图像场景表达方面的工作及其取得的相应成果。

论文的具体组织结构与安排如下：

第一章：绪论。首先介绍了论文的研究背景、意义；然后介绍了本论文研究内容所涉及关键技术的国内外发展现状；最后给出了现有阶段深度图像作为场景表达的限制，并对本文研究内容以及文章的组织结构进行了简要介绍。

第二章：基于非冗余绘制的场景采样方法。首先对多种深度图像对场景进行采样的冗余性问题进行分析；然后给出深度图像采样冗余的不同定义，并引入非冗余绘制算法，展示了非冗余绘制得到的图像和重建结果；最后通过定义三种视无关采样度量方式和一种视相关的采样度量方式，验证了在这四种度量方式下，非冗余图像采样方法相对于传统深度图像采样方法和深度剥离方法在采样性能上的优势。

第三章：适用于非冗余深度图像的场景重建和软阴影绘制方法。首先对非冗余绘制的场景重建所面临问题进行分析；然后对算法整体流程进行概述，并针对算法步骤详细介绍了场景重建算法的特点、参考图像对应的偏移图像的构成、采样点四个角点的模拟生成采样过程；最后给出了重建场景在软阴影绘制中的应用算法，并且验证了非冗余绘制算法，相对于传统深度图像采样方法和深度剥离方法，在软阴影绘制中能重建出的几何场景误差最小，验证方法的有效性。

第四章：基于像素聚类深度图像的反射绘制方法。首先分析目前的研究现状并说明存在的问题；然后对整体算法流程进行了概述，并针对算法步骤详细介绍了屏幕空间聚类的划分规则、不同反射模型下聚类相机深度图像的生成方式、反射光线与深度图像的求交过程以及为支持光泽反射的改动。最后对该方法的反射绘制效果和时间性能上同光线跟踪方法的对比，验证了方法的有效性。

第五章：基于反射深度图像时间相关性的反射绘制复用方法。首先对基于时间相关性关键帧复用算法直接应用在反射绘制时所面临问题进行分析；然后对算法整体流程进行概述，提出反射深度图像这一扩大化的深度图像概念，并针对反射投影算法步骤详细介绍了不同帧之间的像素最佳反射匹配算法、反射中间图像偏移生成方式、输出图像合成步骤，以及关键帧的选取策略。最后对该方法的反射绘制效果和时间性能上同光线跟踪方法和直接投影方法进行对比和分析。

最后，论文总结了所取得的研究成果，给出了有待继续深入研究的问题和方向。

第二章 基于非冗余绘制的场景采样方法

计算机实时绘制方法常利用场景的深度图像作为复杂场景的近似表达。但是,传统的从多视点绘制得到的深度图像组各图像的采样像素之间有着很强的冗余性,这种冗余性限制了这组深度图像对于完整复杂场景的表达能力。为了消除这一限制,本章提出一种非冗余绘制算法。非冗余绘制算法通过检测和避免场景中冗余采样点的绘制,捕获到传统采样方法不能捕获到的非冗余像素采样点,而这些非冗余图像采样点中大部分对中间输出图像的生成有用。本章同时展示了非冗余绘制深度图像绘制采样方法,相比于传统深度图像绘制采样方法或者是深度剥离绘制采样方法,在视相关度量和视无关度量上会取得更好的场景采样效果。

2.1 引言

由于复杂场景的真实感绘制开销大,大多数计算机实时绘制方法需要考虑绘制场景的规模。这样的例子包括大型复杂场景在小型终端显示(如智能手机)和大型场景的复杂特效绘制(如软阴影、反射、环境光遮挡和全局光照的绘制)。虽然研究人员已经在复杂场景的简化上有深入的研究,但是这还是一个尚未完全解决的研究问题。目前的难点包括如何在给定一定的拟合误差情况下控制场景几何体总量、如何支持全动态场景和如何在场景的不同层次细节表达之间的平滑转换。

用图像采样的方法来近似几何场景可以通过图像尺寸控制拟合误差和适用于动态场景的实时绘制性能解决上述难题。包含深度信息的图像是对场景的有力表达。这样的深度图像可以借助计算机硬件绘制单元很快地生成,而且由于它生成大量采用点摊销了场景的绘制采样开销。

但是,传统的深度图像绘制的问题是它的采样范围受限于图像生成时单相机所捕获的场景范围。针对这个问题的一种最直观的解决方法是使用多相机组生成多张场景深度图像增加采样点集。但是,多张图像中的图像像素采样点之间有很高的冗余性。为了消除这种冗余性,一种解决方法是归纳式地与已经绘制图像对比消除冗余采样点;另一种方法是使用深度剥离方法从同一视点绘制采样比前一层采样点更深的场景。这种方法每次绘制上层图像没有捕获过的场景。但是这种方法的局限是它总是从单一视点进行采样,这样限制了采样质量。比如,当视点恰好在场景某一面墙面上,那么无论进行多少

层的深度剥离，这面墙都不会被采样到。

本章节引入一种非冗余绘制离散采样方法，在多相机视角下通过该方法能使深度图像采样到三维场景的非冗余采样点。非冗余绘制采样方法基于一种非常直观的设计思路：一个采样点当且仅当之前没有被采样过才能绘制成像。换言之，当深度图像组中的第二张或者说后续的图像在绘制生成时，场景采样点有且仅有在它没有被前面的已绘制的图像绘制过才会被绘制，这样生成的非冗余绘制图像会有更高的采样有效性。相比于前面描述的深度剥离方法，非冗余绘制算法能从不同的视点下进行非冗余绘制而捕获到原视点捕获不充分的采样点，这样提高了采样有效性，提升了采样质量。

本章验证了相比于对比采样方法如传统深度图像绘制方法和深度剥离绘制采样方法^[50]，非冗余绘制有更高的采样有效性。图 9 展示了非冗余绘制的有效性。传统绘制方法受限于采样像素冗余性，这种冗余性可以由非冗余绘制和深度剥离方法所避免。相比于深度剥离方法，非冗余绘制采样方式能捕获到更多的遮挡像素，而且遮挡像素中有更大的比例是对参考视点的中间区域可见。如图 9 所示，鸟巢场景使用三张深度图像分别按照非冗余绘制采样方式、传统深度图像绘制采样方式和深度剥离采样方式进行采样绘制。图 9 中最上行的数值记录了传统冗余图像的冗余像素数目相对图像像素数目的比例，中间行和下面行的数值分别记录了非冗余绘制和深度剥离两种采样方式下能捕捉到的冗余采样点相对于图像采样点数目的比例以及有用冗余采样点相对于冗余采样点的比例（括号内）。传统采样方式和非冗余绘制方式使用相同的三个视点 V_0, V_1 和 V_2 。深度剥离方法根据定义使用一个视点采样，这个视点是三角形 $V_0V_1V_2$ 的中点，相机设置如图 8 红色点所示。

对于传统绘制方法（图 9 上）， V_1 生成的参考深度图像 I_1 和 V_2 生成的参考深度图像 I_2 中采样点有很高的冗余性，分别对应 70% 和 80% 的图像采样点。对于非冗余绘制方式（图 9 中），生成图像没有冗余采样点。非冗余绘制方式生成的参考图像 I_1 和参考图像 I_2 分别有 62% 和 76% 的采样点是对点 V_0, V_1 和 V_2 的遮挡的，也就是说，这些采样点是传统深度图像采样不到的；在这些新采样的遮挡采样点中，分别有 89% 和 67% 的图像采样点是有用的，像素有用是指该像素对 $V_0V_1V_2$ 定义的三角形的某一中间视点生成输出图像帧所需。对于深度剥离方法所绘制的图像（图 9 下），同样不会生成冗余采样点。但是与非冗余绘制采样方法相比，深度剥离方法捕获到更少的遮挡采样点，而这其中有用采样点的比例也更小。换言之，深度剥离所捕获到的额外遮挡采样点中有更大比例对 $V_0V_1V_2$ 中间区域任意点都不可见。

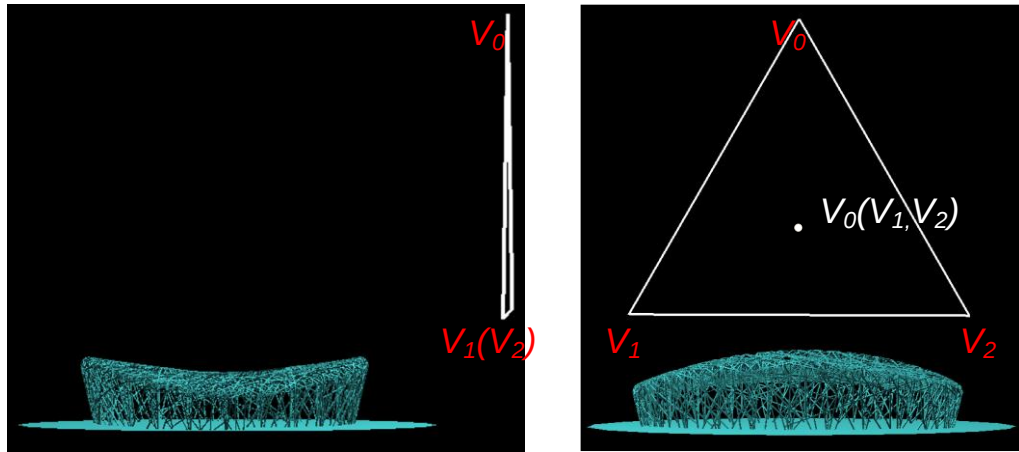


图 8 非冗余绘制采样鸟巢场景视点设置图

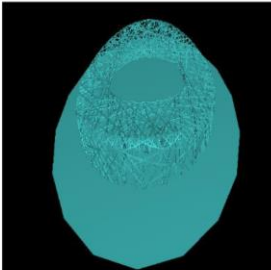
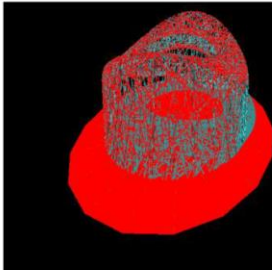
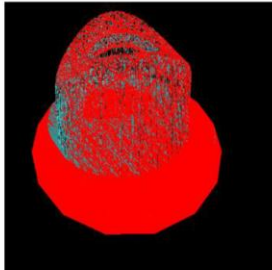
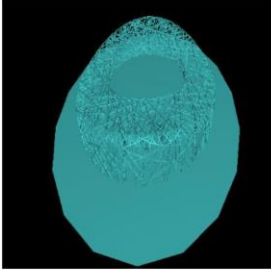
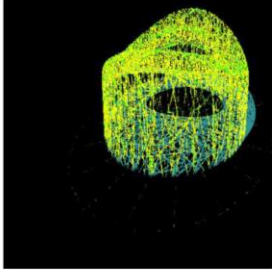
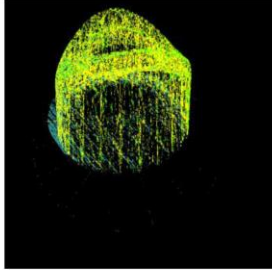
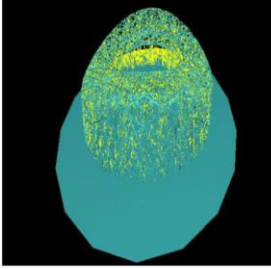
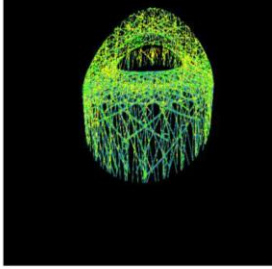
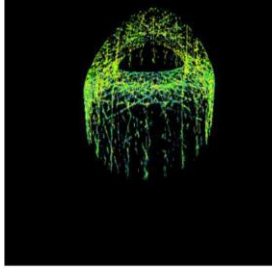
V_0 点生成图像	V_1 点生成图像	V_2 点生成图像
		
0%	70%	80%
传统图像绘制。红色表示冗余采样点。		
		
0% (0%)	62% (89%)	76% (86%)
非冗余绘制。黄色 + 绿色：遮挡采样点。（黄色：有用的遮挡采样点）。		
		
10% (100%)	47% (73%)	52% (71%)
深度剥离。黄色 + 绿色：遮挡采样点。（黄色：有用的遮挡采样点）。		

图 9 鸟巢场景的传统绘制采样，非冗余绘制采样和深度剥离方法的对比图

2.2 非冗余绘制

非冗余绘制方法是一种通过在不同视点场景进行采样生成非冗余图像的方法。本章首先介绍场景冗余性定义，然后介绍非冗余绘制的算法。

2.2.1 冗余定义

给定一个由三角形网格组成的场景 S ，分别从视点 V_0 和 V_1 对场景进行绘制形成两个深度图像 I_0 和 I_1 。给定图像 I_1 里的采样 s_1 ，非冗余绘制采样方法需要验证 s_1 是否已经同已绘制的场景图像 I_0 中的像素采样冗余，本文定义三种冗余类型：

严格冗余：采样 s_1 与 I_0 严格冗余当且仅当 I_0 中有某个采样点 s_0 ，且 s_0 和 s_1 采样到同样三角形上的同一个场景几何点。因为采样几乎不会采样到同一个点，这样的冗余几乎不会发生。这种定义方式仅限于理论，本文不会使用这个定义方式。

可见性冗余：采样点 s_1 与 I_0 可见性冗余当且仅当 s_1 对于 I_0 的绘制视点 V_0 可见。这个定义不考虑 I_0 的分辨率，换言之，如果 I_0 有无限的分辨率， I_0 将肯定会找到与 s_1 严格冗余的采样点 s_0 ，否则受图像采样频率限制， s_0 有很大可能性捕获不到。

像素冗余：采样点 s_1 与 I_0 像素冗余当且仅当 s_1 捕获到的三角形与 s_1 在 I_0 上投影点在 I_0 上索引到的三角形相同。这个冗余定义不需要 I_0 上投影点采样到相同的几何采样点，而仅仅只是需要采样到同一个三角形。

图 10 通过像素可能的不同投影位置来说明三个不同的冗余定义。 $a \sim f$ 所在十字交叉点表示场景采样点 s_1 在 I_0 上可能的投影位置。图中使用不同颜色的三角形来表示场景不同物体的遮挡关系，红色表示遮挡物体，蓝色表示被遮挡物体。在 a, b, c 的举例中，假定 s_1 在 I_1 里采样到三角形 T_0 。如果 s_1 在 I_0 上投影到 a 点，因为 s_1 恰好投影到像素的中心采样点 a 上而被捕获到， s_1 相对于 I_0 属于严格冗余。如果 s_1 投影到 b 上， s_1 对 I_0 中采样点集同时属于可见性冗余和像素冗余，因为 s_1 相对于 I_0 的视点 V_0 可见，而且 b 所在的像素中心点在 I_0 上也正好采样到 s_1 采样的三角形 T_0 。如果 s_1 投影到 c 点，因为在 V_0 和 s_1 直接没有遮挡，所以 s_1 对 I_0 是可见性冗余。但是 s_1 相对于 I_0 并不是像素冗余，包含 c 的像素中心在 I_0 上采样到了另外一个三角形 T_1 。对于 d 而言，假定 s_1 在图像 I_1 里面采样到三角形 T_1 ，在这种情况下， s_1 并不是对 I_0 可见性冗余，因为 d 点被三角形 T_2 遮挡，但是 s_1 对 I_0 是像素冗余的，因为包含 d 的像素点中心在 I_0 上采样了同样的三角形 T_1 。对于 f 而言，假设 s_1 在 I_1 中采样到 T_2 ，这样采样点相对于 I_0 是可见性冗余，

但是并不是像素冗余的。

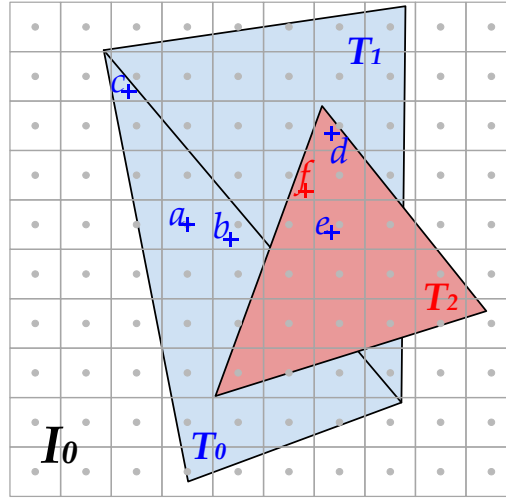


图 10 冗余定义示意图

2.2.2 非冗余绘制定义

给定一个由三角形面片构成的场景 S ，同时给定一组由 n 个小孔相机 PPC_i (i 从 0 到 $n-1$) 构成的相机序列。场景 S 应用非冗余绘制算法（算法 1）生成 n 个非冗余深度图像 I_i 。

算法 1：非冗余绘制算法（Non-redundant Rendering）

输入：场景 S , PPC_i ($i = 0$ to $n-1$)

输出：非冗余深度图像 I_i ($i = 0$ to $n-1$)

for each image I_i ($i = 0$ to $n-1$)

 initialize the z-buffer of I_i to far

for each triangle t in S

 project t with PPC_i to t'

for all pixels p covered by t'

 compute sample s of t at p

if s fails the z-buffer test **then continue**

 //深度检测

for all previously rendered images I_j ($j < i$)

if s is redundant with I_j

 mark s as redundant

break

//如果当前像素与之前图像冗余，舍弃该采样点

if s is not redundant write s in I_i

非冗余深度图像一次绘制一张，每一张图像都是通过绘制整个场景得到。被绘制三角形按照传统 GPU 绘制流水线进行投影和光栅化，然后产生绘制采样点。非冗余深度图像绘制算法当且仅当采样点通过了深度测试且与之前的绘制深度图像判断为非冗余的情况下才绘制采样点，否则舍弃该采样点。这组深度图像的第一个图像，因为没有其他图像做冗余判断对比，是按传统深度图像绘制方法生成的深度图像。接下来的非冗余图像绘制会检测光栅化算法生成的采样点冗余性，舍弃冗余的采样点。冗余的判断按照之前的两种冗余定义进行判断，见算法 2 的可见性冗余判断和算法 3 的像素冗余判断。

算法 2：可见性冗余判断 (VisibilityRedundancyTest)

输入: 采样点 s ，生成采样点 s 的对应的小孔相机 PPC_i ，需要执行冗余判断的深度图像 I_j 与对应视点 V_j ，场景 S

输出: s 与 I_j 的可见冗余性

Unproject s to 3D point a using PPC_i

//投影像素 s 到视点 V_j 下三维点 a

Raytrace $b = V_j a \cap S$

//生成可见性判断光线 $V_j a$ 与场景 S 进行光线求交计算，由最近交点 b 判断可见性

return ($b == a$)

算法 3：像素冗余判断 (PixelRedundancyTest)

输入: 采样点 s ，生成采样点 s 的对应的小孔相机 PPC_i ，生成 s 的场景三角形编号 t ，需要执行冗余判断的深度图像 I_j 与对应视点 V_j

输出: s 与 I_j 的像素冗余性

Reproject s to pixel r in I_j using PPC_i and PPC_j

//投影像素 s 到视点 V_j 下得到投影点 r

let t_r be the id of the triangle sampled by I_j at r

//在 I_j 上找到 r 点存储的三角形编号 t_r

return $t == t_r$

图 11 展示了用非冗余绘制、传统绘制方法和深度剥离方法进行采样的一个简单场景。图 12 展示了使用三种采样方法绘制生成的采样图像和中间视点生成的重建图像 I_R 。非冗余绘制生成两张深度图像 I_0 和 I_1 ， I_0 存储的是一张传统图像采样生成的传统图像。 I_0 捕获到物体 A 和物体 D 的前面（图 12）， I_1 使用非冗余绘制方法采样到物体 B 和物体 C 的前面（图 12 中的第一列）。所有的绘制都启用了背向剔除方法来避免采样到物体的后面。对于传统采样方法， I_0 的采样结果是一样的。 I_1 只采样到物体 B 而没采样到物体 C（图 12 中的第二行）。C 正好对于视点 I_0 和 I_1 不可见，但是它对中间视点 I_R 可见，这样对于中间视点 I_R 就形成了遮挡误差。深度剥离方法在第二层绘制采样到 B 的正面（图 11 右，图 12 的最下一行）。但是 C 对于 I_0 点来讲是双重遮挡的，即 C 被遮挡在 A 和 B 的后面，所以绘制两层深度的图像是不足以采样到 C。如图 12 右所示，用像素冗余定义方式定义的非冗余算法采样场景采样到足够重建出 I_R 视点场景的采样点，并且对场景的还原重建表达结果（见章节 3.2）与原场景相似。如果使用可见性冗余定义方式也会得到相似的结果。

非冗余方法致力于给定视点组下提高采样有效性而不是观察视点的优化，进行图像采样的视点组位置作为参数传递给算法 1。这样的视点组可以是由视点线段的两个端点构成（如图 11），可以是由视点三角形的三个顶点组成（如图 8），还可以是面光源的四个角点组成（见章节 3.3）。

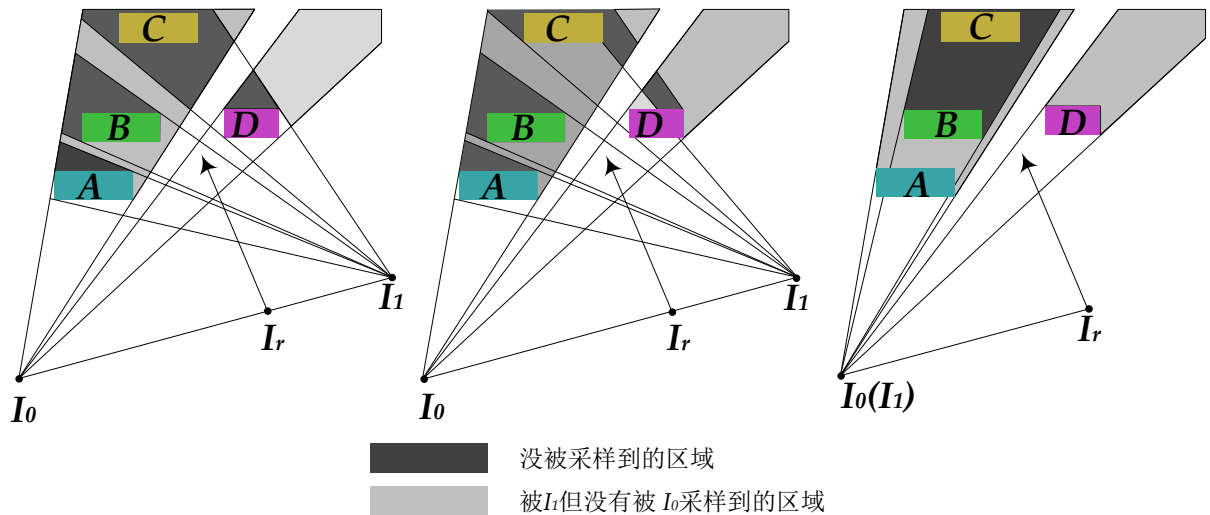


图 11 使用传统采样方法（左），本章方法（中），深度剥离方法（右），采样简单场景 2D 示意图

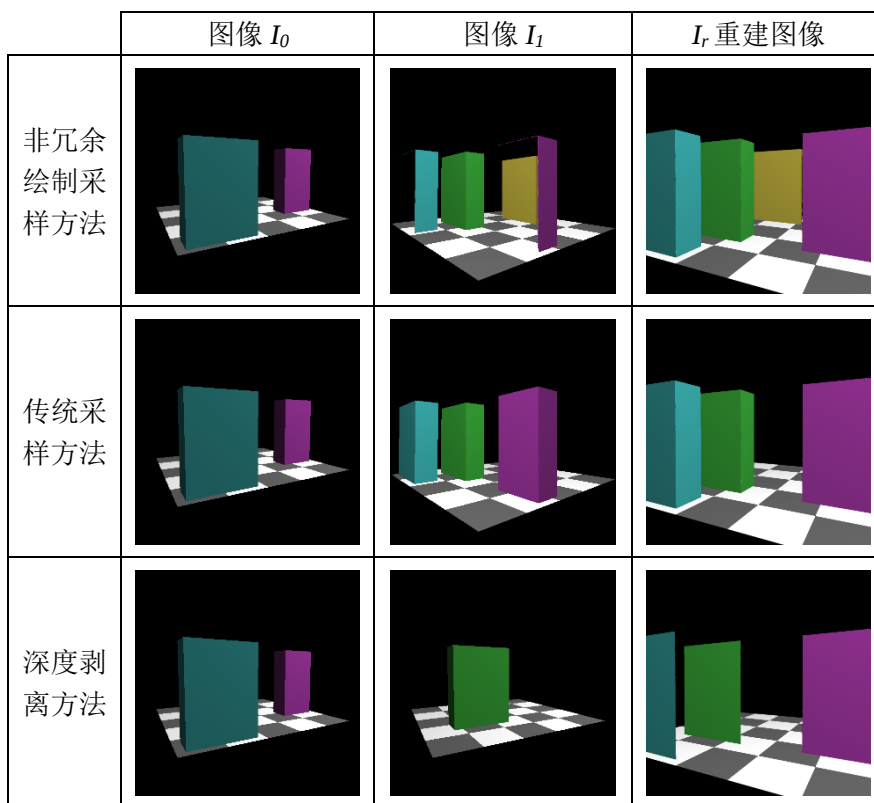


图 12 图 11 中采样的图像和中间视点的重建图像展示图

2.3 实验结果和讨论

本章使用非冗余算法对多个场景进行采样，并与其他采样方法如传统深度图像采样方法和深度剥离方法进行对比。本章节简单描述了采样方法的实现，定义了对比的度量方式及其计算方式，展示和讨论了对比结果并讨论了非冗余绘制采样的局限性。

2.3.1 算法实现

基于像素冗余的非冗余绘制是由片断着色器实现。片断着色器对已绘制的所有图像进行冗余性判断和舍弃冗余像素。三角形编号是由几何着色器生成然后作为参数传递给片断着色器。已绘制的图像通过纹理的方式传递给片断着色器，每个纹素（texel）使用多绘制对象技术记录直接光照、深度和采样三角形编号信息。冗余检测按照算法 3 实现。片断着色器的输出包含颜色、深度信息和片断采样三角形编号的像素，这样形成的图像可以作为片断着色器的纹理传入参数方便后续非冗余图像的绘制生成。

基于可见性的非冗余绘制是按照算法 2 由光线跟踪方法实现。本章光线跟踪使用 NVIDIA 的 Optix 光线跟踪框架^[86]。该光线跟踪框架用层级包围体（bounding volume hierarchy, BVH）作为加速结构。

深度剥离算法的实现和基于像素冗余的非冗余绘制实现近似，它们之间的区别是深度剥离算法是在同一视点下绘制非冗余图像组。

2.3.2 采样有效性度量

给定由三角形建模的场景 S , n 个参考采样图像 I_i (i 从 0 到 $n-1$) 以及对应的视点 PPC_i , 本章给出三种视无关和一种视相关度量方式来量化采样质量指标。像素冗余和可见性冗余定义的非冗余绘制都可以使用这些度量方式。

视无关度量

1. 定义冗余采样点数量为当前参考图像 I_i 中与之前任意参考图像 I_j ($j < i$) 出现冗余的像素数量。根据之前非冗余图像和深度剥离方法定义，在这两类生成的图像中冗余采样点数量为 0，这个度量方式主要是度量传统采样图像采样点集的冗余性。

2. 定义遮挡采样点数量为出现在参考采样图像中但是没有出现在传统采样图像中的采样点。对于一张从视点 PPC_i 非冗余绘制的图像，遮挡像素采样点是指没有被在 PPC_j (j 从 0 到 $i-1$) 点传统绘制图像所采样捕获的采样点。对于深度剥离方法来说，遮挡像素采样是指没有被任何其他传统采样图像 I_i (i 从 0 到 $n-1$) 所捕获的采样点。对于传统采样方法，遮挡像素数量按照定义是 0。遮挡像素度量非冗余绘制和深度剥离方法采样额外遮挡采样点的能力。

3. 定义有用像素数量为参考图像的遮挡像素中为至少一张中间视点输出图像帧所需的像素数目。中间视点输出图像帧是指由参考视点 PPC_i (i 从 0 到 $n-1$) 内插值出的视点绘制的图像帧。比如当 $n = 2$ 时，中间视点输出图像帧的视点在由 PPC_0 到 PPC_1 定义的线段上。当 $n = 3$ 时，输出图像帧的视点是在由视点 PPC_0 , PPC_1 和 PPC_2 定义的三角面片上。这个度量方式是通过从中间视点绘制大量输出图像（比如 1000 帧中间输出图像）然后统计参考图像中为这些中间输出帧所需的遮挡像素数量所得。一个像素采样点为输出图像所需是指这个像素对输出图像冗余，像素采样点对输出图像冗余意味着该像素采样点对输出图像的生成有贡献。这个度量方式对非冗余绘制和深度剥离方法带来的额外遮挡采样点有效性进行量化区分。有用的额外遮挡采样点是指能帮助模拟生成场景对 n 个参考视点定义的区域可见性判断的采样点。有用的采样点对中间采样点可见，也就是说，这些采样点对中间视点的重建有贡献。没用的采样点是指那些对中间视点完全不可见的采样点，而且由于这类采样点不可见，反而降低了这组采样点绘制时遮挡剔除的有效性。

视相关度量

4. 输出图像中的 *可得* 采样点是指输出图像中能在至少一个参考相机生成图像上找到冗余采样点的采样点。度量手段 3 是为了记录参考图像采样点上对于中间视点的有用像素数量，而度量手段 4 则是记录对于给定一张中间输出图像帧中有多少采样点能在参考图像中找到。这个度量方式通过在中间视点下绘制场景得到中间输出图像采样点集然后再去检查每个采样点对参考图像的冗余性。如果一个采样点对于参考图像冗余那么这个采样点即可算作是对该参考图像可得采样点。一个对于任何参考图像都不算是可得采样点的输出图像采样点将会产生遮挡误差，意味着该采样点在参考图像采样点集合中找不到索引。视相关度量方式是度量中间输出图像中有多少像素是因为参考图像缺失采样点而造成遮挡误差。

2.3.3 非冗余绘制采样质量

本章使用上述的四种度量方式来量化比较非冗余绘制方式、传统图像采样方式和深度剥离方法。

表 2 给出使用三种视无关度量在鸟巢、树、草和城市模型上的用基于像素冗余的非冗余定义的度量结果。场景给定相机视点组设置统一按照图 8 红色点表示。有用采样点是以相对于遮挡采样点的百分比形式给出。场景由三个深度图像采样。传统采样方式在 V_1 点生成图像和 V_2 点生成图像上有很多的冗余采样点。非冗余绘制和深度剥离方法没有冗余采样点。深度图像采样的第一张图像也会遮挡采样点，这是因为 V_0 点生成图像是在位于传统采样方式和非冗余采样方式的三个图像生成点定义的三角形中点（图 8 白色点）绘制的，这个视点下会采样到在任意视点下图像都不被采样的几何场景。

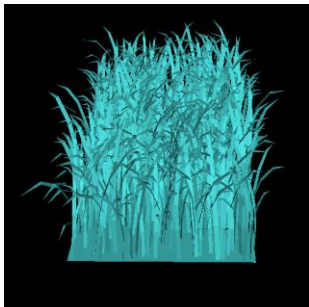
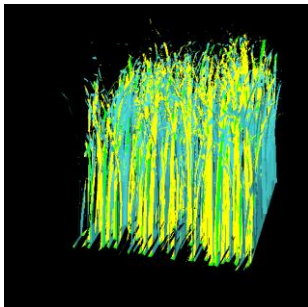
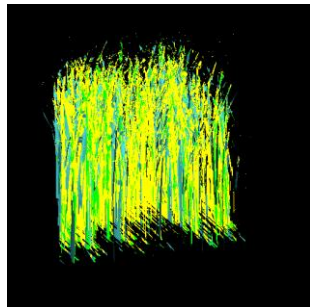
表 2 视无关度量表（像素冗余定义方式）

模型	V_0 点生成图像	V_1 点生成图像	V_2 点生成图像
传统采样的冗余采样点			
鸟巢	0%	70%	80%
草	0%	54%	69%
城市	0%	82%	84%
树	0%	23%	35%
非冗余绘制中遮挡采样点和其中（有用）采样点所占遮挡采样点的百分比			
鸟巢	0% (0%)	62% (89%)	76% (86%)

草	0% (0%)	56% (85%)	75% (80%)
城市	0% (0%)	73% (40%)	80% (23%)
树	0 % (0%)	84% (99%)	88% (98%)
深度剥离方法中遮挡采样点和其中（有用）采样点所占遮挡采样点的百分比			
鸟巢	10% (99%)	46% (2%)	52% (71%)
草	15% (99%)	51% (4%)	71% (79%)
城市	1% (100%)	78% (7%)	88% (8%)
树	44% (99%)	63% (98%)	72% (97%)

图 9 可视化了鸟巢场景（6.7 万三角面片）中的遮挡采样点和有用像素，传统绘制采样方法受制于冗余采样点，冗余采样点可以由非冗余绘制采样方式和深度剥离方法解决。相比于深度剥离方式，非冗余绘制方式能采样捕获到更多的遮挡采样点，而且其中有更多的采样点对三个参考视点之间的输出视点有用。表 3 可视化了草场景（5.6 万面片），城市场景（5 万面片）和树场景（11.3 万面片）的遮挡采样点和有用采样点，图表中对应图像下逗号前和逗号后的数值分别以百分比形式给出遮挡采样点相对场景像素采样点占比和有用采样点相对遮挡像素点的占比。非冗余绘制采样方法相比于深度剥离方法绘制捕获了更多的遮挡采样点，并且有着更多的有效遮挡采样点。唯一的例外是在城市场景中深度剥离方法会捕获到稍微多一些的遮挡采样点。但是，这些采样点中只有很小比例是有用采样点，所以即使是在这个场景下，非冗余绘制采样方法下相比于深度采样方法还是能捕获更多的有用采样点。

表 3 遮挡采样点（绿色+黄色）和有用的遮挡采样点（黄色）可视化图表

场景	参考图像 0	参考图像 1	参考图像 2
草	 0%, 0%	 56%, 89%	 75%, 79%

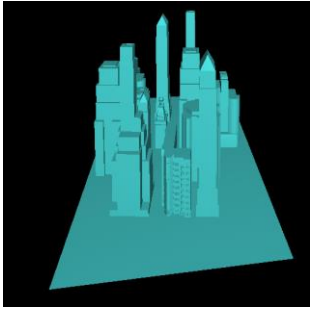
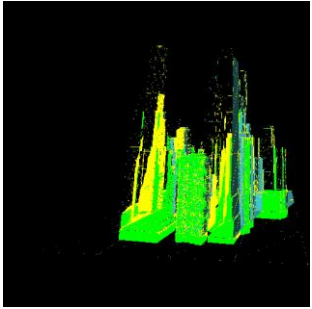
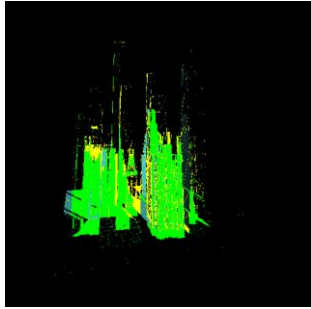
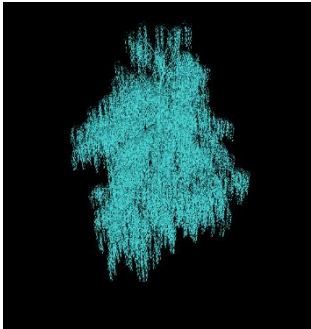
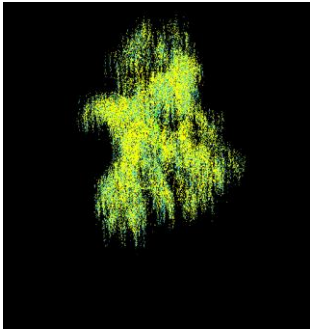
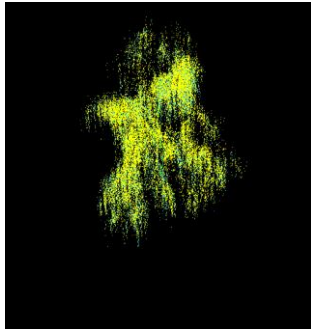
城市			
	0%, 0%	73%, 40%	80%, 24%
树			
	0%, 0%	84%, 99%	88%, 98%

表 4 给出了由中间视点构成的漫游路径下绘制 1000 帧输出图像时每帧平均可得采样点数目和整个路径中最小可得采样点数目所占百分比。相比于传统采样方式和深度剥离方式，非冗余绘制中的平均和最少可得采样点数目占比更高，说明有更多的输出图像像素可以由非冗余绘制图像贡献生成。

表 4 视相关度量表（像素冗余定义方式）

模型	传统采样方法		非冗余绘制采样方法		深度剥离	
	平均	最少	平均	最少	平均	最少
鸟巢	90%	89%	93%	92%	92%	89%
草	86%	81%	96%	91%	93%	88%
城市	97%	96%	98%	97%	96%	91%
树	55%	53%	59%	57%	55%	48%

表 5 和表 6 展示了用可见性冗余定义的非冗余绘制方式的对比结果。同像素定义统计结果一样，在视相关和视无关度量尺度上，非冗余绘制采样方式的表现比传统绘制方法和深度剥离方法更好。当绘制图像的分辨率提升时，像素冗余方式会收敛为可见性冗余。图 13，图 14，图 15 分别给出了可见性冗余定义下当绘制图像分辨率逐步增加时的像素冗余定义的冗余采样点占比，遮挡采样点占比和有用遮挡采样点占比变化情况。

表 5 视无关度量表(可见性冗余定义方式)

模型	V_0 点生成图像	V_1 点生成图像	V_2 点生成图像
传统采样的冗余采样点[%]			
鸟巢	0	80	87
草	0	58	74
城市	0	87	87
树	0	49	67
非冗余绘制中遮挡采样点和其中(有用)采样点所占遮挡采样点的百分比[%]			
鸟巢	0 (0)	55(66)	66(61)
草	0 (0)	48(80)	73(85)
城市	0 (0)	74(29)	80(9)
树	0 (0)	36(92)	54(96)
深度剥离方法中遮挡采样点和其中(有用)采样点所占遮挡采样点的百分比[%]			
鸟巢	10,99	39(38)	45(55)
草	15,99	48(81)	69(76)
城市	1,100	75(10)	88(6)
树	44,99	30(95)	45(94)

表 6 视相关度量表(可见性冗余定义方式)

模型	传统采样方法[%]		非冗余绘制采样方法 [%]		深度剥离[%]	
	平均	最少	平均	最少	平均	最少
鸟巢	95.25	94.46	99.84	99.73	99.50	98.82
草	86.61	83.61	99.72	99.35	99.57	98.15
城市	98.91	97.64	99.82	99.32	99.37	94.23
树	85.36	83.63	98.61	97.78	97.65	93.94

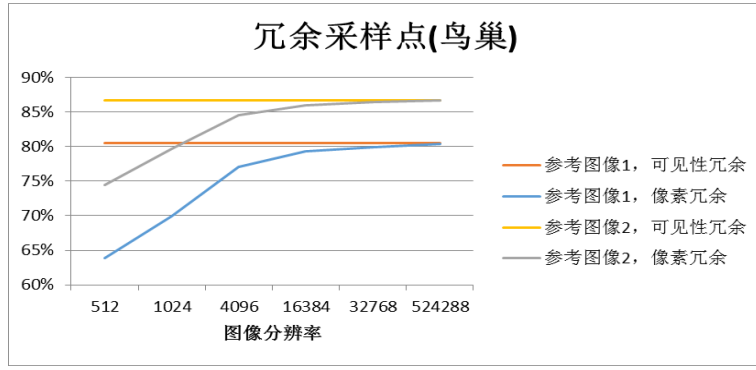


图 13 冗余采样点占比随图像分辨率变化示意图

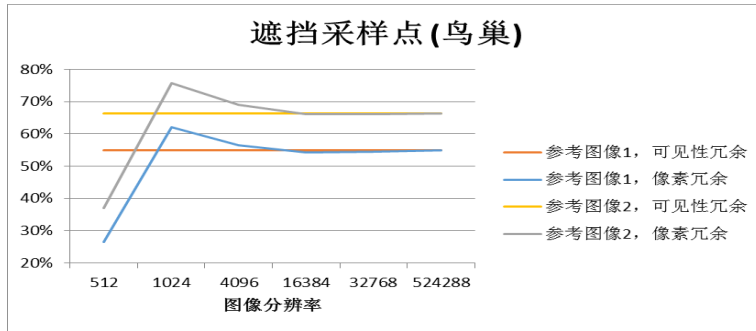


图 14 遮挡采样点占比随图像分辨率变化示意图

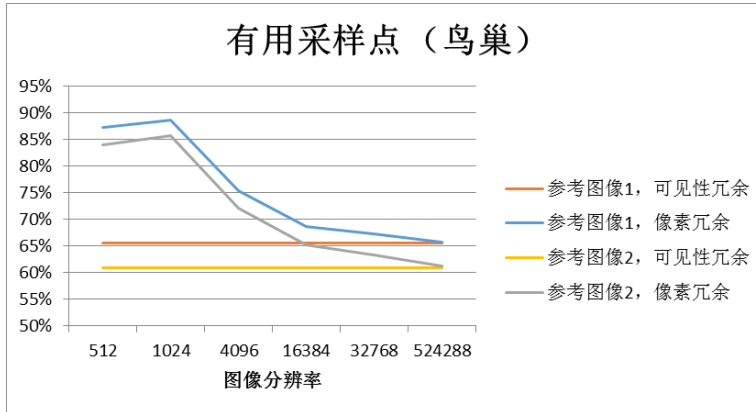


图 15 有用采样点占比随图像分辨率变化示意图

非冗余绘制时间效率

表 7 给出了作用在四个场景的三种离散化采样的时间开销。本章节所做的性能分析是基于 3.4GHz Intel(R) Core(TM) i7-2600 CPU, 4 GB 内存和 NVIDIA GeForce GTX 570 显卡个人电脑平台。测试的时间分别是测试绘制三张传统深度图像, 绘制三张非冗余深度图像和三张深度剥离图像的时间。测试使用的是像素冗余的非冗余图像绘制定义方式, 生成的图像分辨率是 1024×1024 , 非冗余绘制在较简单场景上的效

率接近于传统绘制和深度剥离方式，在最复杂的树场景上非冗余绘制的效率有明显的落差，这是由于生成三角形编号的几何着色器成为了整个非冗余绘制的瓶颈。

表 7 三种绘制采样方式的时间效率

模型	传统采样方法[毫秒]	非冗余绘制采样方法[毫秒]	深度剥离[毫秒]
鸟巢	1.8	3.6	3.3
草	1.7	3.0	3.2
城市	1.6	3.3	2.7
树	2.1	6.4	6.5

2.3.4 算法讨论和局限性分析

虽然非冗余绘制图像能捕获到更多的对于指定区域所需的采样点，但是这不能保证所有的所需采样点都被捕获到。同样，这些新捕获到的采样点也不能保证都对场景应用绘制效果有用。本章的实验展示了大部分新采样点对于相机点构成的中间区域至少一个视点可见，这样大部分采样点是有用的。相比于传统采样图像，非冗余采样图像使用可能有用的其他几何体采样点去代替没用的冗余采样点，所以这是一种对于可见性问题的贪心法解决方案。这有利于使用更少的图像来达到足够的场景采样，或者是使用固定数量的图像达到更好的采样效果。但是决定多少张图像能解决当前所有可见性问题和怎样安排采样相机视点采样用户感兴趣区域是一个 NP 完全问题。所以在动态场景绘制这种不能依赖于离线预计算的绘制条件下还是得使用非冗余绘制这种贪心方法来模拟解决这一问题。

非冗余算法的另一个局限，正如前面所说，是在绘制的时候必须要进行冗余性判断，这会带来时间性能的损失。但是，非冗余绘制方式的时间性能跟深度剥离方式相差不大，而且它能在多个视点进行采样，可以带来更好的采样质量。

未来的工作方向包括考虑通过改变硬件绘制单元的结构来为非冗余绘制采样提供更好的支持，以及在可见性计算，远程可视化或反射绘制等其他程序应用中对非冗余绘制进行评估。

2.4 小结

本章针对多视点绘制得到的场景采样深度图像组各图像之间采样点存在采样点冗

余限制了图像对场景的表达能力这一问题，提出了一种新的非冗余绘制算法。本章给出了三种采样点冗余的定义方式和非冗余绘制算法的实现方法。非冗余绘制算法当且仅当采样点没有被捕获采样过才能绘制成像。非冗余绘制算法不仅能避免冗余像素点的生成还能捕获传统采样方式不能捕获的非冗余采样点，而这些新捕获的非冗余采样点大部分对位于参考视点中间区域的视点生成输出图像有用。本章定义了三种视无关度量方式和一种视相关度量方式对场景采样方法进行对比评估。实验结果表明，传统绘制方法采样点有很高的冗余性，深度剥离采样方法受限于单视点的视域范围，而非冗余采样绘制方式相对于传统深度图像采样方法和深度剥离方法能更有效地捕获到更多的对于指定区域所需的有用采样点，提高了对场景的采样有效性，增加了对完整场景的表达能力。非冗余采样绘制的采样优势表示在，相比于传统深度图像能采样到传统深度图像采样不到的遮挡采样点，而其中大部分采样点都是对中间输出图像有用的。相比与深度剥离方法，非冗余绘制算法能摆脱单视点的限制而捕获更多的遮挡采样点，而且有用的采样点比例更大。

第三章 适用于非冗余深度图像的场景重建和软阴影绘制方法

非冗余绘制图像相比于传统深度图像，在图像上会有更多的碎片。本算法针对这种非冗余绘制的碎片化特性，提出适合非冗余绘制图像的几何重建方法。本章将非冗余绘制算法及其几何重建方法应用到软阴影的绘制中，并且验证了在此应用场景下，相对于其他场景采样方法如传统深度图像绘制采样方法或者是深度剥离绘制采样方法，非冗余深度图像方法有更好的场景重建效果和软阴影绘制效果。

3.1 引言

大部分基于离散化的场景采样方法在场景绘制应用中需要相应的重建算法进行场景重建。比如，场景远程可视化应用需要在主机端得到足够的采样信息，然后转化为方便客户端进行新视点下绘制的轻量级场景，再传输给客户端进行输出图像绘制。在软阴影绘制中，需要为每个光源采样点生成从场景的离散化采样点中重建的场景近似阴影图。虽然基于点的绘制技术和从点云中重建场景的技术在计算机图形学研究里面已经有很长的历史，但是非冗余性图像的高碎片化绘制特性还是给场景重建工作带来了困难。这是因为在非冗余图像中原有场景中的冗余像素生成位置捕获了生成非冗余像素，给采样的场景增加了深度不连续性，造成非冗余性图像绘制图像会有很多绘制碎片（见图 9 及表 3 的中间列和右列），在这种不连续的碎片化图像上直接使用现有的忽略不连续性边缘处几何体的场景重建方法效果不佳。本章一种新的提出适合于处理这种非冗余图像碎片像素的重建方法。本章的方法能够在场景深度不连续性边缘物体和碎片化物体上提供更好地场景表达结果，本章的场景重建结果是密闭的，而且重建结果不会过度绘制。

本章展示了非冗余绘制图像场景重建的方法，并且在软阴影应用中使用了场景重建结果。图 16 中，第一列的图像是使用原始场景生成的软阴影效果图，软阴影由阴影图方法绘制而成。阴影图的生成是面光源被离散成 16×16 的光源采样矩阵，对于每一个光源采样都生成一张阴影图，然后对每一张阴影图的阴影判断结果积分所得。第一列原始场景生成阴影图是采样重建算法生成阴影效果图比对的基准。第二列展示了由非冗余绘制图像重建场景的阴影图生成软阴影绘制图像，其阴影图的生成首先是在面光源的四个角点绘制非冗余绘制图像，然后再用非冗余绘制图像的场景重建结果生成每个光源采样点下场景近似阴影图。第三列展示了用传统绘制方法生成图像近似重建出的阴影绘制

图像。第四列展示了由面光源中点进行四层深度剥离绘制采样图像近似重建场景生成的阴影绘制图像。本章提出的非冗余绘制算法重建场景能够更好地模拟光源采样阴影图，从而生成误差更小的软阴影绘制图像。在图 16 误差可视化示意图中，绿色表示过亮，红色表示过暗。

场景离散采样方法，例如前面一章提出的非冗余绘制采样方法，目的是捕获能满足应用需求的采样点集合，而场景重建方法的目的是根据捕获采样点重建场景。这两种方法可以通过配合使用，通过采样获取的场景近似表达，一定的几何体误差之内重建场景，减少场景面片表示数量来加速图形运算。

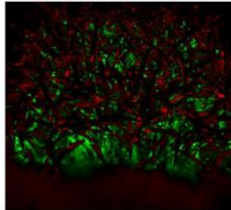
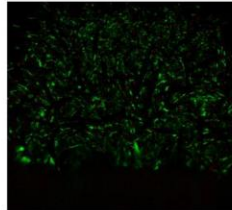
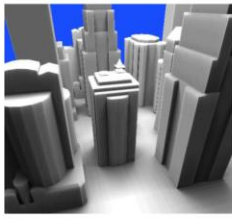
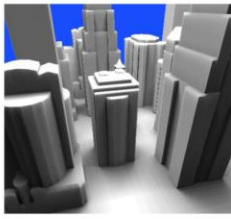
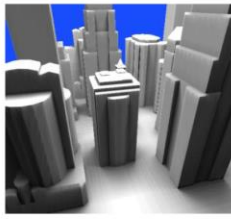
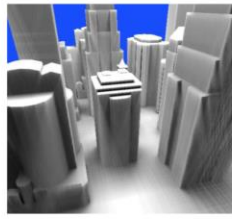
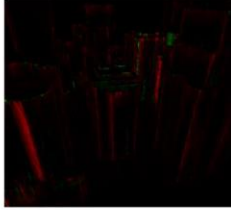
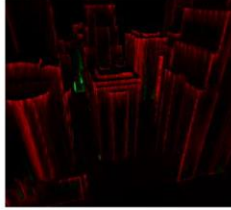
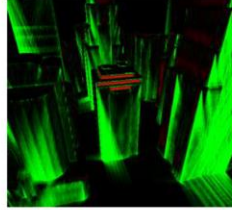
几何体 (真实值)	非冗余渲染采样	常规图像采样方法	深度剔除方法
			
	 强度误差: 1.71	 强度误差: 5.15	 强度误差: 2.55
			
	 强度误差: 1.60	 强度误差: 3.46	 强度误差: 11.36

图 16 使用三种采样方法和原始图像生成的软阴影绘制图

3.2 非冗余绘制的场景重建

大部分的场景离散采样方法需要场景重建方法来重构获得的采样信息。例如，在远

程可视化软件中，客户端必须由服务器传输来的采样点重建出当前输出图像。在阴影绘制中，阴影光线的可见性判断也是需要采样阴影图重建出的场景来执行光线求交遮挡计算。

几何体重建方法大体有两种方法，有相邻采样点连接信息的网格重建法和没有连接信息的孤立点重建法。网格重建法利用了深度图像中天生的采样点连接信息：四个相邻像素点构成的矩形区域可以由两个三角形连接重建，除非相邻像素被深度不连续缝隙分割开。随着 GPU 运算能力的大幅增加，由四个采样点构建出两个三角形的方法可以非常有效地完成。不考虑连接信息的孤立点重建方法是出于忽略采样点之间的邻接性发展起来的。这种方法的困难之处在于需要很精确地估计出采样点在输出图像的影响区域从而能在避免相邻像素重建区域之间出现空洞 (holes) 的同时避免超过影响区域边界的过度绘制。

非冗余绘制算法在消除采样点冗余的同时带来生成图片中物体成像碎片化。这是因为非冗余绘制算法中冗余采样点的剔除和非冗余采样点的绘制这两项操作导致了物体的连续几何表面片元被其他物体采样点替换，所以非冗余绘制图像会比传统深度采样绘制图像有更多的深度不连续性采样点。换句话说，非冗余绘制图像上是传统采样图像中的冗余部分被置换为其他非冗余部分生成的图像，这样每一次置换操作都会造成了深度不连续性。

这种非冗余绘制的碎片化特性给场景重建造成困难。网格重建法在重建几何场景时会在深度不连续性边界制造半个像素宽度的空隙。一个碎片化采样点，因为它没有任何相邻的几何信息可以考虑连接信息，将会被这种方法舍弃，从而不会对场景重建表示产生任何贡献。在碎片化像素较多的图像，比如非冗余图像中，这种重建方法带来的拟合误差将会很严重。点重建方法只提供孤立点的粗略拟合，非冗余绘制图像只会加剧点重建拟合方法的误差。

本章提出一种点重建和网格重建的混合方法，这种方法既不会带来过度绘制又能重建出密闭的几何体来模拟表示原始场景，而且这种方法高度适合严重碎片化非冗余绘制图像。同点重建类似，这种混合方法为每个像素采样点进行面片重建；同网格重建方法类似，不同像素的绘制片元通过共享顶点的表示方法连接起来。适用于非冗余绘制图像的场景重建算法如算法 4 所述。

算法 4：非冗余图像混合重建方法

输入: 非冗余图像 I , 生成图像视点 e

输出: I 所采用的场景的 3D 三角形网格 M

Render non-redundant image H with the same view and same resolution as I , but offset half a pixel in both directions

//绘制非冗余图像 I 的偏移图像 H

For each pixel p in I storing a 3D sample point s with normal n

//在非冗余像素 p 点建立拟合平面

For each corner c of p that is missing from H

//对于每个非冗余像素采样点, 重建像素角点

Define ray r from e through c

Define plane p through s with normal n

Approximate c as intersection of r with p

//在缺失有效角点的位置模拟角点生成

Generate triangles $c_0c_1c_2, c_2c_3c_0$ and add them to M

//生成两个重建三角形

非冗余绘制图像的重建方法示意图如图 17 所示。两个椭圆由图像 I 采样。 I 中像素中心采样点由黑色实心点表示。传统网格重建方法（黑色虚线表示）只能给大椭圆重建出四个三角形，这大大地缩小模拟场景几何表达范围。大椭圆大部分几何体都没有得到重建表示，小椭圆也由于太小只有一个孤立的采样点和没有邻接信息被传统网格重建方法忽略。本章引入偏移图像 H 概念来记录图像 I 中像素的角点信息。

偏移图像 H 中的像素中心采样点由绿色实心点显示。偏移像素中心采样点对应 I 中像素的四个角点。本章提出的混合重建方法把 I 中的每个采样点转换为一个由两个三角形面片建模的四边形。采样点 S_0 在 H 中有四个有效角点，即 C_0, C_1, C_2, C_3 。有效角点是指和中心采样点没有被深度不连续性边缘分割开的角点。采样点 S_1 使用与 S_0 共有的角点 C_0 和 C_1 进行建模，这样使得重建出来的几何体密封而且不会被过度绘制。采样点 S_2 在 H 图像采样中缺失一个角，但是可以由与 S_2 的法线垂直的拟合平面模拟生成。 A_0 和 A_1 是采样点 S_3 和 S_4 根据各自独立的位置和法线定义的不同拟合平面与角点对应的相机采样光线相交生成的不同角点，他们共享同一个坐标信息。采样点 S_5 是个完全的孤立点，虽然它在 H 中没有有效的角点。但是，这个采样点还是通过拟合出四个角点构建一

个四边形完成它的重建贡献。

本章的混合重建方法能提供场景几何体很好的近似表达：本章的重建方法对于连续物体表面能重建出密闭而且非冗余的三角网格。传统网格重建方法和本章的混合重建方法由于都是采用每个采样点生成两个三角面片的方法，所以他们在开销上是相似的。他们之间的不同在于混合重建方法需要一次额外的绘制来生成记录采样点角点的偏移图像。跟传统重建方法一样，对于只是部分覆盖像素但没有覆盖到像素采样点中心的情况，例如图 17 中的像素 P ，两种方法都不会重建。本章的混合重建方法过度模拟了只是覆盖了像素中心采样点这种情况的像素，比如像素 S_3 , S_4 , S_5 。而传统的重建方法忽略了这部分的几何体的表达。所以本章方法在诸如软阴影绘制的应用上有消除拟合误差的优势。

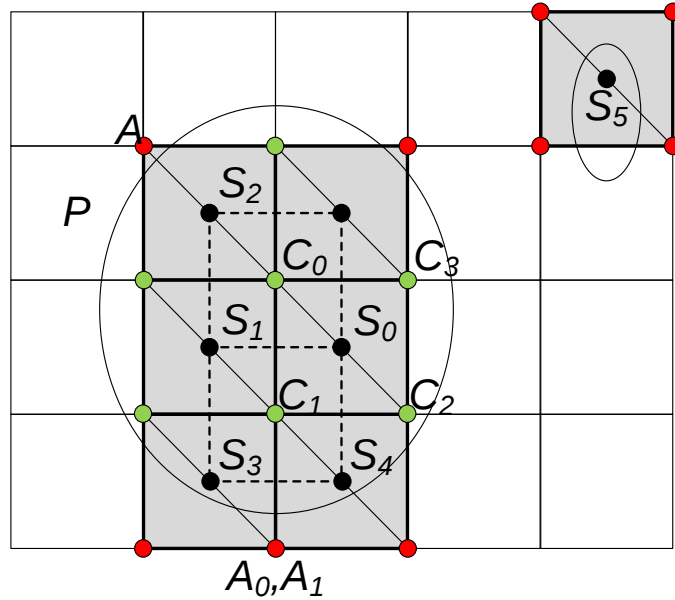


图 17 非冗余图像的混合重建方法

3.3 在软阴影上的应用

软阴影^[81,83]绘制的难点在于估计输出图像的像素采样点和面光源的可见性判断。很多方法通过离散面光源为很多点光源，把对面光源的可见性判断简化为对点的可见性判断。高质量的软阴影绘制需要将面光源按照高分辨率（例如 16×16 , 32×32 ）进行点光源离散化，需要执行大量场景和像素对点光源连线的可见性判断检测。因为可见性判断可以在一个较小的开销内与场景的离散化近似表达而不是原始场景计算求得，所以通过对原始场景的重建离散采样可以加速可见性判断计算。非冗余绘制方式在场景的离散

化采样表达方式比传统深度图像采样有优势，而这种优势也会通过场景混合重建算法转化到软阴影绘制上：首先，非冗余绘制能通过非冗余绘制图像混合重建方法解决传统重建方法由于重建出的场景几何体过少带来的阴影绘制中的漏光现象，其次，非冗余绘制图像混合重建方法不会重建冗余场景，避免了光线和重复场景进行多次求交操作。

本章使用算法 5 进行软阴影绘制。

算法 5：软阴影绘制 (SoftShadowRendering)

输入: 矩形面光源 L , 场景 S ,

输出: 每个采样点记录对面光源可见性的图像 I

1. Render non-redundant depth images $NRDI$ of S from each of the four corners of L

// 在光源角点处生成非冗余图像组

2. For each $NRDI_i$ compute the scene reconstruction mesh M_i

// 对每个非冗余绘制图像混合重建方法重建三角网格

3. For each light sample s

- Initialize approximate shadow map SM

- // 生成点光源对应阴影图 SM

- For each M_i

 Render M_i from viewpoint s onto SM

- // 由阴影图 SM 完成输出图像像素阴影值判断

- For each sample o of I

 Estimate visibility from s to o using SM

场景首先在矩形面光源的四个角点进行非冗余绘制场景采样（步骤 1）。对场景的采样结果使用混合重建算法转换成对原始场景的三角形网格近似表达（步骤 2）。生成的场景近似表达用作生成每个光源采样点的阴影图，最后使用阴影图算法完成输出图像像素对面光源可见性判断（步骤 3）。

3.4 试验结果和讨论

3.4.1 重建质量

本章测试和对比了使用混合重建方法应用在非冗余图像上的重建效果。图 18 展示了从面光源中心视点下使用非冗余图像和混合重建方法生成重建图像和原始几何体生成图像对比。我们的方法能够重建出大部分原始几何体，但是在重建出来的场景在左上

深绿建筑，右上浅绿建筑，左下的红色建筑表面上有缺失。

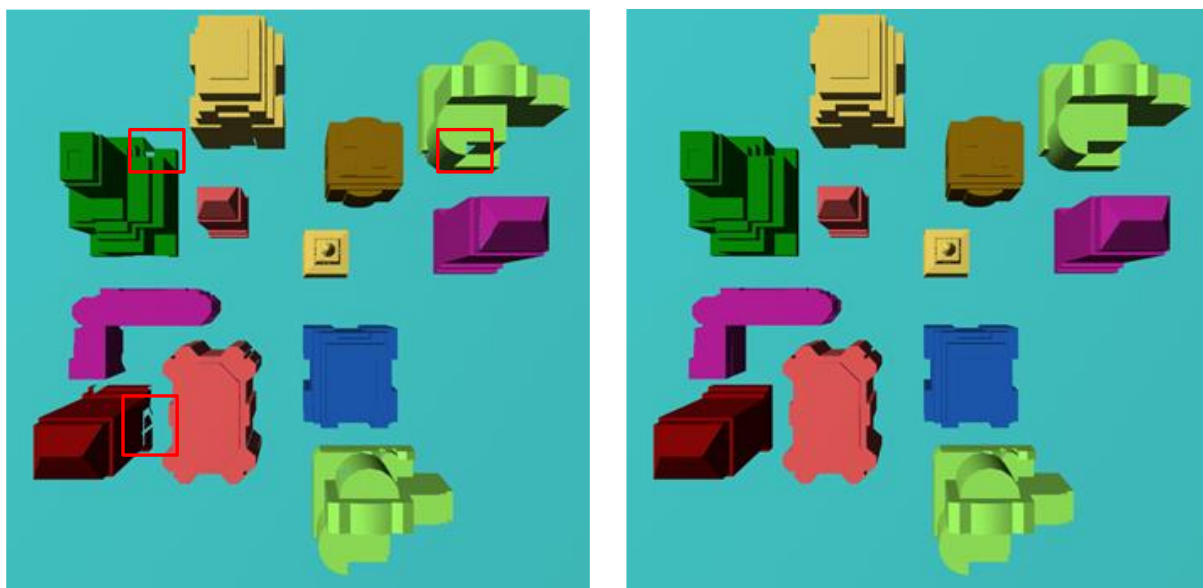


图 18 视点在面光源中心下本章方法重建结果（左）和用真实场景绘制的图像（右）

在软阴影绘制中，因为可见性光线从面光源发出，所以混合场景重建算法不可能被面光源以外的视点调用。但是对于其他的应用程序如远程可视化，可能需要在给定参考视点围成区域外重建出场景。因为我们的采样方法能用非冗余采样点代替冗余采样点，提高了采样的效率，所以我们的方法也同样能提高了重建的鲁棒性。图 19 展示了当视点在参考视点区域之外的重建情况（图中深绿建筑出现面片缺失）。

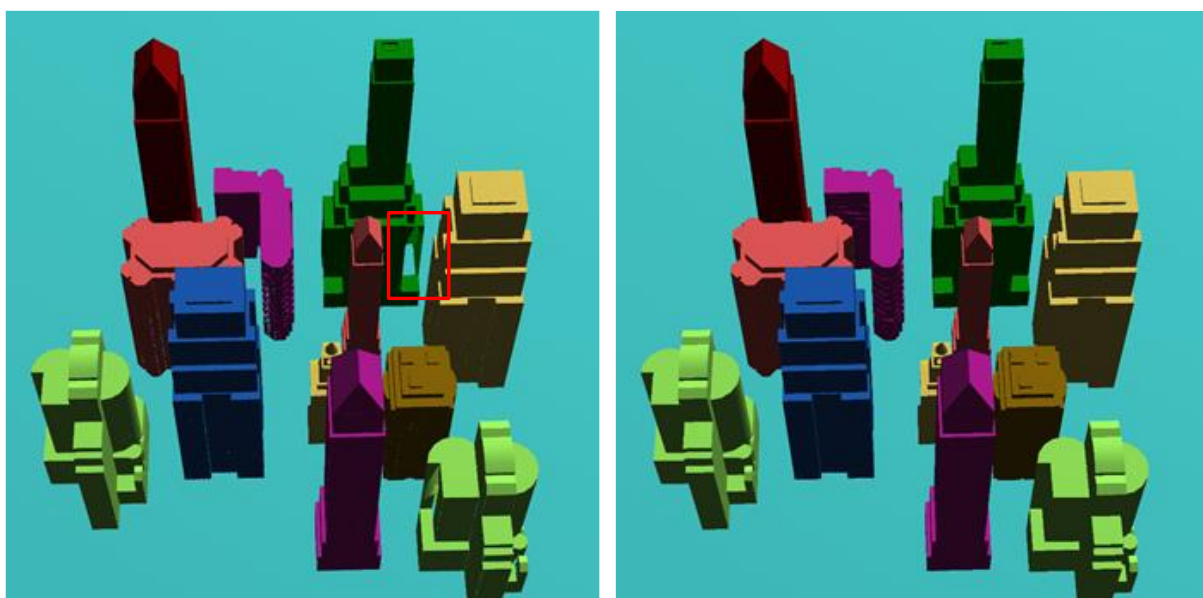


图 19 视点在面光源区域外本章方法重建结果（左）和用真实场景绘制的图像（右）

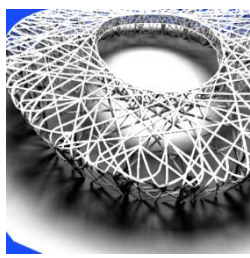
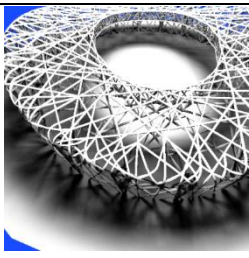
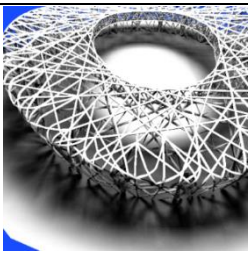
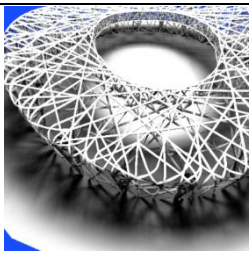
3.4.2 阴影生成质量

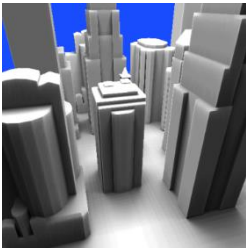
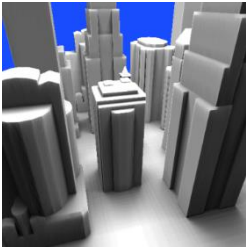
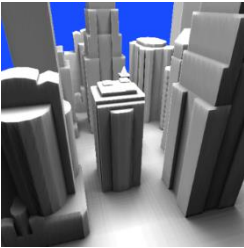
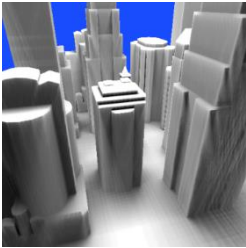
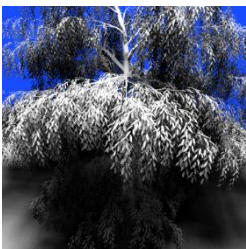
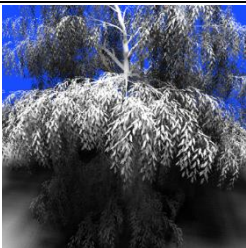
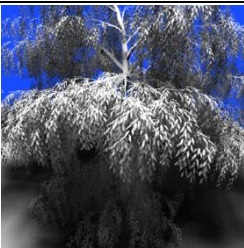
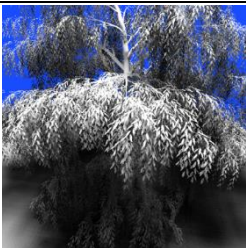
本章测试和对比了非冗余绘制采样方式在测试场景上的软阴影绘制效果。表 8 对比

了非冗余方法（第二列）跟真实几何体（第一列），传统深度图像采样方法（第三列）和深度剥离方法离散采样方法（第四列）生成的阴影图。所有的参考采样图像分辨率尺寸是 1024×1024 ，输出图像分辨率尺寸是 512×512 ，面光源被划分为 16×16 个点光源采样点，面光源的对角线长度等于场景轴对齐包围盒对角线长度。表 8 所示数据斜线前表示图像内像素误差平均值，斜线后表示图像内像素误差最大值。

对于使用几何体生成软阴影的方法，每个点光源下的阴影图是通过绘制原始场景几何体生成的。对于非冗余绘制采样和传统深度图像绘制采样方式，场景采样图像是由面光源的四个角点作为视点绘制生成的。对于深度剥离方式，每一层深度采样图像是在面光源的中点作为视点绘制生成的。在本节中我们关注三种场景的离散采样方法（非冗余绘制采样，传统深度图像采样绘制和深度剥离方法）对原始场景生成阴影图的近似表达能力对比，因此本章使用原始场景生成的阴影图作为真实基准值来测量这三种绘制采样方法引入的近似误差。可见性误差定义为每个像素中对光源采样点可见性解析错误的数量。由于本章将光源离散成 16×16 个点光源采样点，可见性误差范围是从 0 到 255。强度误差定义为每个像素的阴影值误差，它的范围也是由 0 到 255。可见性误差是每个像素的对点光源可见性误判的绝对值之和，而强度误差则是对点光源可见性误判的代数和。在强度误差判断中一个错误的光源可见判断可能会和一个错误的阴影判断即光源不可见判断中和。所以，可见性误差值肯定会比强度误差值高。

表 8 使用原始图像和三种绘制采样方式生成的阴影图表

模型	几何体（真实基准）	非冗余绘制采样	常规绘制采样	深度剥离
鸟巢		 强度误差=3.00/106 可见性误差=6.31/110	 强度误差=5.62/179 可见性误差=7.72/179	 强度误差=3.58/137 可见性误差=7.40/141
草		 强度误差=1.71/112 可见性误差=3.70/116	 强度误差=5.15/181 可见性误差=6.12/181	 强度误差=2.55/199 可见性误差=4.41/199

城市		 强度误差=1.60/48 可见性误差=2.07/48	 强度误差=3.46/74 可见性误差=3.64/74	 强度误差=11.36/125 可见性误差=12.62/125
		 强度误差=9.21/165 可见性误差=17.59/165	 强度误差=13.76/126 可见性误差=21.12/128	 强度误差=10.99/148 可见性误差=18.76/148

非冗余场景绘制采样方式相对于传统深度图像采样方式和深度剥离场景采样方式有更小的误差。由于最复杂的树场景需要不止四张深度图像进行完全采样，我们用树场景测试在极端情况下各个采样方法的对比情况。非冗余绘制由于在所有的场景的误差都较小，所以相比其他采样方法具有更好的鲁棒性。深度剥离方法在城市场景表现较差，传统深度图像采样方式在鸟巢和草场景表现较差。

表 9 展示了本章所对比的三种采样方法可见性误差和强度误差随着采样图像分辨率变化的情况。不出所料，这两种误差会随着分辨率增大而减少。与之前的测试结果一致，非冗余绘制对其他两种采样方式更有优势。

表 9 不同分辨率下三种采样方式的软阴影误差表格

模型	误差类型	分辨率=512			分辨率=1024			分辨率=1536		
		非冗余	传统绘制	深度剥离	非冗余	传统绘制	深度剥离	非冗余	传统绘制	深度剥离
鸟巢	强度误差	4.73	8.65	4.83	3.00	5.62	3.58	2.56	4.45	3.20
	可见性误差	9.57	11.3	10.2	6.31	7.72	7.40	5.04	6.23	5.95
草	强度误差	3.20	7.98	4.57	1.71	5.15	2.55	1.26	4.24	1.84
	可见性误差	7.21	9.62	8.34	3.70	6.12	4.41	2.55	4.94	3.02
城市	强度误差	3.54	7.40	15.8	1.60	3.46	11.3	1.10	2.28	9.97
	可见性误差	4.74	7.80	18.5	2.07	3.64	12.6	1.36	2.39	10.7

树	强度误差	11.2	16.2	11.4	9.21	13.7	10.9	8.90	13.3	10.8
	可见性误差	25.2	28.3	25.9	17.5	21.1	18.7	14.1	18.4	16.0

表 10 展示了可见性误差和强度误差随着光源尺寸变化改变的情况。单元格中斜线前表示图像空间像素误差平均值，斜线后表示图像空间像素误差最大值。光源的尺寸按照与场景轴对齐包围盒对角线长度的倍数或者分数值给定。这两种误差会随着光源尺寸的增大而扩大，这是因为随着面光源尺寸的增大，生成的四张采样图像将必须覆盖更大的视角范围，采样密度下降而带来的采样质量损失。传统绘制方式形成的软阴影误差会比非冗余绘制采样方式大，这是因为非冗余采样方式捕获的采样点集始终是传统采样方式采样点集的超集。深度剥离方法偶尔会比非冗余采样方法好，但是它的近似表达能力不稳定，在有些情况下近似误差会很大。例如在城市场景中，当光源尺寸比例是 1 时，深度剥离方法形成的误差高达 11.4，而非冗余绘制方式仅为 1.60。

表 10 软阴影误差随着光源尺寸的变化情况表

模型	误差类型	尺寸比=1/4			尺寸比=1/2		
		非冗余	传统绘制	深度剥离	非冗余	传统绘制	深度剥离
鸟巢	强度误差	3.22/133	6.42/141	2.90/108	3.23/154	6.25/155	3.15/147
	可见性误差	5.94/143	7.13/141	6.17/117	6.28/154	7.42/155	6.67/147
草	强度误差	1.53/141	3.57/164	2.53/180	1.62/211	4.24/150	2.54/201
	可见性误差	3.05/153	4.16/164	4.22/180	3.40/211	4.88/150	4.31/201
城市	强度误差	2.38/167	4.94/167	3.95/142	2.06/77	3.81/81	5.98/115
	可见性误差	3.39/167	5.13/167	5.89/148	2.86/89	4.01/81	7.50/115
树	强度误差	5.87/187	12.37/169	7.31/218	6.50/177	13.02/142	8.81/205
	可见性误差	12.90/187	17.28/169	14.40/218	14.48/177	19.30/142	16.30/205
		尺寸比=1			尺寸比=2		
鸟巢	强度误差	3.15/147	3.00/106	5.62/179	3.58/137	4.94/128	5.70/120
	可见性误差	6.67/147	6.31/110	7.72/179	7.40/141	8.28/128	9.44/122
草	强度误差	2.54/201	1.71/112	5.15/181	2.55/199	5.55/148	2.79/110
	可见性误差	4.31/201	3.70/116	6.12/181	4.41/199	6.92/148	4.68/110
城市	强度误差	1.60/48	3.46/74	11.36/125	11.36/125	2.67/88	19.91/110
	可见性误差	2.07/48	3.64/74	12.62/125	12.62/125	3.01/90	21.27/110

树	强度误差	8.81/205	9.21/165	13.76/126	10.99/148	16.68/169	15.17/132
	可见性误差	16.30/205	17.59/165	21.12/128	18.76/148	25..06/169	22.97/132
		尺寸比=4					
鸟巢	强度误差	2.67/128	4.13/120	11.20/143			
	可见性误差	5.74/128	7.52/120	13.98/147			
草	强度误差	2.83/94	5.95/137	3.59/98			
	可见性误差	4.98/94	7.86/137	5.40/98			
城市	强度误差	1.95/63	6.77/81	29.64/121			
	可见性误差	2.30/64	7.35/81	31.11/121			
树	强度误差	24.20/148	23.29/159	23.32/138			
	可见性误差	30.97/148	31.44/159	30.24/140			

3.4.3 阴影生成效率

表 15 给出了三种场景采样方法应用算法 5 在实验场景下软阴影绘制的三个主要步骤的耗时情况。三种采样方法时间效率表现相近。非冗余绘制采样算法比传统深度绘制算法稍慢，但是和深度剥离方法效率相当。在软阴影绘制中最耗时的操作是对每一个光源采样点用重建的场景进行阴影图绘制生成。

表 11 软阴影绘制时间表

模型	非冗余绘制	传统绘制	深度剥离
步骤 1: 参考深度图像绘制[毫秒]			
鸟巢	3.9	2.2	4.2
草	4.8	2.2	5.1
城市	4.4	2.1	3.6
树	7.2	5.2	8.6
步骤 2: 重建[毫秒]			
鸟巢	55	90	44
草	88	117	82
城市	112	131	98
树	68	94	57

步骤 3: 软阴影绘制[毫秒]			
鸟巢	710	771	490
草	735	926	642
城市	802	976	541
树	807	954	628

如果要想在软阴影绘制上场景的图像采样重建方法比直接使用原始场景绘制更有优势,重建的近似几何场景就必须要比原始场景更加简单。换言之,原始场景需要比重建的场景网格复杂才能使场景图像采样方法更有效率上优势。图 20 展示了三种采样方法重建结果与使用原场景在阴影图生成上时间随着场景复杂度的变化情况。这里使用不同场景复杂程度的草场景作为测量数据集。场景的离散重建近似网格在阴影图生成时间上变化幅度不大而且会在原始场景三角面片超过二百五十万时体现出时间效率上的优势。

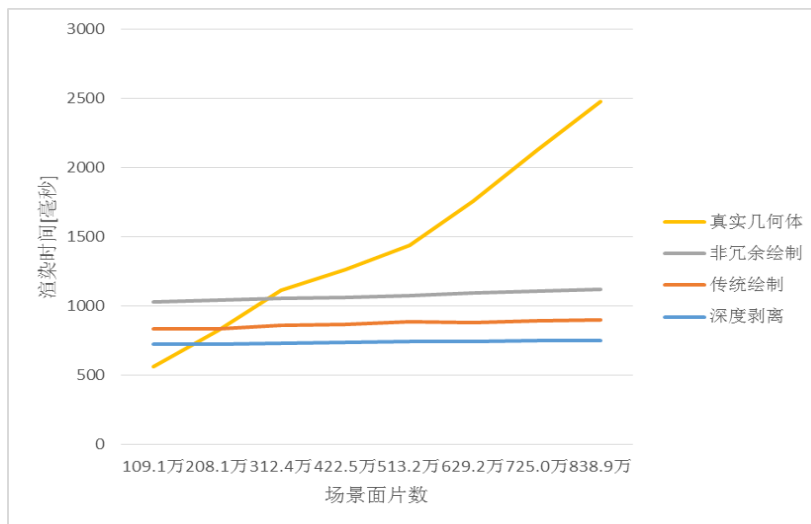


图 20 阴影图绘制时间随场景复杂度的变化情况

3.4.4 算法讨论和局限性分析

和深度剥离方式相似,由于采样非冗余场景意味着采样不连续的几何表面,非冗余绘制生成的采样图像相比于传统采样图像采样连续性要差。这种采样的低连续性导致非冗余绘制图像不适用于传统深度图像的图片压缩算法。

非冗余绘制能快速计算和提高采样点的采样质量。与其他采样方式相似,非冗余绘制采样方式重建场景复杂度在大于原始场景复杂度时会有性能上的损失。

在软阴影绘制中,从面光源的四个角点对场景进行采样是个直观的方式。虽然非冗

余绘制有更强的采样能力，但是这不能保证所有需要的场景采样点都被捕获到。对于像树一样的复杂场景，还需要更多的采样图像才能采样到足够多场景采样点。使用者可以使用错误尝试法，通过定义一个选定阈值来决定还需要多少张图像才能使得阴影计算误差小于选定误差。

软阴影绘制中虽然平均错误很少，但是平面空间内错误的最大值很大，比如，在表 8 中，对于非冗余绘制产生的可见性错误最大值分别是 110, 116, 48, 165。这是因为可见性错误在复杂场景里面并不是平缓变化的，可能会出现某个局部像素中大部分的可见性解析都是错误的。虽然错误值很大，但是非冗余绘制的最大错误还是比传统采样方法（误差是 179, 181, 74, 128）和深度剥离方法（误差是 141, 199, 125, 148）小。

3.5 小结

本章针对非冗余绘制图像中的像素碎片化特点，提出一种适用于非冗余绘制图像的场景重建方法。这种重建方法使用点重建和网格重建的混合思想，通过额外绘制记录深度图像的像素角点信息的偏移图像，完成非冗余绘制图像的密闭几何体重建，重建的几何体没有过度的绘制开销。本章提出一种应用非冗余绘制算法及其几何重建方法的软阴影的绘制算法，并且通过实验结果分析表明，非冗余绘制采样方法的采样性能上的优势能传递到场景重建和近似几何体的软阴影绘制质量上的优势，非冗余绘制图像生成的软阴影效果图相对于其他场景采样方法如传统深度图像绘制采样方法和深度剥离绘制采样方法生成的效果图，相对于物理真实几何体生成的软阴影绘制效果近似误差更小。实验结果同时表明，在重建几何体规模小于原始场景规模时，使用本章方法能提高绘制效率。

第四章 基于像素聚类深度图像的反射绘制方法

本章提出一种基于像素聚类深度图像的镜面反射和光泽反射绘制的方法。我们把相机视点发出的光线称为一阶观察采样光线，这些光线可以由内置投影参数的小孔相机模型建模，一阶观察采样光线采样的场景的可以由投影后绘制出的深度图像近似表达。但是二阶光线譬如说由场景中反射材质生成的反射光线的采样场景时使用深度图像近似表达就比较困难。本章提出一种基于聚类相机深度图像的反射绘制方法。本章算法通过利用相邻反射光线之间的连续性生成聚类相机，通过聚类深度图像来按需近似表达被反射场景，再通过反射光线与聚类相机生成深度图像求交完成反射绘制。本文的方法支持复杂动态场景里反射效果的实时逼真绘制。本章方法能够在实时情况下生成高质量的反射效果。整个算法不需要进行预处理，因此相比于光线跟踪方法能够在动态场景上有更高的绘制效率。

4.1 引言

在计算机图形学应用中，大多数实时绘制方法使用投影后光栅化的绘制流水线方式进行三维场景绘制。因为投影方式能有效地避免了无用的光线与几何体的求交，这种绘制方法非常高效。相比较而言，光线跟踪方式则需要用加速结构来避免部分光线与无关几何体的求交操作。但是，由于反射场景里的反射材质镜像反射了从相机视点发出的一阶光线，由此产生的高阶光线不能被线性投影方式建模，投影后光栅化的绘制流水线方式不适合于绘制反射效果。一种解决这个问题的方法是采用全景深度图像，例如环境贴图来模拟被反射场景，然后用反射光线去索引环境贴图。这种近似方式在被反射物体和反射物体距离较近时会产生很大的误差（见表 12）。

本章提出一种聚类相机生成的深度图像来模拟二阶光线采样点的方法。本文的方法基于反射光线在局部连贯的特性。我们利用这种连贯性把相邻像素的反射光线进行空间聚类。虽然在单个聚类里面的光线是连贯的，但是聚类光线不一定从一个投影中心点发出，所以它们不能被传统的小孔相机模型建模。一种解决方案是减少聚类的尺寸直到用小孔相机模型的拟合误差在可接受范围之内。但是这种方法会产生大量的小孔相机模型，不适合于大型复杂反射场景。另一种解决方案是对聚类内反射光线用非小孔相机模型进行建模，但是这会带来耗时的非小孔相机投影和非线性的光栅化操作，并且即使小的光线近似误差也会导致聚类簇之间的反射绘制效果不连贯。本章的方法采用聚类深度图像表达被反射场景的方法来实现反射绘制。每个聚类反射到的物体场景用绘制深度图

像进行近似。反射光线的颜色信息由反射光线与聚类相机生成的深度图像求交得到。

表 12 用环境光贴图(左)， 光线跟踪 (中)， 和本章方法 (右)生成的反射效果图表

	环境光贴图	光线跟踪	本章方法
起居室			
花瓶			
盥洗室			

本章提出的方法能实时生成高质量反射图像。表 12 展示了本算法生成图像与环境光贴图方法和光线追踪方法的生成图像对比图。环境光贴图方法生成不正确的反射结果（比如，地板上反射的桌子，瓷瓶上反射的地板），这种结果来源于全景贴图方式对场景的错误近似表达。本章方法能绘制出同光线跟踪方式可比的高质量反射效果。本章方法的帧率是 15fps，与之相对的光线跟踪是 1.1fps。本章方法因为不需要预计算所以支持全动态场景。反射光线聚类算法能支持多种复杂反射物。本章算法用聚类相机深度图像按需表达被反射物体场景。同时，本算法通过多根反射光线与聚类深度图像求交支持光泽反射绘制（见图 21，绘制帧率为 5 帧每秒）。

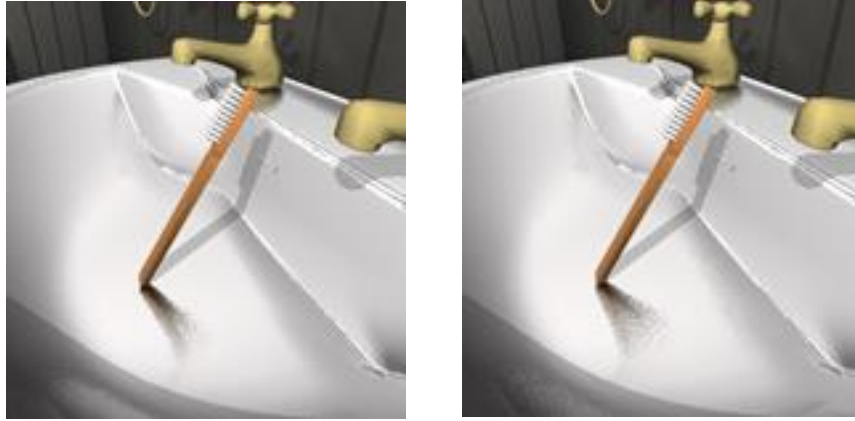


图 21 不同光照反射程度下的光泽反射绘制效果图

本章提出的方法通过为每个聚类绘制一张深度图像的方法来模拟反射光线的绘制。虽然理论上本文的方法可以处理高阶反射光线，但是随光线阶数的增高，反射光线愈发不连贯，用聚类相机来模拟高阶光线将变得愈发困难，生成的聚类数量也会显著上升。而且在通常情况下，低阶反射光线反射错误比较明显，而高阶反射的错误不易察觉，因此，本文方法只关注二阶反射绘制，对于更高阶数的反射现象本文使用环境贴图来进行近似绘制。

4.2 算法整体流程

给定一个包含漫反射表面（即物体表面材质是由纯漫反射特性材质构成），镜面反射表面（即物体表面材质是由镜面反射成分和漫反射成分混合而成），光泽反射表面（即物体表面材质是由漫反射成分和单瓣轴对称 BRDF（single-lobe symmetrical BRDF）模型混合而成）的三角形建模场景 S ，给定观察视点 e ，反射场景 S 以 e 为视点按照算法 6 进行反射绘制：

算法 6：基于聚类深度图像的反射绘制

输入： 带有反射材质的三角形建模场景 S ，观察视点 e

输出： 反射绘制输出图像 I

1. 从 e 绘制 S 。对每个像素 p 记录：
 - a. 漫反射颜色成分 $p.rgbd$
 - b. 镜面反射程度和光泽反射程度 $p.s$ 和 $p.g$
 - c. 法线 $p.n$ 和深度 $p.z$
 - d. 反射物体 ID, $p.rID$
 2. 对于反射物体像素进行空间聚类
-

3. 对于每个聚类 C ，根据以下步骤完成反射绘制

- a. 构建场景聚类相机 K
- b. 由 K 绘制场景 S 得到深度图像 D
- c. 对于 C 中的每个像素 p
 - i. 对于 p 的每个反射光线 r_i

$$r_i \text{ 与 } D \text{ 相交, 即 } d_i = r_i \cap D$$
 - ii. 计算反射颜色成分 $p.rgb_n = G(d_i, p.g)$
 - iii. 像素颜色 $p.rgb = \text{LERP}(p.rgb_d, p.rgb_n, p.s)$

算法 6 主要分为三个步骤。步骤一通过执行场景绘制生成屏幕空间像素，同时计算每个像素的漫反射部分，设置每个像素的镜面和光泽反射程度，通过传统的顶点插值计算出每个像素的法线和深度，并且设置这个像素所属的反射物体 ID。像素的镜面反射程度 $p.s$ 取值从 0 即纯漫反射材质到 1 即纯反射材质。像素的光泽反射程度 $p.g$ 取值从 0 即镜面反射到 1 即反射瓣（Reflection Lobe）最宽时的光泽反射。步骤二用章节 4.3 描述的方法对有着相似反射光线的像素进行屏幕空间的聚类。步骤三依次对每个聚类计算反射，最后完成单幅输出图像的反射绘制。步骤三中具体步骤为，首先对每个聚类按照章节 4.4 生成一个能包含所有反射光线的聚类相机（步骤 3.a），然后由该聚类相机绘制场景得到深度图像作为该聚类反射到的场景的近似（步骤 3.b），最后根据像素反射光线同深度图像的求交完成像素的反射效果着色。

聚类相机生成的深度图像将用来计算此聚类像素的反射颜色。对于聚类里面的每个反射像素，该像素对应的反射光线与聚类相机生成的深度图像按照章节 4.5 描述进行求交（步骤 3.c）。对于镜面反射表面（即 $p.g$ 值为 0），每一个像素对应一条反射光线，而该反射光线是由像素法线、像素位置和观察视点位置所确定。光泽反射程度值越大，该像素反射的光线越多。当前像素的反射成分是用像素反射材质 BRDF 反射模型对应的表面反射函数 $G(d_i, p.g)$ 按光泽反射程度 $p.g$ 混合反射光线与深度图像的相交颜色采样 d_i 所得。像素的最终颜色是由反射成分光照和漫反射成分光照按照镜面反射程度 $p.s$ 混合而成（步骤 3.c.iii）。

4.3 反射光线聚类

第一遍场景绘制生成屏幕空间像素并计算出决定反射光线走向的像素法线值。本章

的方法利用反射光线的连贯性特点，将相邻的反射像素的反射光线聚类到簇里。本章的方法基于如下考虑设计聚类算法：

- a. 聚类的数目应尽可能的少，因为每个聚类的生成都意味着要绘制生成对应的聚类相机深度图像。
- b. 生成的聚类相机要尽可能紧凑使得生成的小孔相机既刚好按需捕获聚类所反射的场景又不至于捕获太多不相干的场景。
- c. 相机的聚类算法要快以满足反射绘制的实时要求。

本章的方法将屏幕内反射像素根据法线所在的球坐标桶聚类到一个三维变量里面。本章的屏幕空间反射上像素聚类方法按照如下步骤进行：

离线预计算：

1. 将反射三角形划分为不同反射物

在线实时计算，对每一帧：

2. 对每个反射物体 R

计算反射物体在屏幕空间的投影面积 a_R

3. 计算屏幕可见反射物的数量 n_v

4. 对每一个反射像素 p ，将像素 p 划分到聚类号 (i, j, k) ，其中

$$i = \text{MAXTHETABINS} \times p.n.\theta / 360^\circ \times a_R / n_v$$

$$j = \text{MAXPHIBINS} \times p.n.\varphi / 180^\circ \times a_R / n_v$$

$$k = p.rID$$

只按像素的法线朝向 (i, j) 进行聚类将将会导致两个空间距离很远的相似朝向像素也会聚类到同一个聚类簇里。比如，表 12 最上一行的花瓶、茶壶和灯上具有含有相似法线的像素。将这些像素聚类到一个聚类簇里面将会造成这个聚类相机的范围非常大。这违背了设定的聚类规则。反射像素聚类也可以由输出图像里进行由上向下的屏幕空间反射像素四叉树构建进行聚类，但是这种方式计算太慢。

步骤 1。离线计算时，我们将反射三角形按物体进行划分，物体编号将作为聚类三维变量中的第三个变量，另两个变量则存储的是法线球坐标哈希值，这样避免了把不同物体的相似朝向像素划分到同一个聚类里。物体编号按照场景物体天然划分进行编号。图 22 分别可视化了起居室中的 25 个和盥洗室场景中的 15 个反射物体，灰色表示漫反射物体。本章的方法只对反射像素进行反射绘制，而在离线计算中也从场景中区分出反

射表面材质物体，这样可用于快速进行屏幕空间聚类。当反射物体的数目确定下来后， θ 和 φ 维度上的聚类簇数组分辨率需要每帧都进行设置，设置的参数随输出图像和反射物体的不同而变化。 θ 和 φ 维度上的聚类分辨率依赖于两个数值。

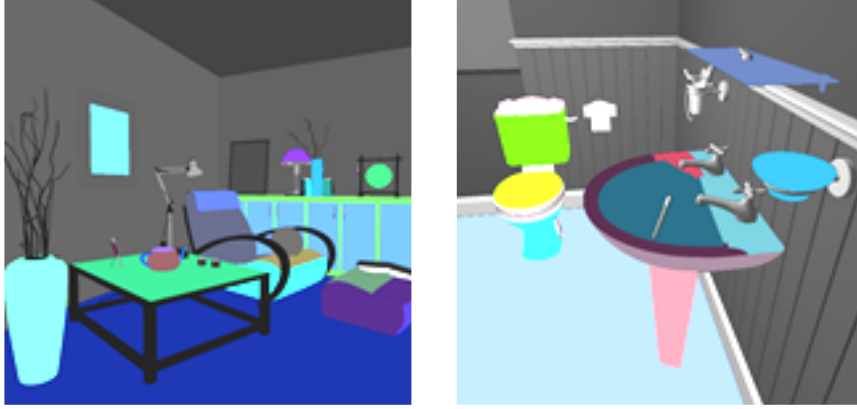


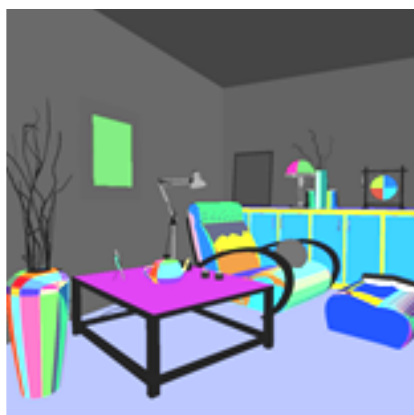
图 22 场景反射物体可视化图

步骤 2。其中一个数值 a_R 记录了反射物在屏幕空间投影像素数目和输出图像空间中所有可见反射物像素数目的比值。

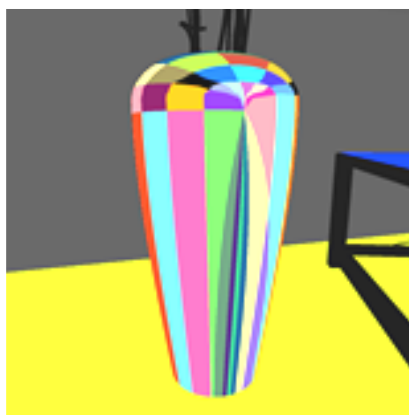
步骤 3。第二个数值 n_v 记录了输出图像中 a_R 值不为 0（步骤 2）的可见反射物数目。

步骤 4。在实时在线计算中，每一个反射像素根据法线和所对应物体的编号赋值给对应的聚类簇。 θ 和 φ 的最大可能聚类簇数目 $\text{MAXTHETABINS} \times \text{MAXPHIBINS}$ 用 a_R 和 n_v 两个参数进行调整。 MAXTHETABINS 和 MAXPHIBINS 是常数值，在本章所涉及的实验中分别设置 13 和 8。反射物在屏幕空间的投影像素越多，生成的聚类越精细，反射物的数目越多，生成的聚类越粗糙。虽然在文中的两个场景里面最大的可能聚类数为 $25 \times 13 \times 8 = 2,600$ 和 $15 \times 13 \times 8 = 1,560$ 。但是，聚类的确切数目只和屏幕上有可见像素的聚类数目有关。譬如，如果有一个在屏幕空间的球形，聚类的最大数目只是 $1 \times 13 \times 8/2 = 52$ ，表示球体上只有一半的像素是可见的。

图 23 展示了表 12 和图 21 里面用来绘制反射的聚类，灰色表示漫反射物体。场景中的地板只用一个聚类来表示。虽然桌子表面和地面虽然他们的法线是相同的，但是他们还是被分配到了两个不同的聚类。虽然按照法线球坐标聚类方法在法线轴方向处生成了较多的相机，但是球坐标系统以其简要性而被采用。根据像素的法线而不是反射光线进行聚类不仅避免了在物体边缘产生太多的聚类，而且不同帧之间效率比较稳定。



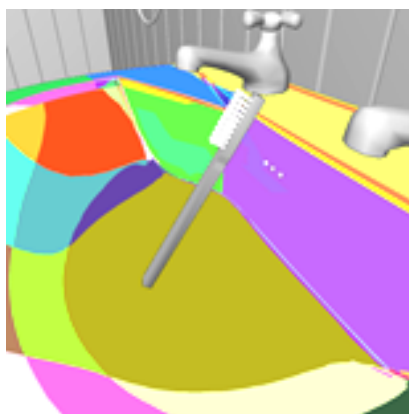
123 个聚类



38 个聚类



121 个聚类



53 个聚类

图 23 可视化表 12 和图 21 中屏幕像素聚类示意图

4.4 聚类相机的构建

当屏幕空间的反射像素聚类划分完毕,接下来进行聚类的反射效果计算。第一步是为当前聚类建立传统小孔相机模型,用该小孔相机绘制出的深度图像近似表达此聚类反射到的场景。本章选择小孔相机模型是基于如下考虑:第一,小孔相机支持绘制深度图像采样捕获场景;第二,反射光线在深度图像上的投影是一条直线,这样使得光线与深度图像进行求交操作简洁高效;第三,小孔相机能完美地对平面反射物体生成的反射光线进行建模。平面反射物体是人造反射场景的重要组成部分,能正确处理平面反射是反射算法的一个重要考量。

建立聚类相机的目的是捕获聚类像素所反射的场景。相机构建算法需要考虑该小孔相机模型要有足够大的视角使得相机能包含该聚类反射光线的场景交点;同时也要考虑聚类相机的相机观察光线要尽可能地与反射光线近似,这样能缩小生成深度图像近似反射场景的遮挡误差。遮挡误差是在反射光线相交到深度图像没有采样到的被反射物体时产生。聚类相机按照如下步骤进行构造(图 24)。

$$\text{a. } P_c = \frac{1}{N} \sum P_i$$

$$\text{b. } n_c = \frac{1}{N} \sum n_i$$

- ## 2. 设置聚类相机远裁剪面 D

- ### 3. 设置相机投影中心 e_c

a. 计算 $e_0 = \text{Reflect}(e, P_c, n_c)$

b. 沿连线 e_0P_c 进行投影中心偏移: $e_c = e_0 + (P_c - e_0)f$

- #### 4. 设置聚类相机的图像平面

5. 计算绘制图像的分辨率 $w \times h$

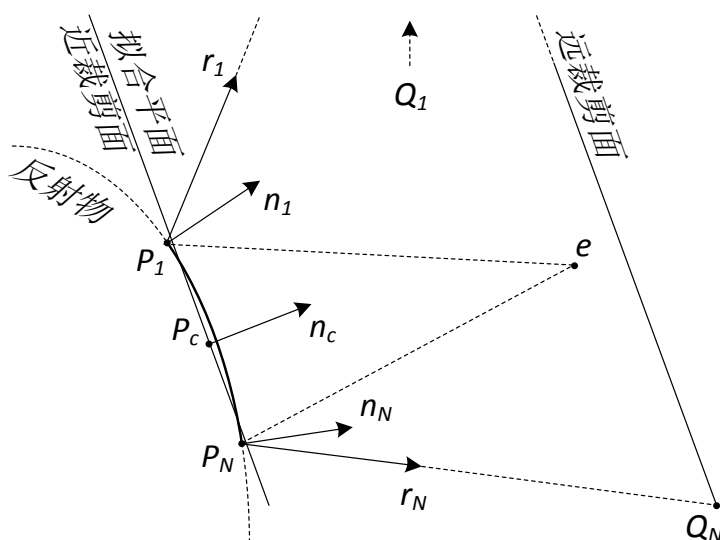


图 24 聚类相机拟合平面和远近裁剪面示意图

步骤 1。聚类的拟合平面由聚类的中心点和聚类平均法线定义。聚类的中心点是所有该聚类像素对应的 3D 反射物表面点的平均点。聚类的平均法线是指聚类像素对应法线的平均法线。该聚类的拟合平面同时也作为聚类相机的近截面。图 24 展示了聚类相机的构建的 2D 示意图。一个弯曲的反射物体上聚类像素分布由 P_1 到 P_N ，法线分布由 n_1 到 n_n 。观察相机的视点设置在 e 点。

步骤 2。对于镜面材质反射物体，聚类相机与远裁剪面设置的距离设置为场景的对角线长度。对于光泽反射物体，聚类相机远裁剪面设置要近些，具体设置参加章节 4.6。

步骤 3。聚类相机的投影中心 e_c 应设置在发出的采样光线能尽可能地模拟该聚类反射光线的点。本章方法构建的 e_c 可以使反射光线起点在聚类拟合平面上投影的轴对齐包围盒 (Axis-Aligned Bounding Box, AABB) 和反射光线终点在拟合平面上的 AABB 相似。如图 24 所示, P_1 和 P_N 展示了光线 r_1 到 r_N 的起始端点, 而 r_1 和 r_N 的终点为 Q_1 和 Q_N 。

本章的算法用了两小步来计算 e_c 。首先视点 e 按聚类拟合平面镜像对称到点 e_0 (图 24)。然后把 e_0 朝聚类中心 P_c 进行偏移。偏移量是由标量 f 控制。 f 的设置目的是使反射光线的起点在聚类拟合平面上的投影包围盒 $aabb_n$ 和终点的投影包围盒 $aabb_f$ 相似。本章的算法根据等式 6 设定 f 值。式中 d_n 是 $aabb_n$ 的对角线长度, d_f 是 $aabb_f$ 的对角线长度。

$$f = \frac{d_n}{d_f} \quad 6$$

图 25 展示了图 24 中的反射物是凸体情况下的相机生成。点 $Q_1', P_1', P_c', P_N', Q_N'$ 表示图 24 中对应点投影到聚类平面上的投影点。线段 $Q_1'Q_N'$ 表示 d_f , 线段 $P_1'P_N'$ 则表示 d_n 。对于 e_0 点, $aabb_n$ 小于 $aabb_f$, 即 $d_f > d_n$, $0 < f < 1$ 。导致投影中心由 e_0 向 P_c 偏移。 e_0 的偏移同时会缩小 $aabb_n$ 和 $aabb_f$ 。相比于 $aabb_n$, $aabb_f$ 减少得更大, 最终使得反射光线起点和终点的投影点包围盒尺寸相近。因为 e_cP_1' 的朝向比 e_0Q_1' 更能拟合反射光线的方向, 将聚类相机投影中心定在 e_c 点比在 e_0 能更好地模拟反射光线。

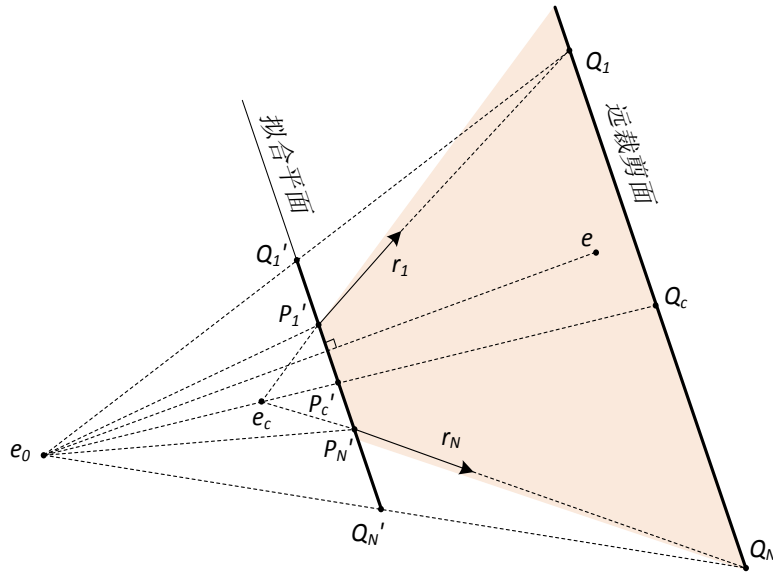


图 25 聚类相机视点 e_0 的构建示意图

对于平面反射物体对应的聚类相机生成, 因为平面所有的反射光线都会汇聚到 e_0

点，所以聚类相机应该采用 e_0 作为投影中心。这是完美地对反射光线进行建模的方式，而且这样建模生成的深度图像没有遮挡误差。在本文所述的聚类相机生成方法里，对于平面反射物体来说，反射光线的起点包围盒在拟合平面上的投影 $aabb_n$ 和终点包围盒的投影 $aabb_f$ 一致， $f = 1$ ， e_c 设置正如理想情况设置在 e_0 点，与期望相同。

步骤 4。聚类相机的图像平面是由将反射光线的起点和终点点集按照最终计算出的 e_c 为投影中心再进行投影。图像平面范围设置为起点和终点的投影点包围盒 $aabb$ 。采用这样方式生成的聚类相机的观察角度可以完全包括这个聚类的反射光线区域（图 25 中的橘黄色区域）。

步骤 5。聚类相机生成深度图像的分辨率应该设置为与聚类的像素数目成匹配。聚类像素的连续行或者连续列像素投影到拟合平面上，计算列间或者行间的平均投影距离，用此距离来度量聚类相机图像平面上投影像素平均大小 p 。聚类相机生成深度图像尺寸为 $w \times h$ ，其中 $w = aabb.w/p$ ， $h = aabb.h/p$ 。

4.5 反射光线与深度图像求交

聚类相机构建后按照聚类划分依次对场景进行绘制得到聚类被反射物体的深度图像。深度图像是该聚类被反射物体的强有力表示：它可以由绘制流水线快速生成，每个采样点既能表示被反射物体光照也能表示被反射物体深度；由于深度缓存纹理的可控层级细节特性，深度图像可以按用户要求表达被反射物体不同层次的细节；深度图像可以和反射光线进行有效的求交操作。

给定光线 R 与聚类相机生成的深度图像 DI 。搜索 R 与 DI 的光线最近相交点的过程是首先将 R 投影到 DI 上。假设 R' 是 r 在生成 DI 的聚类相机正则空间上的投影；相交搜索点 t 从 R' 的投影近端点开始搜寻，步长为一个像素，沿 R' 搜索相机点。如果相交处光线深度比图像上的索引深度还深，本章算法就认为反射光线与深度图像存在交点，并且停止交点搜索算法。如果搜索到 R' 的终点也没找到相交像素，本章算法认为该反射光线与深度图像没有交点。反射算法与聚类深度图像的求交算法由算法 8 给出。求交过程利用了现代 GPU 的并行性，可以同时为多个反射像素进行并行计算。

算法 7：反射光线与聚类生成深度图像求交算法

输入：光线 R （起点 R_{begin} ，终点 R_{end} ）与聚类相机 C 生成的深度图像 DI

输出：光线 R 的反射颜色

```
 $t_{begin}, z_{begin} = \text{project}(R_{begin}, C), t_{end}, z_{end} = \text{project}(R_{end}, C) \quad // t_{begin}, t_{end} \in [0, 1]^2$ 
```

```
//将光线端点投影到聚类相机图像空间下端点坐标和端点深度
```

```
stepX = (tend.x - tbegin.x) × DI.x, stepY = (tend.y - tbegin.y) × DI.y
```

```
//搜索线段的步长取决于投影线段在 x 或 y 轴投影像素较长的长度
```

```
delta = stepX > stepY? (tend - tbegin)/stepX: (tend - tbegin)/stepY
```

$$k_1 = \frac{1}{z_{begin}}$$

$$k_2 = \frac{z_{begin} \times z_{end}}{z_{begin} - z_{end}} \quad // k_1, k_2 \text{ 是光线非线性搜索的两个系数}$$

```
for(t = tbegin; t <= tend; t += delta)
```

$$\alpha = \frac{t - t_{begin}}{t_{end} - t_{begin}} \quad // \text{投影空间比例}$$

$$depthR = \frac{1}{\frac{\alpha}{k_1} - k_2} \quad // \text{由投影空间比例获取搜索点光线深度}$$

```
depthDI = tex(DI, t).w \quad // \text{搜索点处取得深度图像深度}
```

```
if(depthR < depthDI)
```

```
    //如果搜索点深度比深度图像上深度深，认为该搜索点为交点，
```

```
    return tex(DI, t).xyz \quad // \text{返回交点处颜色}
```

```
endif
```

```
end
```

```
return texCube(Rend - Rbegin)
```

```
//如果找不到交点，按照反射光线方向返回环境光贴图颜色
```

由于在投影空间内直线深度是非线性变化的，本章算法应用投影空间下的齐次坐标变化规则求得搜索点深度，再用搜索点深度进行光线相交判断。算法 8 中参数值 k^1, k^2 为给定光线非线性搜索的常数化参数^[97]，光线在投影空间的非线性深度变化在图形学研究领域已经被人熟知。它常被作为基础算法应用于多视角下光线跟踪^[17]、三维扭曲^[75]、绘制几何层级细节^[80]和折射绘制^[97]中。本章将光线的非线性搜索应用在反射光线与聚类深度图像求交上。如果搜索直到光线在深度图像上投影的终点也没找到相交像素，本章算法认为该反射光线与深度图像没有交点而去使用环境光贴图索引值。

4.6 光泽反射绘制

光泽反射物体的反射光线与镜面反射相比有很多不同：光泽反射光线并不是按照一个确定方向延伸，而是按照以镜面反射光线方向为轴的反射瓣均匀分布，所以反射光线

的角度会扩大。本章的算法用以下的三个改进支持光泽反射绘制：

1. 聚类相机的视景体的构建需要考虑扩大的反射角度。本章的光泽反射绘制算法在聚类相机视景体计算中考虑到扩大后的反射角度，用扩大后的反射光线终点包围盒作为视角的计算参考和图像平面的生成，这样聚类最后绘制生成的聚类深度图像中包含了反射光线扩大后的被反射场景。

2. 光泽反射只对靠近反射物表面的物体进行反射成像。本章的光泽反射绘制算法通过改变聚类相机的远裁剪面来完成这一特性。光泽反射绘制算法使用反射物表面的光泽程度来控制远裁剪面设定。对于粗糙的反射物表面，远裁剪面设定为更靠近反射物体，这样形成的深度图像丢弃了远处被反射物体。反射物表面越粗糙，远裁剪面越近。当物体材质的粗糙程度大到一定程度，即相机远裁剪面和近裁剪面之间的空间小到没有被反射物体时，深度图像将不会绘制被反射物体，这时光线与深度图像的求交也可以省去。为了做出被反射物体由远及近靠近反射物体的反射渐变效果，最终的像素颜色是由镜面反射颜色与物体的漫反射颜色混合而成，而混合的系数在被反射物体靠近远截面时候递减为 0。

3. 光泽反射需要每个反射物采样点的多根反射光线颜色。光泽反射绘制算法对每一个聚类像素生成多根反射光线，对每根反射光线都进行于深度图像的求交操作。反射光线按照以反射光线方向为轴的反射瓣进行均匀采样，反射瓣的宽度跟材质光滑反射程度相关。表面材质越接近镜面反射材质，反射瓣的宽度越小，需要的反射光线数目越小；反之，反射材质越粗糙，反射瓣的宽度越宽，生成的反射光线数目越多。对于本章展示的图片，光泽反射表面生成的反射光线数目为 8 到 49 根。

4.7 实验结果及分析

本章使用两个存在多种反射物体的复杂室内场景验证本章算法。起居室场景（表 12，上和中）有 28 万面片，其中 15 万是漫反射面片，另外 13 万面片是具有反射材质的面片。盥洗室场景（表 12，下）有 9 万面片，其中 4.4 万是漫反射面片，另外 4.6 万片是具有反射材质的面片。

4.7.1 反射质量

镜面反射

如表 12 所示，本章的方法能绘制出高质量的反射效果。跟环境贴图方法不同，即使被反射物体靠近反射物体，反射效果也是正确的，且正确地表示了反射物和被反射物之间的相对位置。同时在聚类之间，反射效果并没有出现不连续点，这一方面是由于本章使用的场景采用连续表面作为反射物表面，致使反射光线的分布也是连续的，另一方面也反映本章的方法是基于物体真实光线的求交，能正确绘制出反射效果。

本章方法生成的镜面反射效果和基于物理方法光线追踪方法生成效果是极其相似的，本章节算法生成图像都是经过均匀 2×2 超采样的 512×512 图片（即需要 1024×1024 个采样点）。作为对比的光线跟踪方法同样生成 512×512 尺寸输出图片，每个像素生成 4 条反射光线。本章使用第三方 NVIDIA 的 Optix 光线跟踪框架^[86]作为对比组。图 26 展示了本章方法生成图像和光线跟踪方法生成图像的误差图。误差图片的像素 RGB 通道平均绝对值都很小，每个通道分别为 6, 3, 9。（像素的每个通道值都放缩到 0-255 区间）。

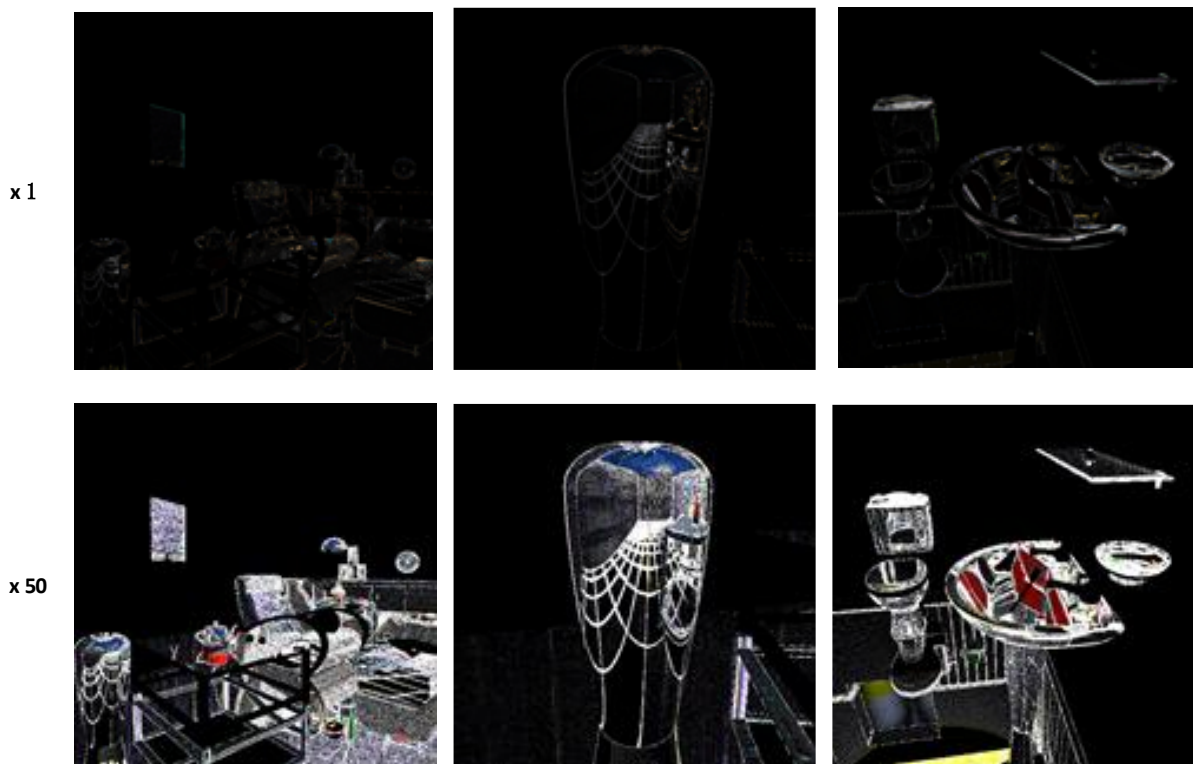


图 26 （上）表 12 中三张本章方法生成图和光线跟踪误差图。（下）放大 50 倍后的误差图

误差来源之一是由于深度图像的采样方式和光线跟踪的采样方式不同。本章的方法使用双线性插值生成深度图像像素中间采样点的值，而光线跟踪直接使用光线进行场景求交采样。

误差来源之二是由于应该出现在深度图像上的被反射物体没有出现在深度图像上。

反射光线并不一定严格经过聚类相机的投影中心，所以有可能一些光线求交到聚类相机没有捕获到的物体上，即使用聚类相机观察光线近似反射光线出现错误，产生遮挡误差。本章测量了一千帧中反射像素遮挡误差。起居室最大/平均像素遮挡误差是 1.3%/0.62%，盥洗室场景是 3.5%/2.75%。遮挡误差可以由聚类的精细程度来控制。聚类数目越多，聚类相机要模拟的反射光线越少，表达的可靠程度越高（图 27）。

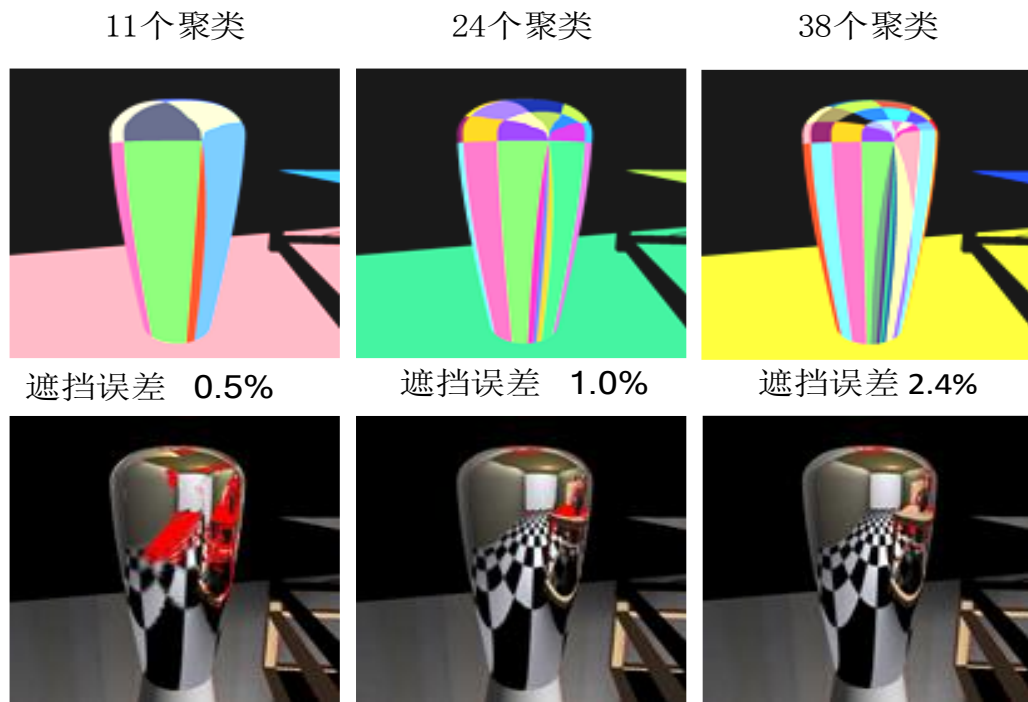


图 27 表 12 中花瓶场景遮挡误差像素（红色）会随着聚类数目增加会减

误差来源之三来源于反射物体的边界，当反射物在屏幕上的投影只有很少几个像素时，这时聚类相机只包含一根或几根光线，这将有可能导致在聚类相机深度图像上找不到交点，本章算法让这样的反射光线在环境贴图中索引颜色值（图 28）。



图 28 表 12 中使用环境贴图的像素（红色高亮）

本章方法用深度图像来表示当前帧的所有被反射物体，这允许了用层次细节来表示

不同层次的被反射物体。深度图像的分辨率与聚类反射光线数目成比例。图 29 展示了当深度图像分辨率下降导致了最后生成反射图像的模糊（花瓶上远处地板的反射图样）和几何边缘的走样（地板上反射的桌子腿）。

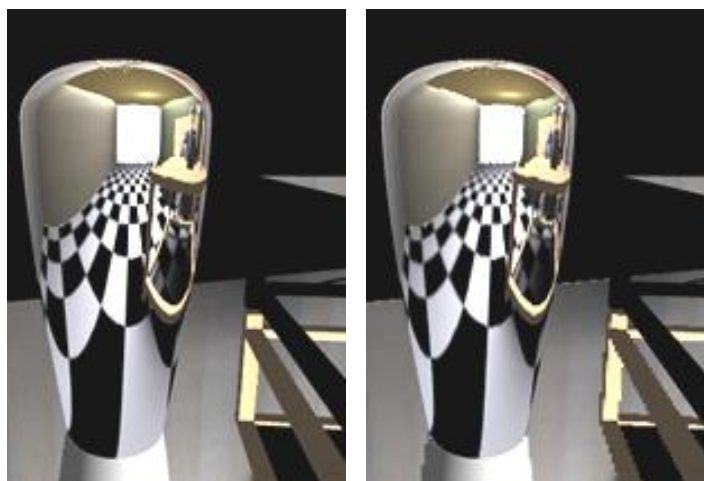


图 29 （左）本算法生成的反射和（右）比当前分辨率低 3×3 倍的聚类深度图像生成的反射光泽反射

本章的方法使用多根光泽反射光线与深度图像求交来计算光泽反射像素的颜色。图 30 展示了本章的方法能绘制出与光线跟踪效果相似的光泽反射结果。图 30, 图 31, 图 32 所展示图像是使用每个光泽反射像素 8 根光泽反射光线生成的图像, 图 30 显示忽略漫反射成分的输出图像, 图 31 显示混合漫反射成分的输出图像, 图 32 展示光线跟踪方法绘制结果。图 33 展示了用每个光泽反射像素 49 根光泽光泽反射光线生成的误差更小的光泽反射图像。

基于聚类的深度图像反射绘制算法有较好的显存空间复杂度, 这是因为我们的算法中作为被反射场景近似的聚类深度图像不必同时存储在 GPU 显存上, 一旦某一聚类的反射特效绘制完毕, 该聚类反射绘制所需的深度图像占用显存空间就被释放等待被下一个聚类深度图像绘制复用。本章所示生成 512×512 反射特效输出图像应用 GPU 显存开销是 87MB, 而其中绝大部分 (64MB) 是被用来作为记录聚类相机空间颜色和深度的帧缓存以及记录屏幕空间位置、法线、聚类编号和颜色的帧缓存的存储空间。



图 30 忽略漫反射成分的低光泽程度（左）和高光泽程度（右）本文反射方法输出图像

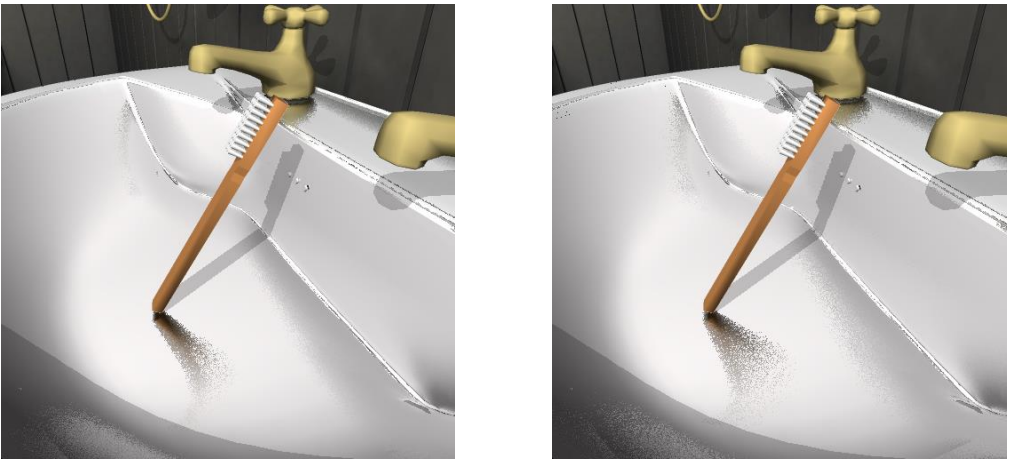


图 31 混合漫反射成分的低光泽程度（左）和高光泽程度（右）本文反射方法输出图像

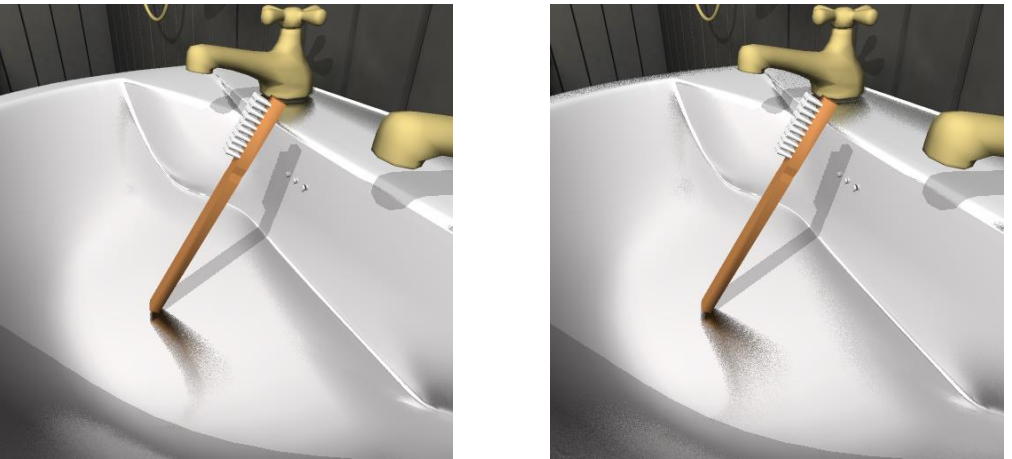


图 32 混合漫反射成分的低光泽程度（左）和高光泽程度（右）光线跟踪方法输出图像



图 33 图 30 中反射像素使用 49 根反射光线生成的反射图

4.7.2 时间效率分析

本章节所做的性能分析是基于 3.4GHz Intel(R) Core(TM) i7-2600 CPU, 4 GB 内存和 NVIDIA GeForce GTX 570 显卡个人电脑平台。

镜面反射

镜面反射与光线跟踪的时间效率对比在表 13 中给出。光线跟踪方法使用层级包围体 (bounding volume hierarchy, BVH) 作为加速结构, 在动态场景中, BVH 加速结构每一帧都进行重建。效率分析是在每个场景里面绘制 900 帧图像的两个独立路径统计的。场景中包含了大量的反射物体使得超过半数的像素都需要进行反射绘制。本章的算法能在四条路径保持每秒 10 帧以上的绘制效率。在较简单的盥洗室场景里, 效率是 Optix 光线跟踪方法的 5 倍, 在复杂的起居室场景效率比 Optix 光线跟踪框架快 10 倍。为了更好地说明效率特性, 本章节对性能进行拆分分析。

表 13 本章方法与 Optix 的效率对比

场景	反射像素 [%]		Optix [帧每秒]			本章方法[帧每秒]		
	最大	平均	最大	最小	平均	最大	最小	平均
起居室	62	50	1.1	1.0	1.1	25	13	18
	69	58	1.1	0.9	1.0	20	11	15
盥洗室	55	45	3.4	3.1	3.3	20	14	17
	58	44	3.4	3.1	2.3	29	15	18

表 14 给出了本章算法各主要步骤在两个场景上的最大和平均时间。第一遍绘制, 用场景几何体进行简单光照绘制生成屏幕空间像素和第二遍绘制, 将物体表面反射像素按照法线朝向的球坐标进行桶聚类, 只占用很少的时间。聚类相机的生成就要耗时很多, 因为需要在像素聚类算法中进行多遍 GPU 核函数运算操作, 而此操作涉及的拟合平面法线, 平均距离点和轴对称包围盒计算的时候都会涉及到存储区关键块的同时写操作。

深度图像的绘制操作同样费时因为它需要对每一个聚类相机绘制一遍场景。最后反射光线和深度图像的求交需要花费深度图像的绘制一半的时间。

表 14 章节 4.2 中各个步骤的耗时图（毫秒）

场景	第一遍绘制 & 像素聚类 步骤 1 & 2		聚类相机构造 步骤 3. a		深度图像绘制 步骤 3. b		光线与深度图像求交 步骤 3. c. i	
	最大	平均	最大	平均	最大	平均	最大	平均
起居室	0.9	0.6	30	23	37	26	15	7
	0.8	0.5	28	24	38	28	28	14
盥洗室	0.8	0.5	24	20	39	28	18	11
	0.7	0.5	26	20	30	24	16	8

其他绘制的参数例如：聚类的数目（表 15）、生成图像分辨率（表 16）、聚类相机生成深度图像尺寸（表 17）和场景中反射物体三角面片数目规模（表 18）也会对本章算法产生影响。

表 15 不同聚类数目下的平均帧率

聚类数	156	141	123	86	32
帧率 [帧每秒]	12	13	15	15	20

表 16 不同生成图像分辨率下的平均帧率

生成图像分辨率	2,048 × 2,048	1,024 × 1,024	512 × 512	256 × 256
帧率 [帧每秒]	7.5	15	23	29

表 17 不同聚类生成图像分辨率下的平均帧率

生成图像分辨率	1/2 × 1/2	1 × 1	2 × 2
帧率 [帧每秒]	15	15	14

表 18 漫反射三角形数目不同规模下生成图像分辨率下的平均帧率

漫反射三角形数[x1,000]	289	162	132	63
帧率 [帧每秒]	10	12	15	15

本章算法在第一遍绘制时生成反射像素，并把反射像素划分到不同聚类里。因此，本章算法并不太依赖于有反射材质的三角形面片数目规模，而更依赖于聚类的数目（表 15）。绘制算法的所有步骤都与最后生成图像的分辨率相关：最后生成图像越大，生成的反射像素越多，平均每个聚类包含的像素数也越多，相机重建时也需要考虑更多的光线，聚类相机生成的深度图像也会更大这导致了更大的聚类相机深度图像和更长的光线求交时间，这些都导致了在高分辨率下帧率的下降（表 16）。但是反射绘制总绘制时间并不会随着生成图像尺寸增加而线性增加，这是因为生成图像的尺寸变化并不会改变反射场景几何体，所以屏幕空间下的反射聚类数目是一样的，只是每个聚类要包含更多的

反射像素。

输出图像生成效率与聚类生成的深度图像大小关系不是特别大。强行改变深度图像尺寸为本章算法计算值的两倍和二分之一都不会对绘制帧率有太大改变（表 17）。在表 14 中只有最后两个步骤和聚类相机生成深度图像相关。步骤 3. b 对聚类的数目即聚类相机深度图像张数更加敏感，而对生成图像的尺寸不是特别敏感。步骤 3. c. i 相比于步骤 3. b 所需时间少，即使生成图像增大意味着更多的搜索像素和搜索步长，求交操作也对整个绘制效率影响有限。漫反射三角形数目影响步骤 3. b（表 18）。漫反射三角形数目的减少会逐步加速整个绘制效率直到步骤 3. b 的开销与整个开销占比忽略不计。光线跟踪在起居室场景比盥洗室场景慢而本章方法在两个场景表现相差不大（表 13），前者的原因是在于起居室场景三角面片数目多，这限制了光线跟踪方法的性能，而后者的原因是这两个场景中的漫反射三角形数目较小不能使本章方法得到明显的效率变化。

本章算法相比光线跟踪方法有较好的场景规模扩展性，即随场景复杂度的增加相比光线跟踪方法性能损失更低。本章算法中屏幕空间像素聚类、聚类相机的创建和反射光线与聚类深度图像的求交操作这三步的时间性能更依赖于输出图像的分辨率。正因为如此，生成简单场景（如康奈尔盒子）的高分辨率反射绘制输出图像时光线跟踪算法处理会更快。虽然生成聚类深度图像这一步和场景复杂度有关，本章所用的两个不同面片规模场景的差异不会明显影响到深度图像的创建的时间（表 13），这是由于聚类深度图像生成步骤有 GPU 的硬件绘制单元的充分支持。而不同规模的场景对光线跟踪所需的加速结构（本章用的 BVH）创建有很大的影响。比起盥洗室场景（260ms），起居室的加速结构创建过程用了显著更长的时间（910ms）。本章提出的算法更依赖于场景中的聚类数（表 15），而聚类数目在表 13 所用的四个漫游路径变化不大，聚类的数目变化情况如表 19 所示。总之本章算法的绘制效率对场景复杂度有较好扩展性的原因是因为 GPU 绘制聚类深度图像相比于光线跟踪加速结构的创建更不依赖与场景复杂度。

表 19 表 13 所用四条路径中的聚类数表

场景	盥洗室		起居室	
聚类数平均值	148	120	123	107
聚类数最大值	195	162	194	158

在反射物的细节复杂程度上，本章算法能很好地处理镜面反射物，而镜面反射物在人造场景中经常遇到。有着高曲率的复杂反射物对本章方法和光线跟踪都是有难度的。对于本文方法，复杂反射物意味着更多的聚类数目。光线跟踪算法必须为每个高曲率的

反射物表面像素发射数十或者数百根反射光线对被反射几何体进行过滤。本章方法则可以很轻易地通过聚类深度图像的不同细节层次化表示 LOD 对被反射场景进行过滤。不管是本章方法还是光线跟踪方法都是为了能产生连贯反射的大反射物表面设计开发的；对于拥有非常高细节的反射物，环境光贴图方法仍然是一个选择，因为高细节的反射物的反射是否精确很难判断，这样就规避了精确反射的计算。

光泽反射

光泽反射和镜面反射绘制的区别在于光泽反射绘制包含深度图像和多根光泽反射光线的求交过程。除开更多的反射光线需要计算之外，光泽反射中单根反射光线与深度图像相交也会比光泽反射所用时间长。这是因为单孔相机可以很好地模拟镜面反射中每个采样像素只生成一根反射光线这种情况，而在光泽反射中，反射光线比起聚类相机光线相比更加分散，这样导致光泽反射光线在聚类相机生成深度图像上投影更长，搜索距离也 longer。而且光泽程度越高（即表明材质更粗糙），反射光线的分散程度越高，光线求交的开销也越大。光泽反射的时间效率在表 20 中给出。相比于镜面反射，光泽反射效率要低，但是还是保持了对光线跟踪方法的效率优势。

表 20 图 30 和图 32 中光泽反射绘制的时间效率

光泽程度	每个反射像素生成光泽反射光线数	求交时间[ms]	帧率 [fps]	
			Optix	本章方法
偏镜像	8	112	1	6
	49	651	0.4	1.4
偏粗糙	8	163	1	5
	49	814	0.4	1.1

4.7.3 算法实现

第一遍实现漫反射着色和屏幕空间像素生成的绘制（章节 4.2，步骤 1）是由 GPU 上的着色器完成。因为反射像素聚类（步骤 2）算法只涉及像素自身信息，而不需考虑到邻接像素信息，也是由 GPU 着色器实现。

像素聚类相机构成（步骤 3. a）在聚类中心点计算、拟合平面法线计算和像素投影轴对齐包围计算需要进行关键区的并行写入操作，所以本章算法用到 GPU 端的 CUDA 原子操作计算和共享存储器加速方法。本章方法先尽可能地使用标准写入操作来代替原子操作得到一个好的起始值，再通过这个起始值来减少原子操作耗时。比如，当搜索一串数组中的最大值时，可以先用标准写操作得到数组中的一个比较大的值，再用这个比较

大的值作为初始值来执行原子化最大比对写入操作。又因为这个比较大的值只是比整个数组中一小部分数值小，这样最大值计算只需要少量的原子更新操作，这样相比于用数组中的第一个值作为起始值的情况节省了很多原子操作运算量。

聚类深度图像绘制（步骤 3. b）和最后生成反射像素颜色的反射光线与深度图像求交操作（步骤 3. c）是在 GPU 上完成的。步骤 3. b 使用了基于物体的视景体剔除方法，步骤 3. c 中平面物体产生的反射光线与深度图像的求交能迅速得到结果，这是因为这类光线在深度图像上的投影起点和终点都收敛为一个点（见章节 4. 4，聚类相机的构建）。能正确（没有遮挡误差）和有效地（聚类相机的创建方法保证大部分捕获到的反射场景都会对最后反射效果绘制所用）处理平面反射物体反射是本章方法的一个优势，因为在人工环境里大部分的反射物体是镜面物体。

表 21 表 13 所用四条路径中平面反射物像素与反射像素数目比值表

场景		起居室		盥洗室	
平面物体反射像素 [%]	平均	41	33	17	20
	最大	52	51	33	34

4.7.4 算法讨论和局限性分析

本章反射绘制方法使用了几种模拟方式。首先场景反射光线是通过聚类后拟合成传统小孔相机的观察光线进行近似。聚类模拟地越细，生成的小孔相机模型越能模拟反射光线。聚类相机生成的图像不是直接用来形成反射，而是间接地生成正确反射绘制效果。换言之，反射不是由被反物场景经过图像扭曲得来，而是通过反射光线与聚类深度图像求交正确计算出被反射物体采样点在生成图像上的位置。但是，小孔相机模型不一定能捕获反射所需的所有采样点，这样就导致了上述章节讨论的遮挡误差问题。

其次，被反射的场景图像由深度图像近似表达，这导致了在图像空间又需要对几何体和颜色进行再采样过程。这种模拟的表达结果误差可以通过增大深度图像的分辨率来减小。最后，通过环境贴图去模拟小的反射像素聚类的反射绘制问题（见图 28 中的处于反射物边缘的反射像素）。

本章方法最适合处理镜面反射这类每个反射像素生成一个反射光线的情况。光泽反射生成的反射光线是在镜像反射光线轴对称瓣分布，光泽反射光线不能很好地由传统相机进行模拟。而且光泽反射生成需要与深度图像进行费时的多次求交操作。

本章的方法在每次输出图像绘制时需要对每个聚类绘制生成对应聚类深度图像。这些绘制操作因为其目的是生成聚类深度图像来记录被反射场景，所以不可避免。除了场

景中的物体被不同反射物的反射,和除了相邻的两个聚类边缘为了保持反射连续性发生的一小部分重叠这两种情况,聚类生成的深度图像是非冗余的。

聚类深度图像绘制效率随着漫反射场景复杂度变化的稳定性问题也是影响输出图像生成效率的一个重要因素。通过对漫反射场景进行层次化剖分是一种可能的优化方案,但是这种方法不适合全动态场景。另外一种策略是改变聚类方法,通过引入重要性的概念避免不必要的碎小聚类生成。基于 KD 树的划分方法^[89]能获得更小的聚类数目。但这还需要调查这种方式带来的效率的提升能否补偿它耗时的聚类算法时间开销。我们同时也会调查是否能够通过把聚类相机分组使得每个组里面聚类相机的视景体都是分离的。这样通过一次绘制就可以在一张组合深度图像上绘制出这一组相机所需的深度图像,提高深度图像绘制效率。

本章提出的方法支持二阶反射光线的绘制。当反射光线相交到具有反射材质的场景时,通过环境光贴图或直接用材质漫反射成分颜色作为反射光线光照值。高阶反射光线很有可能在凹体上通过互相多次反射产生。我们可以通过两种方式处理高阶反射情况,一种方法是不管物体凹凸,反射光线按照常规设定与聚类深度图像进行求交,进而忽略掉高阶反射的情况。另外一种方法是通过检测聚类在聚类相机近平面前面还是后面来检测这个几何体的凹凸情况,进而去检查反射光线的可能交点是相交到聚类自身还是其他物体,当相交到聚类自身时用聚类本身漫反射成分作为反射光线光照值。两种解决方法都不能提供正确的高阶反射光线绘制结果,但是第二种方法可以在额外开销很小的情况下提供稳定的高阶反射效果。

本章的方法最适合于镜面反射物体的反射绘制。物品表面材质越光滑,生成的反射角锥越小,反射光线越聚拢,反射光线求交场景越能有效地由深度图像近似表达,光线与深度图像的求交所需要的搜索步长越少。当反射物表面越粗糙时,反射光线的连续性会降低,这时聚类小孔相机会扩大视角甚至会变成全景相机模型。因为反射光线和法线是不直接相关的,按照聚类法线进行相机建模也提升不大。本章验证了基于聚类相机生成深度图像的反射绘制方法能在镜面反射上取得很好的反射效果,同时也能处理高光滑反射程度的反射绘制。

相比于使用绘制爆炸图[70]方法来模拟被反射顶点投影到屏幕空间的方法,本章的方法在处理复杂反射物表面和多次反射绘制时有优势。本章的方法能够处理复杂表面物体是因为本章方法并不是尝试去提供反射顶点投影的常数解,反之,本章是把反射物体

映射到深度图像上，接着投影反射光线到深度图像上搜索求解。深度图像与投影反射光线的求交是一个搜索的过程，比常量的顶点投影爆炸图方法更加耗时，但是它可以进行复杂物体表面的正确绘制。反射点的求交限制在反射光线在深度图像上的一维投影，它的时间复杂度更深度图像的尺寸有关。

相比于基于图像的方法，本章的方法在镜面反射上更有优势因为镜面反射需要很高的存储图像分辨率，这造成光场方法应用在镜面绘制进行搜索时开销过大。本章的深度图像的方法很好地压缩了被反射物体数据。相比于其他的被反射场景替代方法，本章的方法优势在于只模拟了当前帧所需的部分被反射场景信息，并且模拟的可靠程度符合当前帧的需求。相比于环境贴图方法，本章的方法更加准确。相比于被反射场景的视无关模拟方法，本文方法更加有效，能够支持动态物体的反射绘制。

本章的方法在复杂反射场景中支持实时的反射绘制。对于帧率是主要考量标准的场景绘制应用，可以只在有反射绘制材质的场景子集里面使用本章方法。随着 GPU 光栅化绘制技术的发展，因为本章能很好地迁移到 GPU 流水线上，本章方法相对于光线跟踪方法的优势还会扩大。

本章的方法表明，如今基于深度图像的绘制应用范围十分广泛。它不仅可以作为视空间下场景的模拟，还可以通过聚类相机捕获二阶光线相交的场景。在本文中，二阶光线是指由场景反射材质物体生成的反射光线。在未来工作中，本文方法可以扩展到其他类型的光线上。本章也可以通过用更强力的可以模拟发散光线的非小孔相机模型代替用来模拟被反射场景的常规小孔相机来缩减聚类的数目。

4.8 小结

本章针对被反射物体难以由深度图像采样表示的问题，提出了基于聚类相机深度图像表示反射场景的反射绘制方法。本章方法首先通过屏幕空间像素聚类的方法对反射像素进行视空间划分，然后由聚类相机生成该聚类反射场景的深度图像，最后再由反射光线与聚类深度图像的求交计算反射光线的颜色值。本章方法的聚类相机方法能按照聚类几何体的表面特性按需生成拟合该聚类反射光线的相机模型，并且通过反射光线和深度图像的相机保证了高质量的反射结果生成。实验结果表面：本章方法可以在实时绘制情况下生成高质量的镜面和光照反射效果图像，生成图像质量与光线跟踪方法没有明显差异。本章算法不需要预处理，能支持动态场景的绘制，在动态场景下效率能达到光线跟踪反射绘制方法的 10 倍以上。

第五章 基于反射深度图像时间相关性的反射绘制复用方法

时间相关性 (Temporal Coherence) 复用方法是一种将已计算好的深度图像映射到新视图的常见场景快速绘制方法。常用的基于投影的时间相关性复用方法是假设像素的光照特性在不同视点下基本保持不变 (视无关) 的。然而反射的物理特性决定了反射的绘制会根据入射光线的变化而发生变化 (视相关) 的, 因此现有方法不能直接用于反射绘制。本章提出一种通过像素的场景采样点拟合平面建立反射像素在不同视点下的映射投影方法。此方法适合于含有反射的深度图像像素复用, 使用此方法能减少反射绘制的总时间开销, 提高绘制效率。

5.1 引言

使用时间相关性 (Temporal Coherence) 来分摊绘制开销、提高绘制效率的想法已经广泛地应用在多种绘制特效例如全局光照^[87]、软阴影绘制^[82-83]和环境光遮挡的^[85]绘制之中。这种方法的基本思路是复用已经计算好的帧中内容生成新的帧中内容。因为通常复用已经计算好的记录在深度图像中的光照信息比重新进行光照计算开销要少, 所以基于时间相关性的复用算法很有效。这种方法成立的前提假设是像素的值只依赖于场景的设定、几何体附着的材质和光源的配置, 而不会随着视点的改变发生明显的变化 (即视无关的)。这样关键帧深度图像存储的光照信息就可以在不同的视点下进行深度投影复用。这种基于深度的投影方式对于生成包含视无关信息的输出图像是有效的, 但是, 它不适合生成含有反射材质的场景。原因是反射是视相关的绘制特效, 反射的绘制会随着观察相机的变化而变化, 正确的反射结果需要对每个反射像素生成的反射光线和绘制场景进行求交取得着色光照值。所以, 反射不能直接按视无关成分进行模拟。直接使用基于深度的投影方法进行反射图像生成会产生较大的误差。如图 34 和图 39 所示, 直接使用深度投影方法生成图片与物理真实光线跟踪方法生成图像不一致。

本章提出一种新的使用时间相关性计算反射在不同视点之间映射的方法。本章的方法扩展了深度图像的定义, 提出反射深度图像概念, 在深度图像基础之上, 额外存储光线跟踪引擎计算的反射颜色和反射光线传播距离。这使得深度图像不仅可以作为复杂场景的视无关信息的近似表示, 也能表示如反射等视相关信息。本章的基本想法是, 镜面

反射成像随着视点的移动方式而在反射物表面上移动（如图 35，当相机位置从 E_{old} 移动到 E_{new} 时反射成像在关键帧中从像素 s 点移动到像素 s^* ），镜面反射的绘制仍然存在时间相关性，这样关键帧的反射仍然可以被相邻的输出图像生成复用，从而分摊后续生成帧反射计算的开销，提高绘制效率。本章方法使用偏移像素缓存记录关键帧反射像素在新视点下的偏移值。这样新的反射绘制生成图像可以由关键帧中像素按照偏移像素缓存中对应位置的值进行偏移获得。对于在新视点下在关键帧中没有索引的像素，或者说索引的像素的可靠程度太低，本章算法重新使用光线跟踪方法计算这些像素的反射。相比于全部使用光线跟踪进行反射绘制的方法，本章提供的方法大幅度地（盥洗室 77.1%，起居室 82.4%）减少反射绘制开销，提高了绘制效率。

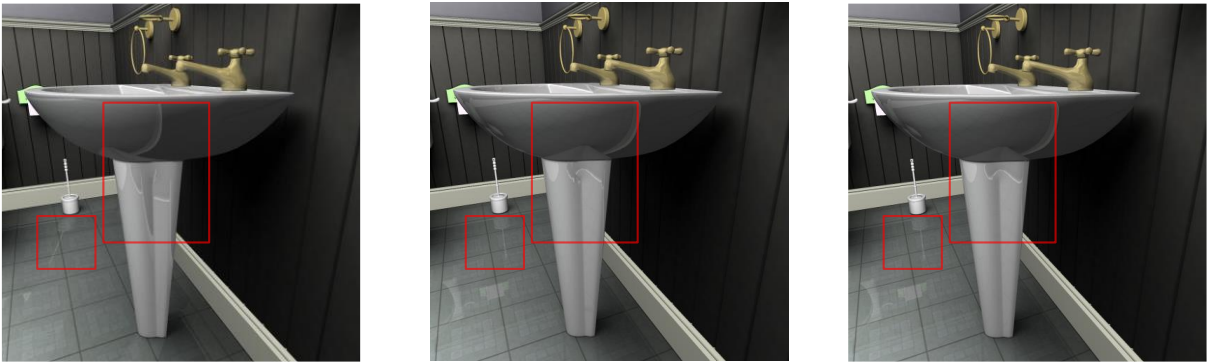


图 34 直接使用深度投影方法（左），本章方法（中）和光线跟踪方法（右）生成的反射绘制图

本章节的主要贡献在于：

- 1) 提出反射深度图像。反射深度图像对深度图像进行了扩展，它不仅能表达相机观察光线采样的场景，还能表示被反射的场景。
- 2) 本章提出一种帧间反射距离计算方法。帧间反射距离计算方法适用于计算反射深度图像中反射像素度量当入射光线从新视点发出时，该像素所记录的被反射物体用反射定理拟合时产生的误差距离。

本章的反射绘制方法跟具体使用的光线跟踪技术无关，任何可以返回反射光线距离和值的光线跟踪算法（包括上一章基于聚类深度图像的反射绘制方法）都可以使用本章方法来减少反射光线绘制开销。

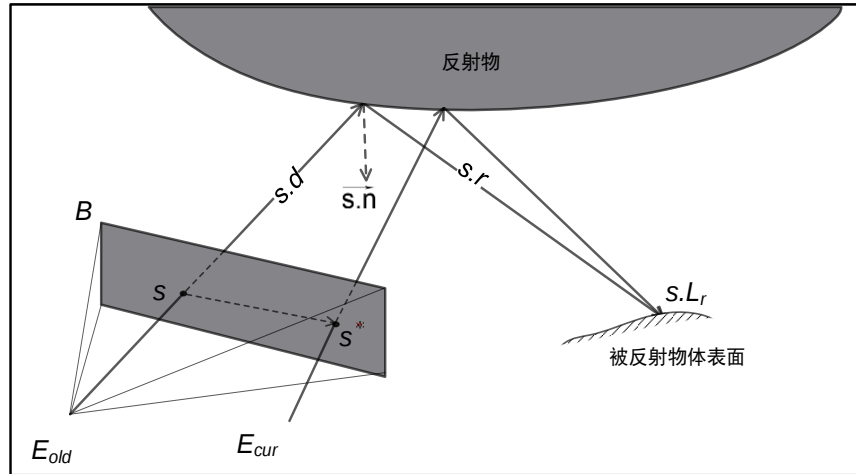


图 35 反射像素随视点移动示意图

5.2 算法整体流程

时间相关性反射效果生成方法使用如下流程进行输出图像的生成。首先，场景的局部漫反射照明通过一次场景绘制生成。其次对于关键帧，输出图像的反射部分首先使用光线跟踪方法进行生成并和其他属性一起存储在反射深度图像中（章节 5.3），然后再经过混合操作生成输出图像；对于非关键帧，本章算法使用基于时间相关性的反射投影算法（章节 5.4）将关键帧中反射投影到非关键帧上，在后处理阶段，反射投影失效的部分重新使用光线跟踪方法重新计算反射，并与漫反射成分合成生成输出图像。本章算法整体流程如算法 8 所示。

算法 8：使用反射投影算法生成最后图像

Shade an image with diffuse components from the output viewpoint

//漫反射成分着色生成

if current rendering frame is the key frame then

//如果是关键帧则记录反射深度图像，最后进行输出图像混合

 Generate reflection buffers

 Blend the output image with the reflection buffers

else

//如果是非关键帧，则使用基于时间相关性的反射投影生成反射，再执行输出图像合成

 Temporal coherence-based reflection reprojection using the key-frame reflection buffer

 Output image composition

end if

5.3 反射深度图像的生成

本章引入反射深度图像的概念。反射深度图像是一系列二维纹理贴图，这些二维纹理贴图既能作为相机观察光线求交场景也能作为反射光线求交的场景的近似表达。作用于每个像素的反射投影映射方法将会应用在这些纹理上生成新的视点下反射图像。反射深度图像中的纹理贴图包括一个法线贴图，一个深度贴图，一个材质贴图和一个反射贴图。

给定一个由三角形建模的具有漫反射材质和反射材质物体的场景 S ，关键帧绘制时观察相机视点位置 E ，反射深度图像 B 的像素 s 含有下列信息

B.1:像素法线 $s.n$ 和像素深度 $s.d$;

B.2:用1表示完全镜面，用0表示完全漫反射材质的反射镜面反射因子 $s.f$;

B.3:用RGB成分记录被反射面元的光照颜色 $s.Lr$;

B.4:反射光线从反射物到被反射物的漫游距离 $s.r$;

除开传统深度图像里面常用于做漫反射延迟绘制的成分如**B.1**之外，其他成分例如**B.3**和**B.4**可以用反射定律重建出被反射场景的场景近似表达。在反射投影完成之后，生成图像按照**B.2**的值对投影后的反射和漫反射成分进行混合。**B.1**和**B.2**成分是通过一次绘制整个场景得到。法线信息和深度信息可以由采样三角形顶点插值得到。反射深度图像中最后两个部分**B.3**和**B.4**可以由大多数的光线跟踪方法或框架生成。反射深度图像作为关键帧相机观察光线和场景表面反射光线采样的场景近似表达，在生成关键帧的时候进行更新，而在生成非关键帧反射的时候作为输入被用来投影生成当前输出图像。

5.4 基于时间相关性的反射投影

基于深度图像的图像投影算法是一种常用的帧间复用算法。这种算法的基本思想是通过建立从关键帧到当前帧的时间相关度的复用映射关系来寻找当前输出图像的像素在关键帧对应像素索引位置。这种映射关系表现为从关键帧索引位置到当前帧像素位置的偏移矢量。

对于视无关成分的映射函数，其映射关系计算的出发点是基于两个帧之间的像素值不会有太大的改变，这样通过反投影出像素的三维坐标就能建立像素在不同帧之间的映射关系。给定当前相机视点 E_{cur} ，对于反射深度图像中坐标为 (x,y) 的像素 s ，该像素

采样点的三维坐标 W 可以将采样点深度反投影到世界空间由下式得到。

$$W = \text{unproject}(x, y, s.d) \quad 7$$

本章定义几何采样点为三维点 W 和它采样点的法线 $s.n$ 所表示的观察光线与场景相交的局部拟合平面。在输出图像生成时只需要将这一像素点按照当前相机的视投影矩阵 (Model View Projection Matrix) 进行投影即能找到在当前生成视点的对应像素 s_{cur} 。投影的可靠程度由 s_{cur} 和 s 的深度差是否在给定误差阈值 e 内判断: $|s_{cur}.d - s.d| < e$ 。为了应对多个像素对应到新相机下同一个像素的情况, 使用深度排序检测并剔除多个像素中不是最靠近新视点的像素。

对于视相关成分的映射函数, 因为视相关光照成分会根据着视点变化而变化, 视无关映射函数的假设条件是不成立的。用反射举例: 反射材质根据反射定理镜像延长了入射光线, 若入射光线反射变化, 反射光线亦会发生改变。如果视相关成分只根据像素深度进行投影会导致错误的投影结果。

本章提出一种基于时间相关性的反射投影映射方法。这种方法思想是广义的基于深度帧间映射思想: 反射像素所对应的三维虚拟点虽然不像视无关照明帧间复用算法假定的是个固定点, 在相机位置发生移动时三维虚拟点的投影值能够在反射物表面发生移动。考虑关键帧中的反射成像, 其中许多反射的成像也会在相邻帧中出现。本章的方法目的就是去寻找这种反射成像在帧间的联系。本章首先通过寻找关键帧中每个像素的反射采样点在入射光线由新相机发出时在关键帧中的最佳匹配位置, 生成并存储关键帧像素偏移矢量(章节 5.4.1), 然后把关键帧中的反射像素按照所记录的偏移量向它对应的匹配点进行偏移生成反射中间图像(章节 5.4.2), 最后扭曲反射中间图像生成在新视点下的输出图像, 步骤如下:

1.反射像素匹配。对于关键帧反射深度图像 B 中的每个像素 s , 这一步是决定当视点移动到新视点下 B 中每个像素的最佳反射颜色匹配像素位置 s^* 。之后, 把模拟反射移动过程的偏移矢量 s 到 s^* 记录在偏移矢量缓存中。

2.反射偏移。给定偏移缓存, 将反射深度图像中的像素按照偏移缓存中的值进行移动, 生成偏移后中间反射图像。

3.反射图像扭曲。使用基于深度图像的投影方法将中间反射图像投影到新视点。

图 36 展示了基于时间相关性的反射投影处理流程。A: 屏幕空间物体的镜面反射相交场景和当前下视点采样场景存储在反射深度图像中。B: 经过最佳像素匹配后, 每个反

射像素到最佳匹配像素的偏移值记录在偏移矢量缓存中。C：反射深度图像上的像素按照偏移矢量上存储的偏移值进行偏移生成反射中间图像。对于欠采样造成的偏移值可靠程度不高区域和由于运动时差造成的间隔区域，图中用红色可视化显示。D：最后用传统基于深度图像投影方法将反射中间图像转换到新视点。

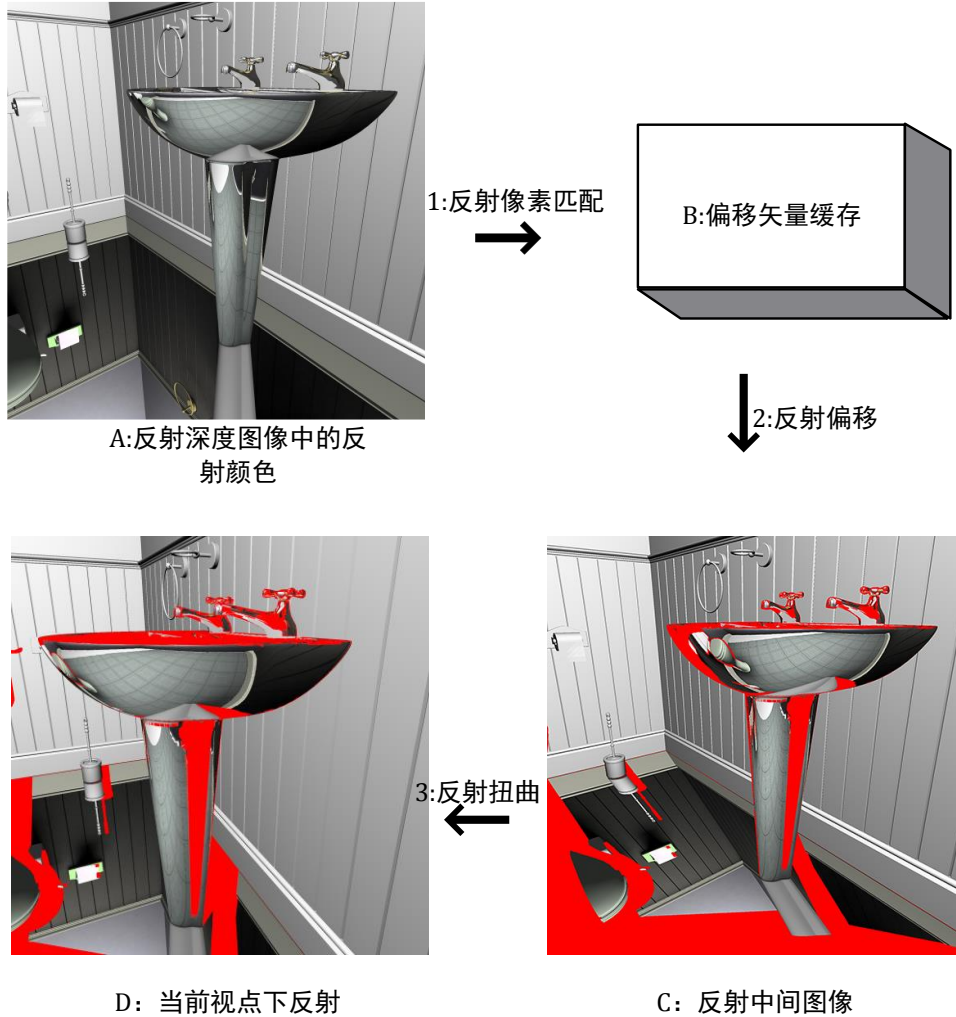


图 36 基于时间相关性的反射投影步骤图

5.4.1 最佳反射像素匹配

反射像素匹配过程决定当反射入射光线从新视点射入时反射深度图像中每个像素点到反射最佳匹配点的映射方式。给定相机在 E_{old} 时生成的关键帧，帧中像素 s 反射光线与场景相交点 R 可通过反射定理和 s 的三维坐标 W 由下式给出。

$$R = W + \text{reflect}(\overrightarrow{WE_{old}}, \overrightarrow{s.n}) \cdot s.r \quad 8$$

s 的最佳匹配点 s^* 定义为关键帧中反射深度图像上的像素点 s^* ， s^* 所在的拟合平面（位置 W_s ，法线 n_c ）最能满足关于新相机位置 E_{cur} ， R ， W_c 的反射定理

$\angle(\vec{n_c}, \vec{W_c R}) = \angle(\vec{n_c}, \vec{W_c E_{cur}})$ 。找到反射最佳匹配点意味着能建立关键帧中反射图像到当前帧反射图像的映射过程。为每个像素找到最佳匹配点是有难度的。因为关键帧反射深度图像近似表达了观察光线和反射光线采样的整个关键帧场景，而每个像素在新视点下都可以定义一个反射光线，找到最佳匹配点意味着需要进行反射光线与整个近似场景求交，但是这种方法太过耗时。另一种方法是通过近似场景的拟合解析式求解，但是由于复杂场景中的物体多样性，很难建立场景的解析表达式，所以这种解析解不是封闭解。对于最佳匹配点的求解，本章算法解决思路是在反射深度图像的有限空间中搜索最佳匹配点。本章的方法通过给定的一种用来描述在新视点下某一像素满足反射等式的距离函数来搜寻最佳匹配像素点。具有最小距离值的像素点定义为最佳匹配像素。本章基于如下考虑来设计这个距离函数：

- a. 距离函数要有普适性。距离函数要能支持凹体，凸体和平面反射物体的帧间映射。
- b. 距离函数的参数是局部的，只需要用像素的局部信息作为输入参数。
- c. 距离函数要收敛，最佳匹配点要有最小的距离函数值。

图 37 所示给定反射深度图像中的像素采样点 s ，相机由关键帧位置 E_{old} 移动到新位置 E_{cur} ，被反射点 R 的位置可以由反射方程与存储在反射深度图像中的漫游距离算得。关于像素 s_n 的距离函数定义 s_n 能模拟从相机 E_{cur} 发出，反射到 R 的光线的能力。本章算法决定了当相机发生移动时有最小距离值的像素即为 s 的最佳匹配像素。本章的算法使用 E_{cur} 到从虚拟点 M 穿过 W_n 的射线作为最短距离值。虚拟点 M 是将点 R 按照由 W_n 和 $s_n.n$ 定义的采样点拟合平面镜像生成。 s 点关于某一像素 s_n 的距离值 d 定义为 E_{cur} 到由 M 和 W_n 定义的射线的欧式距离。

给定匹配距离函数定义，在整个图像空间寻找最佳匹配点是有难度的，这是因为整个屏幕空间的反射物体是由各种形状构成，构建反射物的几何解析表达式的方法是不现实的，所以直接代数解算求出最佳匹配点不可行。本章使用局部搜索算法寻找拥有最小距离值的像素。局部搜索的方法迭代从当前搜索像素移动到该像素邻域中距离值最小的像素，重复此过程直到搜索到最小值或者离开搜索空间。局部搜索方法假设距离函数在关键帧反射物体投影区域内能单调收敛，否则，局部搜索算法可能陷入没有更好可能解的局部最优解之中。但是，在应用的过程中，即使在有几何细节的非连续性反射物表面，使用者也很难发现这种错误解。因为这个原因，这种局部搜索方法能形成稳定的高质量

反射图像。本章所用的搜索算法如算法 9 所示。注意当前场景相机位置离关键帧相机位置越远，需要的搜索步骤越多。为了提高搜索效率，搜索点的起始值可以由存储在上一帧偏移缓存中的搜索终点代替来节省搜索步长。反射物的边界由反射深度图像中像素之间的深度差决定。

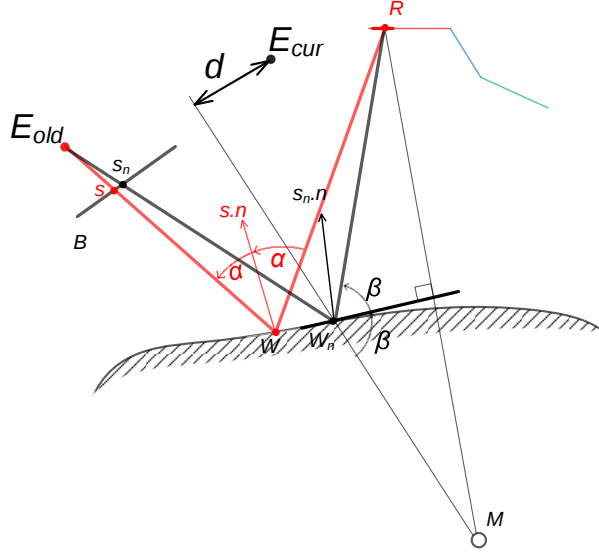


图 37 计算 s 关于 s_n 匹配距离示意图

算法 9：最佳像素匹配局部搜索算法

输入： 反射深度图像 B ，反射深度图像生成时视点 E_{old} ，新视点 E_{new} ，之前帧存储的偏移矢量缓存 O_p

输出： 更新后的偏移矢量缓存 O_c ，记录新视点下每个像素的最佳匹配像素偏移向量

for each reflective sample s at location (u, v) in B **do**

Fetch sample's depth $s.d$, normal $s.n$, reflection depth $s.r$ at (u, v) from B

Surface position $W \leftarrow \text{backProj}(s.r, u, v)$

Reflected position $R \leftarrow \text{reflect}(s.n, E_{old}) \cdot s.r + W$

Fetch sample's previous offset o at (u, v) from O_p

$s_c \leftarrow (o \neq \text{INVALID}) ? s + o : s$

//如果前一帧的偏移值可靠，直接使用前一帧的偏移值

$\text{formerD} \leftarrow \text{distance}(s_c, R, E_{cur})$

while s_c is inside B **do**

$N \leftarrow \{ \text{a } 3 \times 3 \text{ hollow square neighborhood centered on } s_c \}$

```

//在当前搜索点的  $3 \times 3$  邻域中搜索匹配距离最小像素点
 $(s_{min}, d_{min}) \leftarrow \min \forall s_n \in N \text{ distance}(s_n, R, E_{cur})$ 
if  $d_{min} > \text{formerD}$  then
    //找到局部最优, 进行可行判断
    if  $s_c$  is inside the reflector's silhouette then
        record  $s_c - s$  at  $(u, v)$  in  $O_c$  return
    else
        record INVALID at  $(u, v)$  in  $O_c$  return
    end if
end if
formerD  $\leftarrow d_{min}$ 
 $s_c \leftarrow s_{min}$ 
end while
//如果没有搜索到最佳匹配点, 设为不可靠
record INVALID at  $(u, v)$  in  $O_c$ 

```

5.4.2 反射偏移

反射偏移这一步决定反射深度图像里面的数据如何被索引、复用然后映射生成反射中间图像。本章使用前向投影（Forward Reprojection）方法直接根据偏移缓存中的值把反射深度图像中的每个像素移动到偏移后的位置形成反射中间图像。当前大多数前向投影方法需要像素映射方法用关键帧像素去填充新图像。大体上前向投影主要有两种像素映射方法：（1）像素溅射方法（Splatting），即不用深度缓存上的邻接信息而用一个核函数溅射像素采样点的方法。（2）网格方法（Grid），即绘制时使用像素点之间的显式邻接信息构成的网格。

像素溅射的方法是出于避免采样点之间的邻接性发展起来的。这种方法的困难之处在于需要很精确的估计采样点在输出图像的影响区域才能既能避免相邻像素之间空洞（holes）的又能避免超过投射区域边界的过度绘制。使用网格方法可以利用深度图像像素之间天然的连接信息进行网格生成。网格的绘制方法在四个相邻像素矩形之间没有很大的深度差情况下构成两个绘制三角形网格进行绘制。随着GPU运算能力的大幅增加，由四个像素采样点构建出两个三角形的方法可以非常有效地完成。

本章使用网格方法进行反射前向投影，其原因是应用网格的连接信息能实现密闭的反射扭曲图像，而像素溅射方法中的空洞在当漫游相机朝反射物移动的过程中会更加的明显。更加糟糕的是，像素溅射方法形成的空洞会因为运动视差的问题被其他的被反射物体填充。这样即使是像素方法常用的补洞算法如层次填充^[57]也不适于应用在这种情形。如图 38所示，左图表示反射深度图像存储被反射物体采样。中图表示像素溅射方法会产生空洞，而且空洞会被其他物体填充，图中的点对应反射深度图像中存储的像素采样点。右图表示使用网格方法能连接采样点解决这一问题。本章使用和反射深度图像相同分辨率网格生成反射中间图像。网格的每个节点（对应于反射像素的像素中心点）按照偏移缓存里存储的值进行偏移后绘制出反射中间图像，这样反射深度图像中附带的反射信息就会自动偏移到适合新相机的位置。而且，网格方法可以用传入 $s.d+s.r$ 作为绘制深度的方式解决多个反射局域映射到相同位置而产生重叠的问题（见图 36，马桶刷和背景墙重叠）。

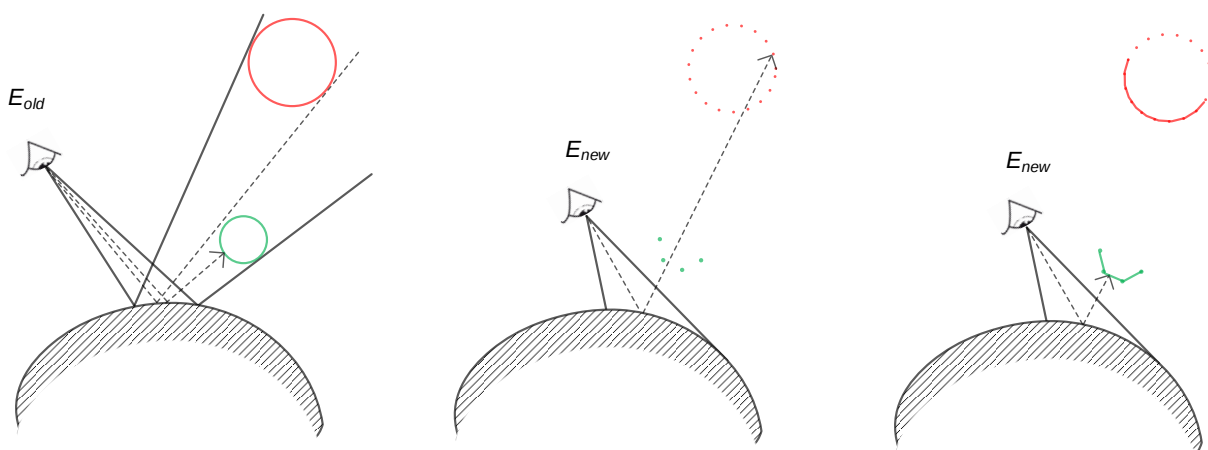


图 38 像素映射方式示意图

反射中间图像生成时由于像素网格之间的深度不连续性在不同偏移像素下会形成间隔，这些间隔区域和由于产生不了可靠像素偏移值生成的区域，会额外设置一个可靠比特位指示这个像素需要重新计算反射。如果偏移的网格面积扩大比超过一定的阈值以致于外插值形成的反射颜色的可靠程度太低，本章算法也会认为这个网格形成的像素不可靠。

5.4.3 生成图像合成

反射中间图像表明了当视点发出改变后，反射颜色根据新视点位置在反射物表面的移动生成的中间图像。一旦反射中间图像生成完毕，传统的基于深度的投影方法就用来

把反射中间图像投影到新视点下。这个过程中由于运动误差造成的遮挡区域和之前反射中间图像中的不可靠区域需要重新用光线跟踪方法计算反射颜色。像素生成颜色是由漫反射成分和镜面反射成分按照由相机视点、几何法线和像素反射层级计算出来的菲尼尔系数混合而成。

5.4.4 关键帧选取策略

关键帧的选取策略涉及到一个效率/质量的折中问题。就质量而言,关键帧中的反射信息可以投影到随后的帧中,但是反射的质量会随着相机距离的增加而产生误差,而且距离越大,这种误差越大(章节 5.5.2)。就效率而言,反射的复用比反射用光线追踪方法计算更有效率。因为反射的投影算法相比于整个算法时间占比较小(章节 5.5.3),整个输出图像的生成大体受限于重新计算反射部分的开销。在反射投影算法之后,本算法使用相对比较耗时的光线跟踪方法去计算反射投影算法中不可靠区域的反射,整个流程的瓶颈大体取决于需要重新进行反射绘制的像素数目(章节 5.5.3)。

反射绘制应用的质量/效率比可以通过设置反射重新计算的像素数量阈值来获得某个固定的增益比,由需要重新计算的反射像素比例决定关键帧是否更新。每一帧中,在反射投影算法完成之后,根据重新计算的反射像素和所有的反射像素的比值来确定下一帧是否是关键帧。如果这个比值大于一定阈值,在下一帧中就会重新计算反射深度图像作为关键帧反射深度图像,否则就使用当前的反射深度图像投影到当前帧进行反射的预测复用。本章使用这个图像生成策略来稳定保证绘制开销效率和绘制质量的性价比。其他方法例如^[73]中周期性的更新像素块里面的像素也能被用在反射复用策略上,但是因为反射是一种高频绘制效果,即反射的像在反射物表面变化很快,这种非均匀地更新方法就会使更新前后的像素在一个像素块里对比非常明显。本章中使用阈值来决定关键帧的方法适合于反射的绘制复用模式,而且它稳定、易于调整和容易实现。关键帧阈值的选择跟场景的配置如灯光、相机移动距离和想达到的反射质量相关。

5.5 实验结果和分析

5.5.1 算法实现

基于时间相关性的反射算法效果在两个有着镜面反射物体的室内场景(起居室场景: 28 万三角面片、15 万漫反射面片、13 万镜面反射面片,表 12 上列图;盥洗室: 9 万三角面片、4 万 4 千漫反射面片、4 万 6 千反射面片,图 34 和表 12 下列图)进行测

试。本章所有的评测都是基于 Intel(R) Core i7-4720 CPU, 16 GB 内存和 NVIDIA GeForce GTX 980M 显卡的 PC 工作站。本章使用 NVIDIA 的使用层级包围体 (bounding volume hierarchy, BVH) 作为 Optix 光线跟踪框架^[86]的求交加速结构。本章使用直接投影方法和纯粹光线跟踪方法作为本章算法的对比组。需要注意的是, 本章关注将关键帧反射图像投影到当前帧带来的反射绘制工作量的减小, 本章的方法和具体的光线跟踪技术无关, 任何可以返回反射光线距离和值的光线跟踪算法 (包括上一章基于聚类深度图像的反射绘制方法) 都可以使用本章方法减少反射绘制工作量。

本章的方法使用 C++ 框架实现。在反射深度图像的生成中, 前两个部分 B.1 和 B.2 是由 OpenGL 着色器生成, 后两个部分 (B.3 和 B.4) 是由 Optix 光线跟踪框架生成。反射深度图像是由两张有 RGBA32 位浮点数 4 通道的纹理存储。本章使用一个线程对应一个像素的 CUDA 进程执行反射像素匹配过程。反射偏移过程 (章节 5.4.2) 需要对每个投影三角形执行可靠程度判断, 这个判断过程是在绘制流水线中的几何着色器 (Geometry Shader) 中完成。在实际操作中, 本章使用 16 作为放大比阈值, 也就是说, 如果投影三角形的面积比投影前面积扩大了 16 倍, 这个三角形就被定义为不可靠, 需要重新计算该点反射值。

5.5.2 反射生成质量

正如图 34 和图 39 所示, 本章方法能生成与物理真实光线跟踪方法可比 (误差图见图 41) 的反射复用效果。图 40 用红色突出显示了本章方法需要用光线跟踪方法重新计算反射的部分, 右边图是左边关键帧图像投影产生的, 而在关键帧图像没有找到索引点需要重新计算的部分用红色显示。和直接投影的方法相比, 本章算法的反射投影部分和根据新视点生成的反射光线重新计算的反射部分没有明显的差异, 但是本章算法可以减少重新计算的工作负载量。在本章节里, 所有的图片和对照组图片都是按照 1024×1024 分辨率生成的。同样, 反射深度图像的分辨率也是 1024×1024 。

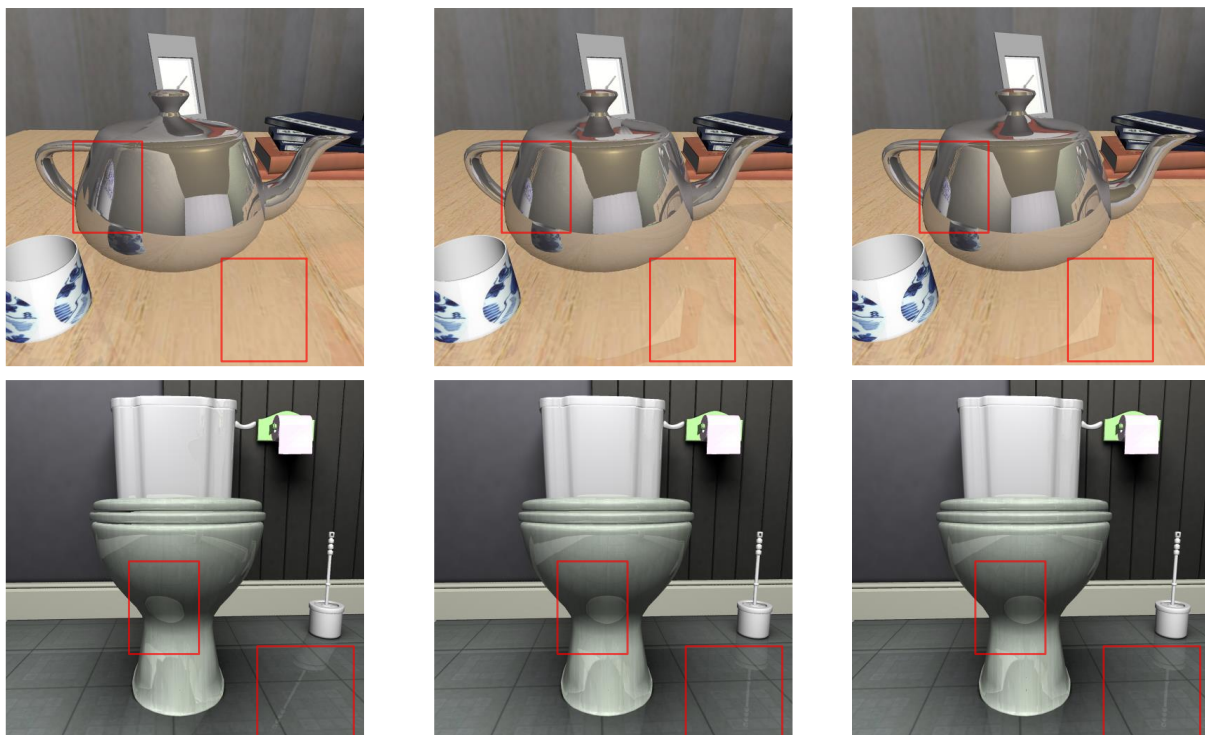


图 39 使用深度投影方法（左），本章方法（中）和光线跟踪方法（右）在起居室和盥洗室生成的反射绘制图

本章的方法能生成与光线跟踪方法可比的反射质量。图 41 展示了本章方法示例图像和光线跟踪方法的误差对比图和放大 60 倍后的误差对比图。示例图像设置下，直接使用深度投影方法会产生很高的可见反射误差。使用本章方法生成的反射绘制图平均光照强度误差的绝对值都较小，左中右三列分别是 0.13, 0.45, 0.12。光线强度误差是把 RGB 三通道的误差值从 $[0,1]$ 区间放缩到 $[0,255]$ 区间测得。误差的来源之一是由于在最佳像素搜索中没有使用全局最优而是局部最优点作为最佳匹配点。另外一个误差来源于被反射物体采样方法的不同，本章方法假定当前场景的反射在关键帧中也出现所以用二次线性内插值方法在关键帧中反射深度图像进行被反射物索引，而光线跟踪方法则直接用光线与场景求交得到反射光线的采样值。在实际过程中，由于分辨率的限制，如果视点有移动，当前帧像素的反射光线几乎不可能和关键帧像素的反射光线交点重合。

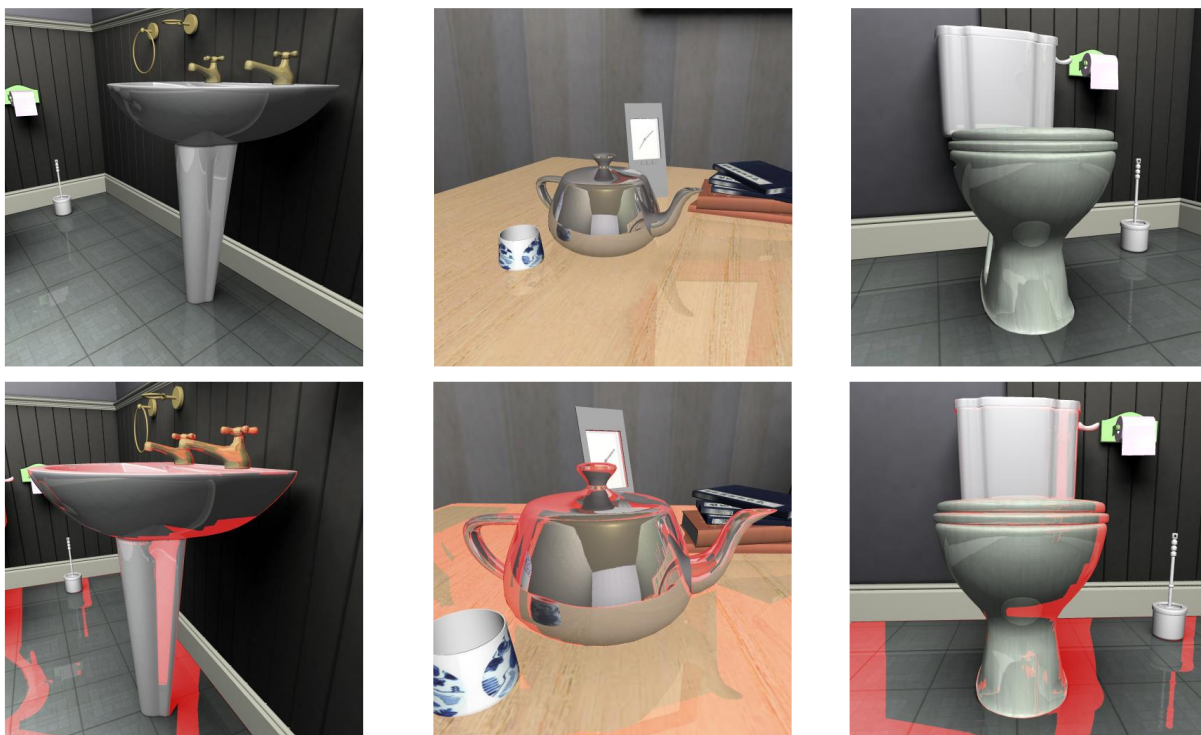


图 40 关键帧图像（上）和投影生成图像（下）

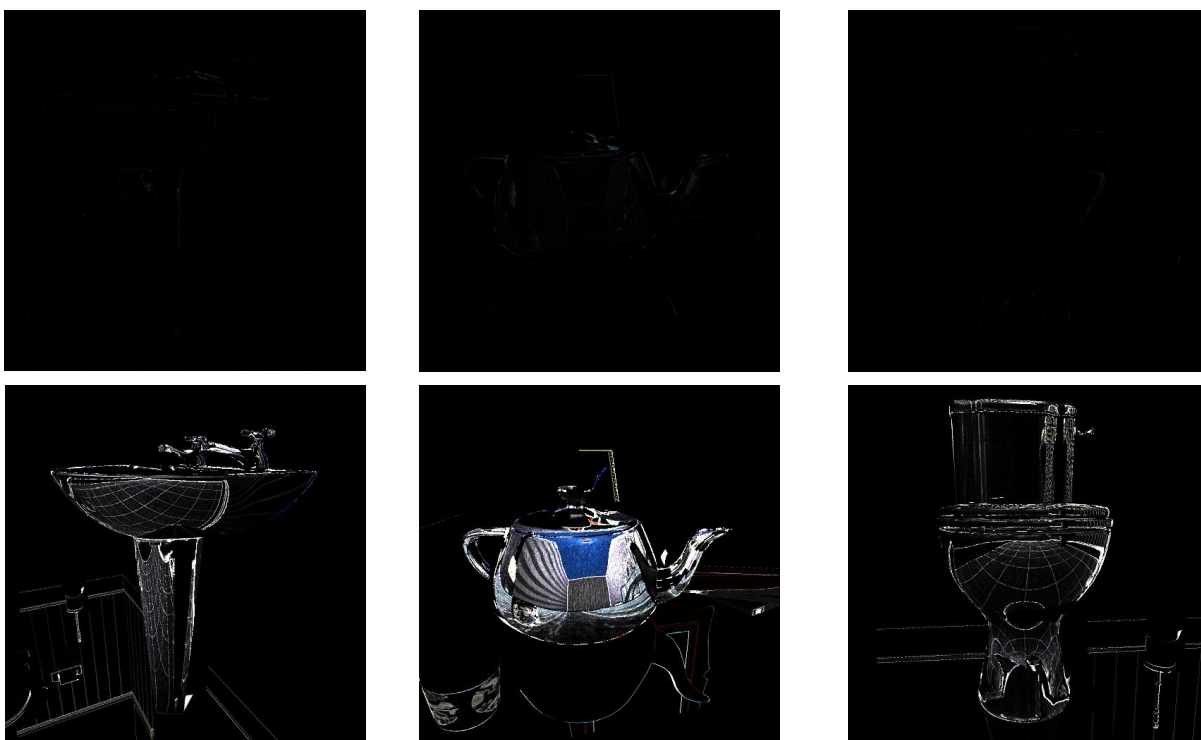


图 41 图 40 中对应输出图像与光线跟踪对比误差图（上）与放大 60 倍误差图（下）

误差还可能由于该被反射光线相交的场景几何体没有在反射深度图像中被捕获到而造成的遮挡误差产生。新视点所需要采样到的场景和关键帧中反射深度图像记录的近似场景不是完全相同，所以当前反射光线相交的几何体可能没有出现在关键帧反射深度图像中，而相交到反射深度图像中的另外一个物体。图 42 展示了这种情况：在关键帧

的反射中，架子的一个柱子藏在另外一个柱子后面。在反射像素匹配过程中，前面的柱子和背景墙上的像素都会根据相机的偏移生成不同的偏移矢量。在反射偏移过程中，前面柱子因为运动视差产生的空位会被背景墙像素填充。这样没有生成不可靠像素，光线跟踪方法就不会对这个像素重新计算正确反射，从而形成了遮挡误差。

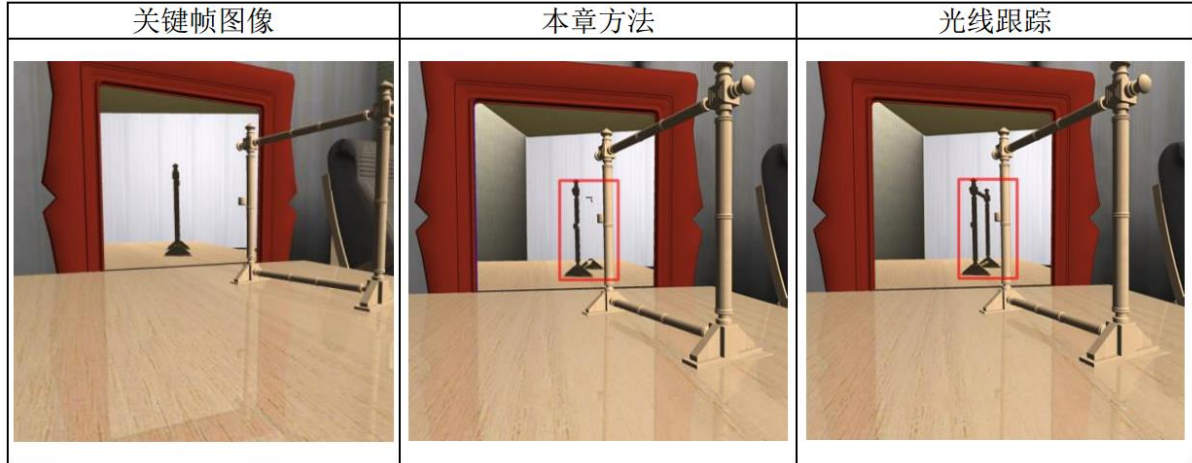


图 42 遮挡误差示意图，遮挡误差是由相交到墙但没有相交到后面柱子的反射光线所致

5.5.3 效率分析

表 22 给出了本章方法与光线跟踪算法的效率对比表。效率数据分别用盥洗室场景和起居室场景按照各自场景两条独立的路径绘制 500 帧的输出图像统计而得。本章算法使用 0.3 作为选取关键帧的阈值。也就是说，如果需要重新计算反射的像素数目高于场景反射像素数目的 30%，就重新计算反射深度图像并更新为关键帧，否则就用关键帧反射深度图像生成后续图像。表 22 中最后一个列展示了重新计算反射像素数目占比。关键帧间隔这一列展示了关键帧在整个绘制过程中的更新频率。本章的方法在盥洗室这个简单场景中获得了 1.8 倍的加速比，在更加复杂的起居室场景中能达到 3.7 倍的加速比。绘制过程中帧率的最低值和光线追踪方法可比，其原因是本章算法用光线跟踪方法更新关键帧，而关键帧更新中的主要开销就是来源于光线跟踪计算绘制反射。唯一不同的是关键帧更新时本章算法需要存储被反射场景信息在反射深度图像中作为被反射场景近似而光线跟踪方法不用存储。本章节拆分了各部分时间效率并展示在表 23 中。

表 22 本章方法和 Optix 的效率对比表

场景	反射像素 (%)		Optix(帧每秒)			本章方法(帧每秒)			关键帧间隔		重算像素比 (%)
	最大	平均	最大	最少	平均	最大	最少	平均	最大	平均	平均
盥洗室	55	45	23.0	13.4	18.5	54.9	13.0	33.4	51	20.6	20.8
	58	44	22.4	14.0	18.0	49.5	12.6	34.9	63	20.0	18.9
起居室	62	50	18.6	7.5	9.4	50.8	7.0	34.7	164	53.1	18.4
	58	44	12.8	6.4	8.8	48.4	6.6	32.0	92	24.2	18.5

表 23 展示按照毫秒统计的基于时间相关性的反射图像生成主要步骤平均时间。本章的主要贡献，基于时间相关性的反射投影时间开销只占整个输出图像生成时间开销的很小一部分时间。反射重计算过程非常耗时。这是因为虽然基于时间相关性的投影能复用并减少了 4/5 反射开销，但是剩下的需要计算的反射像素碎片化严重以致于很难使用光线跟踪方法中关于相邻光线的加速优势，另外光线跟踪应用框架启动时的常量开销值和由于 GPU 负载不够不能使得每个计算单元充分发挥性能都制约了光线跟踪方法在基于时间相关性的反射投影算法中反射重计算的效率。

表 23 算法不同步骤的平均耗时表

阶段	步骤	盥洗室	起居室
		时间[毫秒](百分比)	
漫反射成分绘制		0.6（2.0%）	0.7（2.2%）
基于时间相关性的反射投影	最佳像素匹配	3.4（11.6%）	2.2（7.1%）
	反射偏移	1.3（4.4%）	1.7（5.5%）
	反射图像扭曲	2.7（9.2%）	2.1（6.9%）
生成图像合成	反射重计算	20.3（69.2%）	23.2（75.8%）
	混合绘制	0.9（3.1%）	0.7（2.2%）
总计		29.3	30.5

5.5.4 算法讨论和局限性分析

本章提出了一种基于时间相关性的反射投影算法，此算法利用反射绘制帧间的相关性来加速耗时的反射效果绘制。本章提出的反射深度图像作为常用深度缓存的扩展既能

作为相机采样光线相交的场景表达，也能作为反射光线相交的场景表达。本章引入基于时间相关性的反射投影算法。首先根据当前相机位置计算每个像素的最近反射匹配点偏移并存储在偏移缓存中；其次该偏移值作用于反射深度图像中的反射像素生成反射中间图像；最后使用传统的基于深度的投影算法将反射中间图像投影到当前屏幕空间。本章算法通过使用基于网格的像素映射方法能够轻松判断哪些地方能被反射深度图像中的反射图像投影，哪些地方需要进行反射绘制重计算。

本章的算法只依赖于相机的移动位置，而且能够支持凹体，凸体和平面反射表面。它理论简单，可以被轻松集成在大多数光线跟踪效率是主要瓶颈的反射应用之中。它可以在减少反射光线跟踪的工作量的同时，生成没有明显错误的反射效果。本章的方法可以整合到主机用光线跟踪框架生成着色数据，然后传输数据到轻量级客户端里用来进行图片生成的系统，也可以整合到通过关键帧生成图像流的系统中。

本章算法绘制效率的提升以及工作量的减少是以投影反射带来的绘制质量下降为代价的。这种质量上的下降主要来源于以下几点：1.本章算法假定最佳匹配距离在反射物表面单调变化最后在最佳匹配点时收敛到最小值，但这并不对所有反射物体表面都成立。2.反射深度图像的有限分辨率不能完全表现被反射场景细节。3.运动视差带来的反射光线遮挡误差。本章算法的局限在于它不能做动态场景的反射预测。反射投影是基于先前关键帧中的反射深度图像被反射物的场景表达，但是它仍可用于对动态场景中的静态部分的反射投影预测。另外一个限制是关键帧的有效期窗口。反射随着相机移动而快速变化，如果移动距离较大情况下关键帧中反射信息基本不能投影到当前帧中。本章算法可用通过把搜索不到映射点的反射区域的可靠程度置为不可靠，然后重新进行反射光线跟踪来处理这种情况，这样情况下本章算法退化为单纯的光线跟踪方法。

本章算法使用邻域搜索法寻找反射深度图像上的最佳反射匹配点。本章的邻域搜索的方法只能确定出下一个搜索像素对于当前搜索点的方向，然后将该方向的下一个邻域搜索点作为当前搜索点继续搜索，这造成了搜索的过程和深度图像的分辨率大小有关。帧缓存上的像素数目越多，需要搜索的距离越长，匹配收敛得越慢。本文的方法使用搜索点的位置和法线建立的拟合平面对采样几何体进行一阶建模，未来的研究方向包括使用屏幕空间的差分信息对搜索点局部进行二阶建模。用像素采样点的二阶拟合方程建立能确定出邻域搜索方向和搜索偏移距离的方程来提升搜索的效率，降低搜索的复杂程度。

本章算法同样可以通过再次扩展反射深度图像的定义让它存储更高阶反射光线与场景的相交信息和执行多次反射投影的方法支持高阶反射的投影。但是因为高阶反射基本上很难被人发觉，没必要对高阶反射花费过多的开销，所以本章方法专注于一阶反射光线的投影。

本章方法验证了基于深度缓存投影方式的时间相关性复用技术不仅可以加速一阶视无关光照的绘制特效，还能加速二阶镜面反射绘制特效。以后的可能工作方向之一是继续扩展本章二阶光线时间相关性复用技术到其他视相关高阶光照特效的绘制上，这些光照特效包括折射绘制和焦散效果绘制等。另外一个可能工作方向包括更精确地检测出反射投影区域的可靠性，这样可以通过光线重计算来消除正确反射和错误投影的间隔。同样为了能使反射重计算区域和反射投影模拟区域有平滑渐变，使用轻量级的减少视觉误差方法，例如视空间的过滤模糊降噪技术，也是我们的研究方向。

5.6 小结

常规的基于深度投影的关键帧像素复用方式不能直接应用在视相关的反射效果复用上。针对这一问题，本章提出一种基于时间连续性的反射绘制特效复用方式。本章在关键帧生成时额外记录当前视点下被反射场景信息的反射深度图像。本章引入基于时间相关性的反射投影算法。首先根据新相机位置计算像素点的最佳反射匹配点，然后记录到最佳匹配点的矢量到偏移缓存，接着通过索引偏移值移动反射深度图像中的像素生成反射中间图像，最后使用传统的基于深度的投影算法将反射中间图像投影到当前屏幕空间。通过使用基于网格的像素映射方法能够方便地判断哪些地方能被反射深度图像中的反射图像投影，哪些地方需要进行反射绘制重计算。实验结果表面：本章提出的基于时间相关性的反射投影算法能在生成无明显错误的反射绘制图像情况下减少了反射绘制的总开销，提高了绘制效率。

结论与展望

结论

随着 GPU 运算能力的日益增强, 以及移动计算平台计算能力的提升, 利用硬件友好的深度图像作为场景近似进行特效绘制已成为计算机实时绘制技术发展的新趋势, 在计算机图形学领域中具有重要的科学意义与应用价值。本文在改进深度图像的采样方式、提升离散化深度图像重建质量、设计深度图像能按需表示被反射场景采样策略和复用反射深度图像加速反射绘制等方面做了深入的研究:

1) 提出基于非冗余绘制的场景采样方法

随着三维扫描技术精度的提升和虚拟场景建模技术的不断发展, 三维场景模型的场景复杂度也随之增高, 在 GPU 绘制能力一定的情况下, 场景绘制速度随场景复杂程度增加而降低。通过深度图像组近似表达复杂场景可提升绘制效率, 但是传统深度图像绘制方式下, 不同深度图像的采样点集合之间存在冗余采样点, 这限制了深度图像组对完整场景的表达有效性。本文给出了非冗余绘制方式解决采样点冗余的问题。本文给出三种采样点冗余的定义方式和非冗余绘制算法的实现方法。非冗余绘制算法通过检测和避免场景中冗余采样点的绘制, 捕获到传统采样方法不能捕获到的非冗余像素采样点。本文给出三种视无关度量方式和一种视相关度量方式对场景采样方法进行对比评估, 验证了在这四种度量方式下非冗余绘制采样方法在采样有效性上的优势。这种优势表示在, 相比于传统深度图像能采样到传统深度图像采样不到的遮挡采样点, 而其中大部分采样点都是对中间输出图像有用的; 相比与深度剥离方法, 非冗余绘制算法能摆脱单视点的限制而捕获更多的遮挡采样点, 而且有用的采样点比例更大。

2) 提出适用于非冗余深度图像的场景重建和软阴影绘制方法

大部分场景采样方法在绘制应用中需要相应的重建算法进行场景重建。非冗余性绘制采样生成图像的高碎片化绘制特性给场景重建工作带来了困难。本文提出一种适用于非冗余图像的场景重复方法, 这种重建方法使用点重建和网格重建的混合思想, 通过额外绘制记录深度图像像素角点信息的偏移图像, 完成非冗余绘制图像的密闭几何体重建, 重建的几何体没有过度的绘制开销。本文应用非冗余绘制算法及其几何重建方法到软阴影的绘制算法中, 实验结果表明, 非冗余绘制采样方法的采样性能上的优势能传递到场景重建和近似几何体的软阴影绘制质量上的优势, 非冗余绘制图像生成的软阴影效

果图相对于物理真实几何体生成的软阴影绘制效果近似误差更小。

3) 提出基于聚类深度图像的反射绘制方法

深度图像常常作为相机发出的一阶观察采样光线所采样场景的近似表达,但是二阶光线譬如说由场景中反射材质生成的反射光线的采样场景时使用深度图像近似表达就比较困难。针对这一难点,本文提出了一种基于聚类深度图像的反射绘制算法:首先通过屏幕空间像素聚类的方法对反射像素进行视空间划分,然后由聚类相机生成该聚类反射场景的深度图像作为该聚类反射场景的近似表达,最后再由反射光线与聚类深度图像的求交计算反射光线的颜色值。该算法通过扩大反射角度、改变聚类相机远裁剪面和使用多根反射光线进行求交的方式支持光泽反射的绘制。实验结果表明,本文方法能在实时绘制情况下生成高质量的镜面和光照反射效果图像,生成图像质量与光线跟踪方法没有明显差异。本章算法不需要预处理,能支持动态场景的绘制,在动态场景下效率能达到光线跟踪反射绘制方法的 10 倍以上。

4) 提出基于反射深度图像相关性的反射绘制方法

应用时间相关性算法将已经计算好带有光照信息的深度图像映射到新的相机空间下是一种常见的复杂场景快速绘制方法,这是由于通常复用已经计算好的光照信息比重新进行光照计算开销要少。本文提出一种基于时间相关性的反射投影算法,此算法利用帧间反射的相关性来加速耗时的反射效果绘制。本文提出反射深度图像作为常用深度图像的扩展既能近似表达相机观察光线采样的场景,也能近似表达被反射场景。本章引入基于时间相关性的反射投影算法。首先根据新相机位置计算像素点的最佳反射匹配点,然后记录到最佳匹配点的矢量到偏移缓存,接着通过索引偏移值移动反射深度图像中的像素生成反射中间图像,最后使用传统的基于深度的投影算法将反射中间图像投影到当前屏幕空间。通过使用基于网格的像素映射方法能够方便地判断哪些地方能被反射深度图像中的反射图像投影,哪些地方需要进行反射绘制重计算。本文方法通过复用关键帧中的反射深度图像,减少了反射绘制的总开销,提高了绘制效率。

进一步工作

本论文以深度图像作为场景近似表达为背景,在扩大深度图像的采样范围、解决深度图像采样性冗余的问题、如何由非冗余深度图像重建出几何场景、按需生成被反射场景的聚类相机深度图像近似表达和计算深度图像在高阶光线绘制特效的复用问题进行了研究。但深度图像作为场景近似表达的算法内容广泛,若想继续扩大深度图像的表达

范围和开发其他基于深度图像的绘制特效算法，还需进一步开展理论与实践。今后的研究将主要集中在以下几个方面：

1) 研究深度图像采样相机的布置方式

本文通过检测和避免绘制冗余采样点，捕获非冗余采样点来提高给定视点生成深度图像对场景的表达能力。但是本文没有考虑视点的设定问题。如何根据用户兴趣区域和已捕获的场景信息动态设置后续相机的位置和参数能从另一方面提高生成深度图像对输入场景的近似表达能力。本文下一步工作将研究如何从已生成的深度图像中分析遮挡区域和需要重点采样的区域按需快速生成后续相机参数，更有效地按照应用需求完成场景采样工作。

2) 研究深度图像采样点集的紧凑场景重建方式

本文提出一种使用点绘制和网格绘制相结合的场景重建方法。本文方法通过偏移图像的绘制模拟生成像素的四个角点，从而能重建出密闭的重建曲面。本文方法为每个采样像素生成两个重建三角形，但是没有考虑重建三角形的压缩问题，造成了重建网格面片数目太大，场景近似表达绘制效率优势不明显。下一步工作将研究图像空间下场景几何体的解析式表达来生成紧致的三角形网格，进一步提高深度图像作为场景近似表达的效率优势。

3) 把深度图像绘制方法应用在更为复杂的光学逼真绘制

本文探讨了相机发出的一阶采样光线和反射材质生成的二阶反射光线的绘制情况，除开本文已经探索过的局部漫反射材质，镜面反射材质，光滑反射材质之外，现实世界中还存在着具有复杂微结构表面和基于物理真实复杂模型的混合材质、和时变多光源等影响着色绘制的复杂因素。因此，如何综合考虑这些复杂因素，以生成复杂光照条件下利于 GPU 进行计算的近似表达模型和基于物理真实的数学表达或近似计算方法来计算复杂光照信息作为突破口，进行实时的真实感特效绘制，值得研究人员的进一步研究。

4) 需要进一步提高绘制效率，以满足大规模复杂动态场景的实时绘制要求

复杂场景的绘制算法主要应用在游戏和实时仿真领域，除开绘制开销之外，大量的主机计算资源会分配给物理学模拟和任务逻辑等方面的计算，这限制了能获取到的绘制资源。另外，这类应用场景对计算的实时性有很高的要求。本文所述的场景替代表示方法虽然相比于物理学直接绘制方法有很大性能提升，但是还是不能满足的游戏或实时仿真领域的高实时性能要求。因此，进一步提高复杂动态场景的绘制效率无疑是一个重要

的研究方向。

参考文献

- [1] 赵沁平. 虚拟现实综述[J]. 中国科学F辑: 信息科学, 2009, 39(1): 2-46
- [2] 赵沁平, 沈旭昆, 夏春和, 王兆其. DVENET: 一个分布式虚拟环境[J]. 计算机研究与发展, 1998, 35(13): 1064-1068
- [3] 赵沁平, 沈旭昆, 吴威, 梁晓辉, 郝爱民. 分布式虚拟环境DVENET研究进展[J]. 系统仿真学报, 2000, 12(7): 291-295
- [4] 汪成为, 高文, 王行仁. 灵境(虚拟现实)技术的理论、实现及应用[M]. 清华大学出版社与广西科学技术出版社, 1996
- [5] 汪成为. 灵境技术与人机和谐仿真环境[J]. 计算机研究与发展, 1997, 34(1): 1-12
- [6] Burdea G. Virtual reality systems and applications[A]. Electro'93 International Conference[C], New York: ACM Press, 1993: 164-171
- [7] 赵沁平, 怀进鹏, 李波, 沈旭昆. 虚拟现实研究概况[J]. 计算机研究与发展, 1996, 33(7): 493-500
- [8] 赵沁平. DVENET分布式虚拟环境[M]. 北京: 科学出版社, 2002
- [9] 赵沁平. DVENET分布式虚拟现实应用系统运行平台与开发工具[M]. 北京: 科学出版社, 2005
- [10] Schmeing M, Jiang X. Depth Image Based Rendering[M]// Pattern Recognition, Machine Intelligence and Biometrics. 2011:279-310.
- [11] Yoon S E, Manocha D. R-LODs: fast LOD-based ray tracing of massive models[C]//ACM SIGGRAPH 2006 Sketches. ACM, 2006: 67.
- [12] Horn D R, Sugerman J, Houston M, et al. Interactive kd tree GPU raytracing[C]//Proceedings of the 2007 Symposium on Interactive 3D Graphics and Games. ACM, 2007: 167-174.
- [13] Carr N A, Hoberock J, Crane K, et al. Fast GPU ray tracing of dynamic meshes using geometry images[C]//Proceedings of Graphics Interface 2006. Canadian Information Processing Society, 2006: 203-209
- [14] Purcell T J, Buck I, Mark W R, et al. Ray tracing on programmable graphics

- hardware[J]. ACM Transactions on Graphics, 2002, 21(3):703-712.
- [15] Purcell J, Hanrahan P M. Ray Tracing on a Stream Processor[J]. 2004.
- [16] Liu B, Wei L Y, Yang X, et al. Non - Linear Beam Tracing on a GPU[C]//Computer Graphics Forum. Blackwell Publishing Ltd, 2011, 30(8): 2156-2169.
- [17] Vardis K, Vasilakis A A, Papaioannou G. A multiview and multilayer approach for interactive ray tracing[C]//Proceedings of the 20th ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games. ACM, 2016: 171-178.
- [18] Amanatides J. Ray tracing with cones [A]. SIGGRAPH '84: Proceedings of the 11th Annual Conference on Computer Graphics and Interactive Techniques [C]. New York: ACM Press, 1984: 129-135
- [19] Kirk D. B. The simulation of natural features using cone tracing [A]. Proceedings of Computer Graphics Tokyo '86 on Advanced Computer Graphics[C], New York: Springer-Verlag New York, Inc., 1986: 129-144
- [20] Cook R. L. Stochastic sampling in computer graphics [J]. ACM Trans. Graph., 1986, V5(1): 51-72
- [21] Mitchell D. P. Generating antialiased images at low sampling densities [A]. SIGGRAPH '87: Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques[C]. New York: ACM Press, 1987: 65-72
- [22] Mitchell D. P. Consequences of stratified sampling in graphics [A]. In SIGGRAPH '96: Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques[C]. New York: ACM Press, 1996: 277-280
- [23] Billen N, Dutré P. Line Sampling for Direct Illumination[J]. Computer Graphics Forum, 2016, 35(4):45-55.
- [24] Gouraud, Henri. Continuous shading of curved surfaces[J]. IEEE Transactions on Computers, 1971, C-20(6):623-629.
- [25] Phong B T. Illumination for computer generated pictures[J]. Communications of the ACM, 1998, 18(6):311-317.
- [26] Torrance K E, Sparrow E M, Birkebak R C. Polarization, directional distribution, and off-specular peak phenomena in light reflected from

- roughened surfaces[J]. Journal of the Optical Society of America, 1966, 56(7):916-924.
- [27] Blinn J F. Models of light reflection for computer synthesized pictures[J]. ACM SIGGRAPH Computer Graphics, 1977, 11(2):192-198.
- [28] Gouraud H. Computer display of curved surfaces[R]. UTAH UNIV SALT LAKE CITY COMPUTER SCIENCE DIV, 1971.
- [29] Blinn J F. Simulation of wrinkled surfaces[J]. ACM Siggraph Computer Graphics, 1978, 12(3):286-292.
- [30] Gardner G Y. Visual simulation of clouds[C]// Conference on Computer Graphics and Interactive Techniques, SIGGRAPH. DBLP, 1985:297-304.
- [31] Blinn J F, Newell M E. Texture and reflection in computer generated images[J]. Communications of the ACM, 1976, 19(10):542-547.
- [32] Miller G S, Hoffman C R. Illumination and Reflection Maps: Simulated Objects in Simulated and Real Environments[J]. 1984.
- [33] Kay D S, Greenberg D. Transparency for computer synthesized images[J]. ACM Siggraph Computer Graphics, 1979, 13(2):158-164.
- [34] Cook R L. Shade trees[C]// Conference on Computer Graphics and Interactive Techniques. ACM, 1984:223-231.
- [35] Schaufler G. Dynamically generated impostors[C]//GI Workshop Modeling-Virtual Worlds-Distributed Graphics. 1995: 129-136.
- [36] Maciel P W C, Shirley P. Visual navigation of large environments using textured clusters[C]//Proceedings of the 1995 Symposium on Interactive 3D Graphics. ACM, 1995: 95-ff.
- [37] Williams L. Casting curved shadows on curved surfaces[J]. ACM Siggraph Computer Graphics, 1978, 12(3):270-274.
- [38] Reeves W T, Salesin D H, Cook R L. Rendering antialiased shadows with depth maps.[J]. ACM Siggraph Computer Graphics, 1987, 21(4):283-291.
- [39] Dimitrov R. Cascaded shadow maps[J]. Developer Documentation, NVIDIA Corp, 2007.

-
- [40] Yang B, Dong Z, Feng J, et al. Variance soft shadow mapping[C]//Computer Graphics Forum. Blackwell Publishing Ltd, 2010, 29(7): 2127-2134.
- [41] Brabec S, Annen T, Seidel H P. Practical shadow mapping[J]. Journal of Graphics Tools, 2002, 7(4): 9-18.
- [42] Gascuel J D, Holzschuch N, Fournier G, et al. Fast non-linear projections using graphics hardware[C]//Proceedings of the 2008 Symposium on Interactive 3D Graphics and Games. ACM, 2008: 107-114.
- [43] Fillion D, McNaughton R. Effects & techniques[C]//ACM SIGGRAPH 2008 Games. ACM, 2008: 133-164.
- [44] Soler C, Hoel O, Rochet F. A deferred shading pipeline for real-time indirect illumination[C]//ACM SIGGRAPH 2010 Talks. ACM, 2010: 18.
- [45] Shishkovtsov O. Deferred shading in S.T.A.L.K.E[J]. GPU Gems, 2005, 2: 143-166.
- [46] Jimenez J, Gutierrez D. Screen-Space Subsurface Scattering[J]. GPU Pro: Advanced Rendering Techniques, 2010: 335-351.
- [47] Ritschel T, Grosch T, Seidel H P. Approximating dynamic global illumination in image space[C]// Symposium on Interactive 3d Graphics and Games. ACM, 2009:75-82.
- [48] KOONCE, R. 2008. Deferred Shading in Tabula Rasa[J]. In GPU Gems 3, 429–457.
- [49] Max N, Ohsaki K. Rendering Trees from Precomputed Z-Buffer Views[M]// Rendering Techniques '95. Springer Vienna, 1995:74-81.
- [50] Everitt C. Interactive order-independent transparency[J]. White paper, nVIDIA, 2001, 2(6): 7.
- [51] Bavoil L, Callahan S P, Lefohn A, et al. Multi-fragment effects on the GPU using the k-buffer[C]//Proceedings of the 2007 Symposium on Interactive 3D Graphics and Games. ACM, 2007: 97-104.
- [52] Bavoil L, Myers K. Order independent transparency with dual depth peeling[J]. NVIDIA OpenGL SDK, 2008: 1-12.
- [53] Shade J, Gortler S, He L, et al. Layered depth images[C]//Proceedings of the

- 25th Annual Conference on Computer Graphics and Interactive Techniques. ACM, 1998: 231-242.
- [54] Chang C F, Bishop G, Lastra A, et al. A Hierarchical Representation for Image-based Rendering[J]. University of North Carolina at Chapel Hill, Chapel Hill, NC, 1998
- [55] Levoy M. Light field rendering[C]// Conference on Computer Graphics and Interactive Techniques. ACM, 1996:31-42.
- [56] Landy M, Movshon J. The Plenoptic Function and the Elements of Early Vision[J]. Computational Models of Visual Processing, 1997, 1:3-20.
- [57] Gortler S J, Grzeszczuk R, Szeliski R, et al. The Lumigraph[J]. ACM Siggraph Computer Graphics, 1996, 96:43--54.
- [58] Leonard McMillan. Plenoptic modeling: an image-based rendering system[C]// Conference on Computer Graphics and Interactive Techniques. ACM, 1995:39-46.
- [59] Budge L. D., Strini R. A., Dehncke R. W., et al. Synthetic theater of war (stow) 97 overview[A]. Simulation Interoperability Workshop Proceeding of Spring 1998[C], New York: ACM Press, 1998:213-230
- [60] Bolas M, Kuruvilla A, Chintalapudi S, et al. Creating near-field VR using stop motion characters and a touch of light-field rendering[C]// ACM SIGGRAPH. ACM, 2015:1-1.
- [61] Heidrich W, Lensch H, Cohen M F, et al. Light field techniques for reflections and refractions[M]//Rendering Techniques' 99. Springer Vienna, 1999: 187-196.
- [62] Taguchi Y, Agrawal A, Ramalingam S, et al. Axial light field for curved mirrors: Reflect your perspective, widen your view[C]//Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on. IEEE, 2010: 499-506.
- [63] Yu J, Yang J, McMillan L. Real-time reflection mapping with parallax[C]//Proceedings of the 2005 Symposium on Interactive 3D Graphics and Games. ACM, 2005: 133-138.

-
- [64] Paul G, Jan K, Frédo D. Efficient Reflectance and Visibility Approximations for Environment Map Rendering[J]. Computer Graphics Forum, 2007, 26(3):495-502.
- [65] McCool M D. Approximation of Glossy Reflection with Prefiltered Environment Maps[J]. Proceedings - Graphics Interface, 2000:119--126.
- [66] Szirmay-Kalos L, Aszódi B, Lazányi I, et al. Approximate Ray-Tracing on the GPU with Distance Impostors[J]. Computer Graphics Forum, 2005, 24(3):695-704.
- [67] Voicu P, Mei C, Jordan D, et al. Reflected - Scene Impostors for Realistic Reflections at Interactive Rates[J]. Computer Graphics Forum, 2010, 25(3):313-322.
- [68] McGuire M, Mara M. Efficient GPU screen-space ray tracing[J]. Journal of Computer Graphics Techniques (JCGT), 2014, 3(4): 73-85.
- [69] Vardis K, Vasilakis A A, Papaioannou G. A multiview and multilayer approach for interactive ray tracing[C]// ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games. ACM, 2016:171-178.
- [70] Ofek E, Rappoport A. Interactive reflections on curved objects[C]//Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques. ACM, 1998: 333-342.
- [71] Estalella P, Martin I, Drettakis G, et al. A GPU-driven algorithm for accurate interactive reflections on curved objects[C]// Eurographics Symposium on Rendering Techniques, Nicosia, Cyprus. DBLP, 2006:313-318.
- [72] Popescu V, Sacks E, Mei C. Sample-based cameras for feed forward reflection rendering[J]. IEEE Transactions on Visualization & Computer Graphics, 2006, 12(6):1590-1600.
- [73] Nehab D, Sander P V, Lawrence J, et al. Accelerating real-time shading with reverse reprojection caching[C]// ACM Siggraph/eurographics Conference on Graphics Hardware 2007, San Diego, California, Usa, August. DBLP, 2007:25-35.
- [74] Scherzer D, Jeschke S, Wimmer M. Pixel-correct shadow maps with temporal

- reprojection and shadow test confidence[C]// Eurographics Conference on Rendering Techniques. Eurographics Association, 2007:45-50.
- [75] McMillan Jr L. An image-based approach to three-dimensional computer graphics[D]. University of North Carolina at Chapel Hill, 1997.
- [76] Didyk P, Eisemann E, Ritschel T, et al. Perceptually - motivated Real - time Temporal Upsampling of 3D Content for High - refresh - rate Displays[C]//Computer Graphics Forum. Blackwell Publishing Ltd, 2010, 29(2): 713-722.
- [77] Didyk P, Ritschel T, Eisemann E, et al. Adaptive Image-space Stereo View Synthesis[C]//VMV. 2010: 299-306.
- [78] Yu X, Wang R, Yu J. Real - time Depth of Field Rendering via Dynamic Light Field Generation and Filtering[C]//Computer Graphics Forum. Blackwell Publishing Ltd, 2010, 29(7): 2099-2107.
- [79] Yang L, Tse Y C, Sander P V, et al. Image-based bidirectional scene reprojection[C]//ACM Transactions on Graphics (TOG). ACM, 2011, 30(6): 150.
- [80] Policarpo F, Oliveira M M. Relief mapping of non-height-field surface details[C]//Proceedings of the 2006 Symposium on Interactive 3D Graphics and Games. ACM, 2006: 55-62.
- [81] Heckbert P S, Herf M. Simulating soft shadows with graphics hardware[R]. CARNEGIE-MELLON UNIV PITTSBURGH PA DEPT OF COMPUTER SCIENCE, 1997.
- [82] Eisemann E, Décoret X. On Exact Error Bounds for View - Dependent Simplification[C]//Computer Graphics Forum. Blackwell Publishing Ltd, 2007, 26(2): 202-213.
- [83] Sintorn E, Eisemann E, Assarsson U. Sample based visibility for soft shadows using Alias - free shadow maps[C]//Computer Graphics Forum. Blackwell Publishing Ltd, 2008, 27(4): 1285-1292.
- [84] Mittring M. Finding next gen: Cryengine 2[C]//ACM SIGGRAPH 2007

- courses. ACM, 2007: 97-121.
- [85] Mattausch O, Scherzer D, Wimmer M. High - Quality Screen - Space Ambient Occlusion using Temporal Coherence[C]//Computer Graphics Forum. Blackwell Publishing Ltd, 2010, 29(8): 2492-2503.
- [86] Parker S G, Bigler J, Dietrich A, et al. Optix: a general purpose ray tracing engine[C]//ACM Transactions on Graphics (TOG). ACM, 2010, 29(4): 66.
- [87] Laine S, Saransaari H, Kontkanen J, et al. Incremental instant radiosity for real-time indirect illumination[C]//Proceedings of the 18th Eurographics conference on Rendering Techniques. Eurographics Association, 2007: 277-286.
- [88] Schwärzler M, Luksch C, Scherzer D, et al. Fast percentage closer soft shadows using temporal coherence[C]//Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games. ACM, 2013: 79-86
- [89] Wang R, Wang R, Zhou K, et al. An efficient GPU-based approach for interactive global illumination[J]. ACM Transactions on Graphics (TOG), 2009, 28(3): 91
- [90] Grimsdale C. Supervision-a parallel architecture for virtual reality [A]. Virtual Reality Systems[C], London: Academic Press, 1993: 29-38
- [91] Lewis J. B., Koved L., Ling D. T. Dialogue structures for virtual worlds [A]. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems[C], New York: ACM Press, 1991: 131-136
- [92] Mei C, Popescu V, Sacks E. The occlusion camera[C]//Computer Graphics Forum. Blackwell Publishing, Inc, 2005, 24(3): 335-342.
- [93] Popescu V, Aliaga D. The depth discontinuity occlusion camera[C]//Proceedings of the 2006 Symposium on Interactive 3D Graphics and Games. ACM, 2006: 139-143.
- [94] Rosen P, Popescu V. The epipolar occlusion camera[C]//Proceedings of the 2008 Symposium on Interactive 3D Graphics and Games. ACM, 2008: 115-122.
- [95] Cui J, Rosen P, Popescu V, et al. A curved ray camera for handling occlusions

- through continuous multiperspective visualization[J]. IEEE Transactions on Visualization and Computer Graphics, 2010, 16(6): 1235-1242.
- [96] Popescu V, Rosen P, Arns L, et al. The general pinhole camera: Effective and efficient nonuniform sampling for visualization[J]. IEEE Transactions on Visualization and Computer Graphics, 2010, 16(5): 777-790.
- [97] Oliveira M M, Brauwers M. Real-time refraction through deformable objects[C]//Proceedings of the 2007 Symposium on Interactive 3D Graphics and Games. ACM, 2007: 89-96.
- [98] Chen S E. Quicktime VR: An image-based approach to virtual environment navigation[C]//Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques. ACM, 1995: 29-38.
- [99] Szeliski R, Shum H Y. Creating full view panoramic image mosaics and environment maps[C]//Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques. ACM Press/Addison-Wesley Publishing Co., 1997: 251-258.
- [100] Shum H Y, He L W. Rendering with concentric mosaics[C]//Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques. ACM Press/Addison-Wesley Publishing Co., 1999: 299-306.
- [101] Ihm I, Park S, Lee R K. Rendering of spherical light fields[C]//Computer Graphics and Applications, 1997. Proceedings., The Fifth Pacific Conference on. IEEE, 1997: 59-68.
- [102] Camahort E, Leros A, Fussell D. Uniformly sampled light fields[M]//Rendering Techniques' 98. Springer Vienna, 1998: 117-130.
- [103] Gurrea E C. 4D Light-Field Modeling and Rendering[D]. Phd thesis, The University of Texas at Austin, 2001.
- [104] Sloan P P, Cohen M F, Gortler S J. Time critical lumigraph rendering[C]//Proceedings of the 1997 Symposium on Interactive 3D Graphics. ACM, 1997: 17-ff.
- [105] Katayama A, Tanaka K, Oshino T, et al. Viewpoint-dependent stereoscopic

- display using interpolation of multiviewpoint images[C]//IS&T/SPIE's Symposium on Electronic Imaging: Science & Technology. International Society for Optics and Photonics, 1995: 11-20.
- [106] Vardis K, Vasilakis A A, Papaioannou G. DIRT: deferred image-based ray tracing[C]//Proceedings of High Performance Graphics. Eurographics Association, 2016: 63-73.

攻读博士学位期间取得的研究成果

1. 学术论文

- [1] Lili Wang, **Naiwen Xie**, Wei Ke, Voicu Popescu. Second-Order Feed-Forward Rendering for Specular and Glossy Reflections[J]. **IEEE Transactions on Visualization and Computer Graphics (TVCG)**, 2014, 20(9): 1316-1329, (SCI, IF: 2.168, JCR Q1 区, 2014 年, CCF A 类期刊)
- [2] **Naiwen Xie**, Lili Wang, Voicu Popescu. Non-redundant rendering for efficient multi-view scene discretization[J]. **The Visual Computer**, 2016: 1-15, (SCI, IF:1.06, JCR Q2 区, 2015 年, CCF C 类期刊)
- [3] **Naiwen Xie**, Lili Wang, Philip Dutré. Reflection reprojection using temporal coherence [J]. **The Visual Computer**, 2017: 1-13, (SCI, IF:1.06, JCR Q2 区, 2015 年, CCF C 类期刊)
- [4] 彭通, 王莉莉, **谢乃闻**, 赵沁平. 基于方向光辐射度的微结构表面全局光照绘制[C]. 第十八届全国计算机辅助设计与图形学学术会议(CAD/CG2014), 2014

2. 发明专利

- [1] 王莉莉, **谢乃闻**, 杨晨, 赵沁平. 一种基于多相机的快速反射绘制方法[P], 授权, 专利号: ZL201310462932.3
- [2] 王莉莉, 彭通, **谢乃闻**, 周士恒, 赵沁平. 一种基于方向光辐射度的微结构表面全局光照绘制方法[P]. 申请专利号: 2014.09.05, 201410370963.0

3. 已获荣誉和奖励

- [1] 博士研究生国家奖学金, 2014
- [2] 国家建设高水平大学公派研究生基金, 2015
- [3] 北航博士生创新基金, 2015
- [4] 北航校长奖学金, 2015
- [5] 北航光华奖学金, 2016
- [6] 北航计算机学院创新奖, 2014
- [7] 优秀学生干部 x 2, 2012, 2013
- [8] 校级三好学生, 2012

4. 参与主要科研项目

- [1] 国家 863 计划项目“智慧城市课题四—多源密集型动态运行数据呈现技术与服务系统”，2013.1-2015.12
- [2] 国家自然科学基金“时变复杂场景的图像空间光照算法研究”，2012.1–2013.12
- [3] 国家自然科学基金项目“面向实时逼真绘制的非传统相机模型研究”，2012.3-2016。12
- [4] 国家自然科学基金重大项目“可交互人体器官数字模型及虚拟手术研究”，2012-至今

致 谢

当我在电脑上敲下“致谢”之时，硕博连读生涯的六年时光，也渐渐浮现在我的脑海。上上个月，我度过了自己 29 岁的生日，求学的生涯，占据了我生命三成以上的时光。从一个不识一丁的孩童到一位受过高等教育的青年。白驹过隙，其实想来，五味杂陈。

以一个还未走入社会的身份来谈论成长未免太过书面气。再年轻几岁的时候，我也希望自己能对这个世界产生影响，让世界知道我的存在，到头来，越发想与自己讲和。一路走来，我不断地给我自己的为了进行定位，进过企业，做过码农。到最后，阴差阳错却命运使然地还是选择读博这条道路。并不是因为博士比较高尚，而是我所做的，不过是在选择现在这个的年纪，选择最容易在世界上留下存在痕迹的生活方式，也许若干年后被我的后代还是通过 cnki 检索到我的科研，也是一件欣慰的事情。博士攻读的过程无疑是一种痛苦的磨砺。当然，我并不能声称自己所遇到的困难有多么了不起。在这个过程中，我获得最多的东西，反而是一种生活的体验。

硕博连读六年，我首先要感谢自己的博士生导师赵沁平教授。能够成为他的学生是我的荣幸。赵老师曾在北航计算机学院创新奖学金颁奖典礼上讲述博士的意义。“假设人类的知识是一个圈，博士之前的教育让你领会到人类知识的最边缘，在北航钻研个几年，直到有一天，你突破了这一点，你突破的这个点，那就是你的博士学位，但是，这点贡献在知识圈只算沧海一粟。”这句话一直伴我左右。于我而言，能做的便是将学术中的“创新”视为自己坚守之物，虽能力有限，但绝不懈怠；同时这句话在我困顿的时候激励我前面仍有无垠的地带等待我去发掘，继续钻研！

特别感激王莉莉教授。作为我博士阶段的副导师，给予了我生活上、学习上无微不至的关心，这篇博士论文的源头，便来自于我们的交谈。难忘岁月，永记于心。在博士论文的写作过程中，来自美国普渡大学 Voicu Popescu 教授和来自比利时鲁汶大学 Philip Dutre 教授抽出时间给我的论文提出诸多建议。再次感谢比利时鲁汶大学 Philip Dutre 教授，在比利时的求学过程中对我的学习和生活给予了莫大帮助。

感谢一起工作过的博士师兄弟们：沈哲、杨晨、马志强、刘鑫达，你们给予了我很大的支持。感谢已毕业的几位同学：薛冰、金其江、杨峥、章二林、李鸿飞、王泽、石裕隆、陈益、张鑫维、王立平、周世恒、戚庆林、彭通、张博宁、吴军、李念，在此祝

愿你们在未来的工作生活中一帆风顺。谢谢 G701 室的白隼瑄、翟泉、夏清、高阳、李晓、李韵潇、张东波、马际洲几位师弟和宋文凤师妹，有了你们 G701 不再枯燥，不再单调。尽管 G701 环境单调，但是有你们，也让我觉得整个屋子充满活力。

感谢中国国家留学基金委基金编号 No. 201506020037 和北航博士创新基金给予的研究资金支持。

我最应感谢自己的家人，他们替我承受了生活的艰辛，使得我可以专注于自己的学业。在博士的攻读过程中，我甚少可以抽出时间陪他们。父母总担心我的压力过大，常常劝慰。如今论文尘埃落定，我希望可以抽出更多的时间，假期多陪他们走走。

感谢我的女朋友，我想将我的博士文献给你。几个月后，我要带着你去参加我的毕业典礼，亲手把我的博士帽，戴在你的头上。

作者简介

谢乃闻，男，1988 年生于四川省德阳市，汉族。

2007 年考入华中科技大学学院计算机科学与技术系，2011 年获工学学士学位。2011 年免试入学北京航空航天大学计算机学院攻读硕士学位，2012 年转为攻读博士学位，在虚拟现实技术与系统国家重点实验室攻读博士学位至今（期间从 2015.9 至 2016.9 月，在比利时鲁汶大学作为访问学者进行公派访学研究）。主要研究方向在于实时绘制，基于图像绘制，图像处理和光线跟踪。