



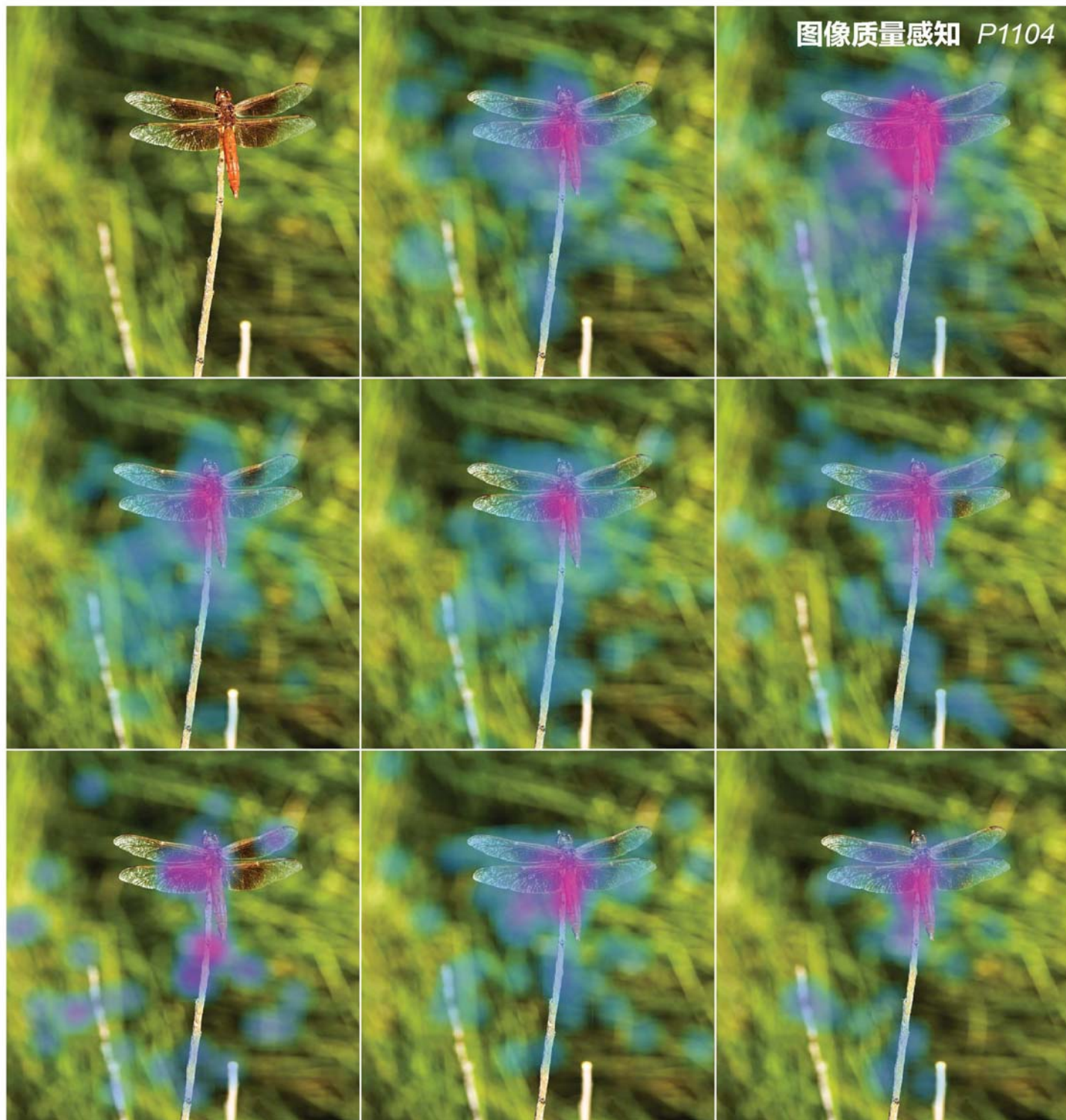
JOURNAL OF IMAGE AND GRAPHICS

主办: 中国科学院遥感与数字地球研究所
中国图象图形学学会
北京应用物理与计算数学研究所

中国图象图形学报

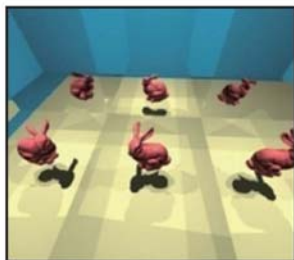
2014
07
VOL.19

ISSN1006-8961
CN11-3758/TB



多特征融合的图像自动变形
(第1012页)

多通道Haar-like特征多示例学习目标跟踪(第1038页)

快速虚顶点光子映射的焦散
绘制(第1062页)**综述**

打印机光谱特性化方法研究综述

万晓霞, 刘强 0985

图像处理和编码

图像场方向导数的局部区域重建

叶学义, 齐珍珍, 何志伟, 李汪兵 0998

Lytro相机的光场图像校正与重对焦方法

周文晖, 林丽莉 1006

多特征融合的图像自动变形

肖红光, 肖树根, 刘京, 李淳芃, 陈立福 1012

结合频谱扩散的指纹图像滤波增强

卞维新, 封举富, 罗永龙, 徐德琴 1021

图像分析和识别

中心投影变换的抗噪性分析

黄永东, 李三三, 杨建伟 1031

多通道Haar-like特征多示例学习目标跟踪

宁纪锋, 赵耀博, 石武祯 1038

自主移动机器人自适应室外道路检测

杜明芳, 王军政, 李静, 崔广涛, 方建军, 曹海青 1046

分层树结构字典编码的行为识别

周同驰, 程旭, 吴镇扬 1054

计算机图形学

快速虚顶点光子映射的焦散绘制

李瑞瑞, 秦开怀 1062

双重视图下的3维网格模型可见水印

陆尘, 朱长青, 王玉海 1068

虚拟现实与增强现实

个性化编辑的轻量化3维树木模型构建

董天阳, 纪磊, 刘思远, 范菁, 熊丽荣 1074

遥感图像处理

带距离徙动校正的压缩感知PFA成像方法

段化军, 朱岱寅 1085

结合边缘和区域的活动轮廓模型SAR图像目标轮廓提取

王沛, 周鑫, 彭荣赜, 符鹏 1095

第2届全国图像图形联合学术会议专栏

视线跟踪技术的图像质量主观数据库构建及其可视化

张昀, 牟轩沁 1104

丰富图像标签的正则化非负矩阵分解方法

夏召强, 冯晓毅, 彭进业 1112

CONTENTS

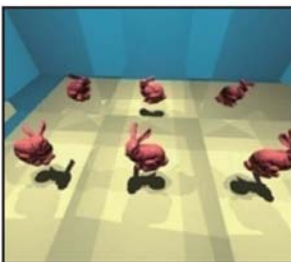
JOURNAL OF IMAGE AND GRAPHICS



Multi-feature fusion-based image morphing(P1012)



Multiple instance learning based object tracking with multi-channel Haar-like feature(P1038)



Fast caustics rendering by photon beam tracing based on virtual vertex transformations(P1062)

Review

- Review of spectral printer characterization
Wan Xiaoxia, Liu Qiang 0985

Image Processing and Coding

- Reconstructing image local regions based on directional derivative of a field
Ye Xueyi, Qi Zhenzhen, He Zhiwei, Li Wangbing 0998
- Light field image rectification and refocus method for Lytro camera
Zhou Wenhui, Lin Lili 1006
- Multi-feature fusion-based image morphing
Xiao Hongguang, Xiao Shugen, Liu Jing, Li Chunpeng, Chen Lifu 1012
- The fingerprint enhancement using filtering combined with spectrum diffusion
Bian Weixin, Feng Jufu, Luo Yonglong, Xu Deqin 1021

Image Analysis and Recognition

- Robustness to noise of central projection transformation
Huang Yongdong, Li Sansan, Yang Jianwei 1031
- Multiple instance learning based object tracking with multi-channel Haar-like feature
Ning Jifeng, Zhao Yaobo, Shi Wuzhen 1038
- Adaptive outdoor road detection for autonomous mobile robot
Du Mingfang, Wang Junzheng, Li Jing, Cui Guangtao, Fang Jianjun, Cao Haiqing 1046
- Action recognition using hierarchically tree-structured dictionary encoding
Zhou Tongchi, Cheng Xu, Wu Zhenyang 1054

Computer Graphics

- Fast caustics rendering by photon beam tracing based on virtual vertex transformations
Li Ruirui, Qin Kaihuai 1062
- Visible watermarking for three-dimensional mesh model in two views
Lu Chen, Zhu Changqing, Wang Yuhai 1068

Virtual Reality and Augmented Reality

- Construction of lightweight three-dimensional tree models supporting personalization editing
Dong Tianyang, Ji Lei, Liu Siyuan, Fan Jing, Xiong Lirong 1074

Remote Sensing Image Processing

- PFA imaging with range migration correction by compressive sensing
Duan Huajun, Zhu Daiyin 1085
- Active contour model based on edge and region attributes for target contour extraction in SAR image
Wang Pei, Zhou Xin, Peng Rongkun, Fu Peng 1095

Column of UCIG'2013

- Eye tracking based image quality assessment database and visualization method
Zhang Yun, Mou Xuanqin 1104
- Regularized non-negative matrix factorization method for image tag enrichment
Xia Zhaoqiang, Feng Xiaoyi, Peng Jinye 1112

中图法分类号: TN301 文献标识码: A 文章编号: 1006-8961(2014)07-1062-06

论文引用格式: Li R R, Qin K H. Fast caustics rendering by photon beam tracing based on virtual vertex transformations[J]. Journal of Image and Graphics, 2014, 19(7): 1062-1067. [李瑞瑞, 秦开怀. 快速虚顶点光子映射的焦散绘制[J]. 中国图象图形学报, 2014, 19(7): 1062-1067.] [DOI: 10.11834/jig. 20140710]

快速虚顶点光子映射的焦散绘制

李瑞瑞, 秦开怀

清华大学计算机科学与技术系, 北京 100084

摘要: 目的 模拟绘制焦散效果是真实感图形绘制的重要组成部分。利用可编程 GPU 硬件, 基于图像的光子映射方法绘制速度快, 但由于使用了近似采样和计算, 会产生失真。为了克服这一现象, 针对理想的镜面反射、折射体, 提出一种快速绘制焦散效果的方法 (VOBPBT)。方法 该方法首次定义了光子路径映射图的概念, 并提出通过基于虚顶点光子映射的光子束跟踪来准确构建光子路径映射图的方法; 此外, 方法也创新性地提出利用光子路径映射图来查找焦散三角形, 创建焦散映射图的方法。结果 实验结果表明, 本文绘制结果真实, 可以处理多次递归反射、折射, 能够绘制连贯的高频焦散效果, 同时可以达到交互的绘制性能。结论 本文 VOBPBT 方法在可交互计算机仿真、计算机游戏、虚拟漫游等应用领域具有一定的实用价值。

关键词: 焦散; 光束跟踪; 光子映射; 虚物体; GPU

Fast caustics rendering by photon beam tracing based on virtual vertex transformations

Li Ruirui, Qin Kaihuai

Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China

Abstract; Objective Caustic rendering plays an important part in the research of photorealistic image rendering. Based on programmable GPUs, image-based photon mapping renders fast but suffers from artifacts caused by sampling and computing approximately. **Method** To overcome this drawback, this paper proposes a fast method called virtual object-based photon beam tracing (VOBPBT) to render caustic effects. VOBPBT originally defines the photon-path map and then constructs the map. It also presents a way to find caustic triangles in the photon-path map and splat them onto the caustic map. **Result** This method can recursively render reflective and refractive effects as well as high-frequency caustic effects realistically in an interactive speed. **Conclusion** VOBPBT is useful in interactive computer-aided simulation, computer games, as well as virtual tours.

Key words: caustic; beam tracing; photon mapping; virtual objects; GPU

0 引言

反射和折射可能会引起光线汇聚的现象, 这种

光线汇聚产生的视觉效果被称为焦散, 它是真实感图形学不可缺少的重要组成部分。由于计算复杂度高, 针对理想的镜面反射、折射, 能够快速、真实地模拟其产生的焦散效果是一个难点, 也是国内外学者

收稿日期: 2013-12-02; 修回日期: 2014-01-14

基金项目: 北京市自然科学基金项目 (4102027); 博士点基金项目 (20090002110006)

第一作者简介: 李瑞瑞 (1981—), 女, 清华大学计算机系博士研究生, 主要研究领域为计算机图形学、虚拟现实、图像处理与可视化。

E-mail: ilydouble@gmail.com

通信作者: 秦开怀, E-mail: qkh-dcs@tsinghua.edu.cn

的研究热点。

光子映射是计算焦散效果的主要方法之一,它通过发射成千上万的光子来计算漫反射表面的辐射强度,其核心是光线跟踪。光子映射包含两个主要的绘制步骤:首先,从光源处出发,发射和跟踪光子;接下来,在可见像素位置的周围收集光子的光照贡献。光线跟踪需要计算每条光线与场景中的每个几何面片的交点,由于计算复杂度高,即使采用 GPU (graphic processing unit) 硬件加速,例如 OptiX^[1];或者利用一些空间加速的数据结构,例如 KD-Tree^[2]或者 BVH(bounding volume hierarchy)^[3],针对复杂动态场景,目前也难以满足实时交互绘制的需求,并且由于光子映射使用数目巨大的光子,光子的收集也需要庞大的计算开销,通常绘制一帧高质量的图像可能需要几秒钟^[4]。

可编程 GPU 出现以后,涌现出了基于图像的近似光子映射方法(Image-based Photon Mapping)^[5]。其主要思想是使用纹理图像对几何场景进行采样,然后在采样的纹理图像中通过线性或者切线搜索寻找交点,构建基于图像空间的光子映射图,从而近似计算出焦散效果。针对一些场景,其光子映射图可能存在着较大的误差,生成的焦散光斑的形状和实际情况出入较大;不仅如此,基于图像的近似光子映射方法存在光子的欠采样和过采样的问题,即欠采样会丢失高频光照信息,过采样会带来绘制性能的下降。尽管有一些研究工作通过层次化计算^[6]、自适应投射尺寸^[7]等方法来改善这一问题,但很多情况下,仍然受其所带来的失真问题的困扰。

基于光束跟踪的思想,一些研究工作不通过优化或简化光线跟踪来提高光子映射的性能,而是利用光线的连贯性,每次跟踪一束光线而不是一根光线。光束跟踪方法^[8]在 20 世纪 90 年代被提出,一开始主要针对平面反射;对于折射等非线性问题变换则比较复杂^[9-10]。秦开怀等人^[11-13]提出“光学变换虚顶点”的概念和方法,利用参数化方程和迭代求解一束光线的交点,解决了非线性折射的问题,并将这一方法扩展到隐式曲面表示的反射、折射体。遗憾的是,这一工作绘制速度较慢,没有使用 GPU 硬件进行加速,也未扩展应用到焦散效果的绘制。国外的研究者发现光束跟踪可以用来加速绘制焦散效果^[14],特别是针对水面产生的焦散效果^[15-16],但是由于多次迭代光束跟踪非常复杂,他们的方法通常假设光束只递归一次,且光束与几何

场景的交点之间不存在遮挡关系^[9]。

受到胡伟、秦开怀^[11]工作的启发,针对多边形面片表示的理想镜面反射、折射体,针对完全动态的场景,本文创新地提出一种基于虚顶点光子映射的快速焦散绘制方法,可以模拟多次递归反射、折射产生的焦散效果,可以正确地计算遮挡关系,并利用 GPU 硬件加速达到可交互的绘制性能。本文方法具有两个主要的创新点:

1) 基于光束跟踪,提出针对任意多边形表示的镜面反射、折射体的光子束跟踪方法,在每一束光内,优化迭代求解光子的位置,并提出基于 GPU 加速的多次递归光子束跟踪方法。

2) 提出光子路径映射图的构建方法以及基于光子路径映射图的光子束重建方法,避免由于欠采样所带来的失真。

1 虚顶点光子映射

从人眼方向进行观察,对于反射、折射体,可以看到真实物体在反射、折射表面上所成的虚拟像,这些像通常是空间和几何连贯的。利用这一特性,提出一个基于虚顶点光子映射的焦散绘制方法 VOB-PBT(virtual object based photon beam tracing)。它包含两个主要的绘制流程:首先是将视点放在光源处,沿着光的方向发射出多个光束,递归地跟踪每个光束,生成 1 幅光子路径映射图,如图 1(a)所示;然后,在光子路径映射图上,构建焦散三角形,并将这些三角形通过坐标变换、投影变换绘制到 1 幅纹理映射图上,构成焦散映射图^[7],在最终的着色阶段,叠加焦散映射图中保存的光能信息,从而绘制出焦散效果。

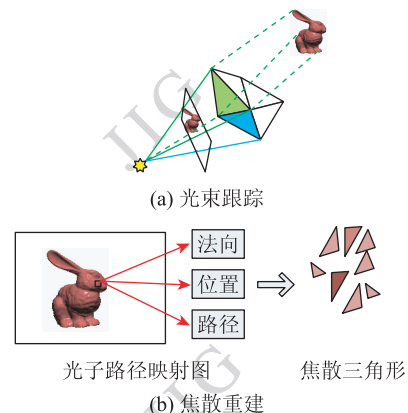


图 1 VOBPBT 方法绘制流程

Fig. 1 VOBPBT rendering process

1.1 虚顶点计算

在以多边形面片表示的几何场景中,从某一视点 V 观察过去,透过某一反射、折射体平面,场景中的几何顶点 P 会产生与其对应的虚拟顶点 P' (图 2),称为虚顶点^[12]。对于反射平面,其顶点 P 到虚顶点 P' 的变换是与视点 V 无关的,在计算时,将顶点投影在反射平面上,其虚顶点就位于顶点与投影点的连线的延长线上,并且虚顶点到平面的距离与顶点到平面的距离相同。对于折射平面,虚顶点的位置可以通过视点、顶点的位置以及折射率通过解一元四次方程计算出来^[12]。

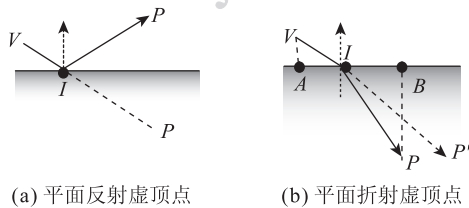


图 2 虚顶点计算

Fig. 2 Computation of virtual vertex

通常,场景中包含数以万级的顶点,计算它们对应的虚顶点的过程可以利用可编程图形流水线中的顶点着色器程序(Vertex Shader)进行加速,通过一趟绘制,计算出来所有对应的虚顶点。在 GPU 中,求解方程比较复杂,针对图 2(b)的示意图,采用如下二分查找方法:

算法 1 二分查找方法

假设 η 为折射率, ε 为误差;

1) 计算视点 V 在平面上的投影点 A ;

2) 计算顶点 P 在平面上的投影点 B ;

3) 取 A 、 B 的中点为测试点 O , 计算 $d = \sin \angle OVA - \eta \sin \angle OPB$, 如果误差小于给定的阈值 ε , 则 O 点即为交点 I , 得到虚顶点 P' , 结束;

4) 判断 d 的大小, 若大于 0, 则将 B 点移到 O 点的位置, 若小于 0, 则将 A 点移到 O 点的位置, 跳到步骤 3)。

1.2 多次递归光子束跟踪

光子束跟踪算法是将视点置于光源处, 沿着光的方向向每个反射、折射体都发射一个光束, 对于每个光束, 可以计算出场景所有顶点的虚顶点。但在观察中发现, 对于位于反射平面正面一侧的虚顶点和折射平面背面一侧的虚顶点进行计算, 是没有意义的, 如果将它们输出, 则会产生冗余的存储和开

销。因此对图形流水线的几何着色器进行编程, 对于无意义的虚顶点计算将不被输出到图形流水线的下一个阶段。为了实现多次递归, 为每一个递归层次设计了一个虚顶点缓冲区。在首次递归计算中, 利用 OpenGL 图形流水线的 transform_feedback 功能, 初始化第 1 层的缓冲区。在接下来的递归中, 如图 3 所示, 将第 1 层的顶点缓冲区作为输入, 进行虚顶点计算, 并将计算后有效的虚顶点存入第 2 层顶点缓冲区中。这样地不断迭代直到达到最大递归深度。

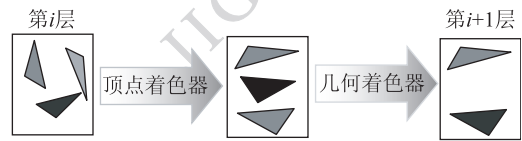


图 3 多次递归虚顶点缓冲区

Fig. 3 Multi-Recursive Virtual Vertex Buffer

算法 2 光子束跟踪递归算法

设 i 为当前递归深度, K 为最大递归深度;

PhotonConeTracing(int i) {

对每个真实或虚拟场景中的反射、折射平面:

1) 开启蒙版和深度测试, 绘制当前反射、折射平面, 设置其覆盖区域的蒙版值为 i ;

2) 计算场景有效虚顶点, 读回结果到顶点缓冲区 i ;

3) 将当前层次计算得到的虚顶点输出到流水线;

4) 开启蒙版和深度测试, 在蒙版值等于 i 的区域进行像素着色处理;

5) 如果 i 小于 K , 执行

PhotonConeTracing($i + 1$);

6) 开启蒙版测试, 绘制当前反射、折射平面, 恢复其覆盖区域的蒙版值为 $i - 1$ 。

}

光子束跟踪算法的性能受反射、折射平面个数和递归深度的影响。当递归深度增加时, 算法计算虚顶点的趟数会呈指数级的增加。但事实上, 不是每个反射、折射平面都是可见的, 也不是每个虚顶点构成的虚反射、折射平面都是可见的, 因此, 在算法设计时, 在算法 2 第 1) 步之后加一个校验判断, 通过 OpenGL 提供的查询, 如果判断其有效覆盖的像素小于 1, 那就没有继续递归的必要, 可以直接结束, 这样可以减少不必要的计算, 进一步提高绘制效

率。同时,本文算法可以正确地计算消隐,通过在算法2的第1)步和第4)步开启深度测试来保证虚拟反射、折射平面之间以及虚拟反射、折射平面与虚拟场景之间的正确遮挡关系。OpenGL的图形流水线提供了硬件加速的光栅化、蒙版测试和深度测试,通过设置蒙版值以及蒙版测试来判断反射、折射面片所覆盖像素,就避免了光线跟踪中的求交以及判断交点是否在面片上的运算,大大提高了性能。不仅如此,对相互独立的每个镜面面片的光束跟踪利用新一代图形流水线进行并行加速^[17],从而进一步提高性能。

1.3 光子路径映射图

光子路径映射图是包含光子路径信息的纹理映射图,它是由位置纹理、法向纹理、光能纹理,以及路径纹理构成。在每一次递归的光束跟踪中,虚顶点表示的几何面片都会输出到图形流水线,经过光栅化和像素着色,光线能够到达的区域在光栅化后就离散地表示为一个个的小像素。在较高绘制分辨率下,可以理解为光子。与一般光子映射图不同,VOBPBT不仅输出光子所在位置和亮度、而且输出法向以及光子所经历的路径信息,并将这些信息保存在纹理中。VOBPBT假设理想的镜面反射、折射,并且对于折射,忽略其少量反射所带来的焦散。为了记录路径,对所有的反射、折射平面进行编号,对于每个镜面片,在完成光束跟踪每次递归时,将当前编号作为参数传入GPU程序,由GPU将这一编号累积记录在虚顶点的颜色属性信息中,虚顶点的颜色属性包含R、G、B、A 4个字段,每个字段最高可以支持32位。通常来说,递归的次数比较少,可以直接利用顶点缓冲区中R、G、B、A 4个分量分别记录第1次递归到第4次递归的所相交的面片编号。

1.4 焦散重建

有了光子路径映射图,焦散重建就是利用这张光子路径映射图,计算出1幅焦散映射图。已有的计算焦散映射图的方法是将光子映射图中的光子投射到一个平面上。光子映射图的分辨率,采样光子的多少,将影响最终的图像质量。如果采样较少,焦散映射图会呈现较多孤立的点,不能收敛;相反,如果要求质量较好,则需要采样较多的光子,降低绘制性能。2009年,Wyman^[2]提出自适应多分辨率投射的方法来缓解这一问题,但他的方法具有两个明显不足:1)由于使用了近似,其光子位置计算的误差

较大;2)该方法在图像空间生成固定数目的光子,在某些欠采样的情况下,光子需要投射的区域范围较大,不仅会降低绘制性能,而且会损失高频光照。

事实上,相邻的光子很有可能在同一个光束中,并且用于表示连续的光强变化。根据光子中保存的路径信息,在光子映射图的邻域范围按照如图4中编号的顺序查找是否存在三角形,如果三角形3个顶点代表的3个光子,它们的路径一致且光子的位置相近(距离小于一定阈值),那么就认为这3个光子之间的三角形区域是连续的,这个三角形就称为焦散三角形,从而可以构建出三角形光束,近似地计算出可见几何表面的焦散光强。如果所有相邻的点都没有形成焦散三角形,那么就将此顶点单独作为一个投射“点”,按照基于图像的光子映射方法,计算其投射半径,输出到图形流水线。

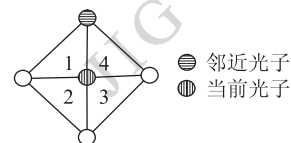


图4 在图像中查找焦散三角形

Fig. 4 Finding caustics triangles

生成的三角形所携带的光强可以用公式 $I = \varphi/A$ 来计算,其中 φ 是进入三角形的光能, A 是投影三角形的面积。这些三角形和投射“点”在流水线中通过顶点变换和光栅化绘制到光源空间的1幅图像上,形成焦散映射图。在最终绘制阶段,对于人眼空间的每个可见像素点,首先将其坐标变换到光源空间,根据变换的坐标查找焦散映射图,就可以计算出这一像素点的焦散光强。

2 实验

实验环境如下:硬件采用 Intel Core i7 处理器 2.67 GHz,3 G 内存,采用 NVIDIA GeForce GTX 560 Ti 显示处理器;软件采用 Windows XP SP3, Visual Studio 2008, GPU 编程语言 GLSL 以及 OpenGL 4.2。本文方法中,所有主要的计算和绘制流程都是在显卡上完成的,实验生成的图像均为 1024×1024 像素分辨率,光子路径映射图均为 1024×1024 像素分辨率。

本文方法可以比较真实地模拟多次递归镜面反

射、折射所产生的焦散效果。图 5 对比了相同折射场景以及绘制条件下,基于图像方法、本文方法以及 POV-Ray 参考方法的结果,前两者的绘制速度分别是 33.39 帧/s 和 59 帧/s,但是本文方法和光子映射得到的结果差异较小,而基于图像的方法,不仅在折射上有较大的失真,而且由于欠采样,造成地面上形成的光斑不连贯。事实上,基于图像的光子映射方法对场景作了一些假设,其算法具有限制^[18],使其不仅难以处理折射体内部镶嵌几何物体的情况(图 6(a)),而且难以处理多次递归反射的情况(图 6(b))。

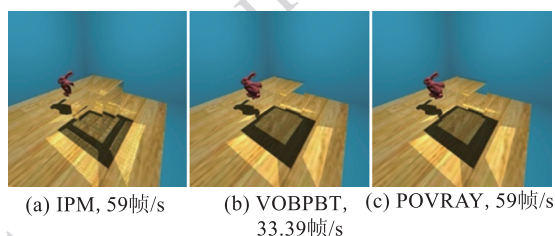
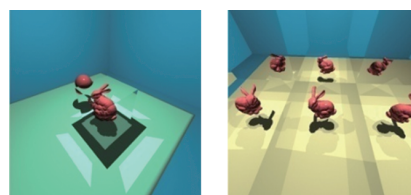


图 5 立方折射体:8 738 面片,递归 2 次,12 个镜面

Fig. 5 Cubic scene with 2 recursive refractions:

8 738 triangles, 12 specular triangles

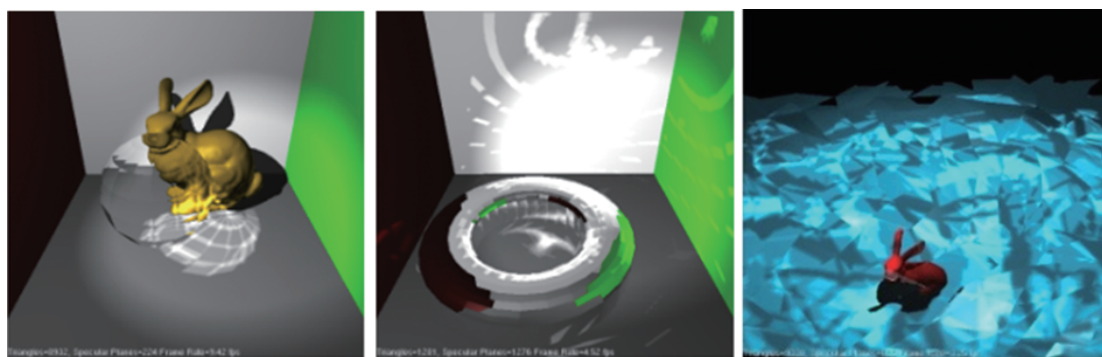


(a) 折射, 30.38 帧/s (b) 反射, 31.97 帧/s

图 6 邻近物体多次递归反射折射

Fig. 6 Multi-reflection & refraction of nearby objects

为了验证本文方法的焦散效果,对更加复杂的反射、折射体模型进行了实验。图 7(a)是由 224 个三角形面片组成的非平滑折射水晶球,采用本文方法不仅在地面上可以得到清晰的焦散效果,同时也可以较真实地在邻近地面的兔子身上形成清晰的焦散光斑,与早期基于光束跟踪绘制水面的方法^[11-12]不同,本文方法充分考虑了遮挡关系。图 7(b)描绘了本文方法应用于包含反射环的场景结果,反射环包含 1 287 个面片,递归一次,其绘制性能为 2.26 帧/s。图 7(c)是本文方法应用于水下焦散绘制结果,可以看出,本文方法可以处理任意形状的折射面,针对完全动态的场景,可以达到可交互的绘制性能。



(a) 8 932 面片, 224 镜面片, 9.98 帧/s (b) 1 281 面片, 1 276 镜面片, 2.26 帧/s (c) 9 330 面片, 625 个镜面片, 13.81 帧/s

图 7 复杂场景反射折射

Fig. 7 Reflection & refraction in complex scenes

实验中发现,本文方法的性能与场景中包含的镜面面片数,以及递归深度有关。增加递归深度以及场景中镜面面片数,会带来性能的下降。对于图 7(a)中的场景,本文实验了不同面片数、递归一次以及递归两次的实验性能,其性能曲线如图 8 所示。由于采用 GPU 硬件加速,以及对递归树的剪枝优化,本文方法的性能呈现趋于平缓地下降,即使对于包含上百个面片的场景,递归 2 次,其绘制性能仍然可以达到每秒 3~4 帧,与基于 OptiX 的光子映射^[1](1~2 s 绘制 1 帧)相比,具有一定的实用价值。

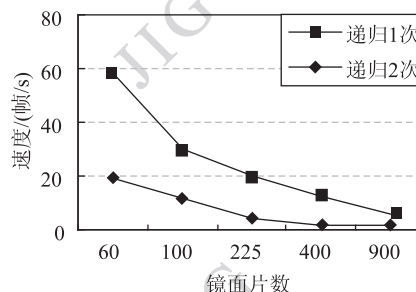
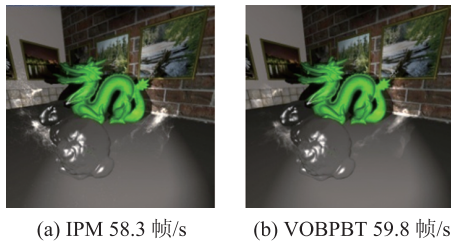


图 8 镜面面片数对于性能的影响

Fig. 8 Performance changes caused by the number of reflective & refractive surfaces

由于性能的原因,本文方法不适合包含几十万面片的镜面反射、折射体,事实上,对于这样的动态场景,经典的GPU加速光子映射算法也不能很快地绘制焦散效果。通过观察发现,对于特别复杂的反射、折射体,由于人眼对于其产生的反射、折射效果不敏感,因此可以容忍近似地反射、折射计算。本文方法也可以和基于图像的光子跟踪相结合来解决复杂反射、折射体的问题。即对反射、折射体进行编号,在基于图像的近似反射、折射计算时将路径信息保存在颜色缓冲区中,构建光子映射图,然后再按照本文方法生成焦散效果。图9对比了本文方法结合图像方法和单纯使用基于图像的光子映射的结果。可以看出,同样使用 1024×1024 个光子进行投射,虽然两者的性能都是在60帧/s左右,但是本文方法得到的结果(图9(b))在保持连贯的同时,也保存了高频特征,具有较强的真实感。



(a) IPM 58.3 帧/s (b) VOBPBT 59.8 帧/s

图9 复杂的折射场景:319 630 镜面片

Fig.9 Complex scene: 319 630 refractive surfaces

3 结 论

快速准确绘制焦散效果的方法,可以处理多次递归反射、折射,能够绘制连贯的高频焦散效果,同时可以达到交互的绘制性能。本文方法的性能会随着折射体的面片数量增加而下降。为了提高性能,对于包含数以万计的面片的复杂反射、折射体,可以和基于图像的光子跟踪相结合。未来的工作可能会包括使用一些空间数据结构来优化算法,通过预处理对折射体的面片进行聚类来减少计算开销等。

参考文献 (References)

[1] Parker S G, Bigler J, Dietrich A, et al. OptiX: a general purpose ray tracing engine [J]. ACM Transactions on Graphics, 2010, 29(4): #66(1-13).

[2] Zhou K, Hou Q, Wang R, et al. Real-time KD-tree construction on graphics hardware [J]. ACM Transactions on Graphics,

2008, 27(5):126(1-11).

[3] Fabianowski B, Dingliana J. Compact BVH storage for ray tracing and photon mapping [C]//Proceedings of Eurographics, Dublin, Ireland, North-Holland, 2009: 1-8.

[4] Wang R, Zhou K, Pan M, et al. An efficient GPU-based approach for interactive global illumination [J]. ACM Transactions on Graphics, 2009, 28(3): 91(1-8).

[5] Wyman C, Davis S. Interactive image-space techniques for approximating caustics [C]//The 2006 Symposium on Interactive 3D Graphics and Games. New York, USA: ACM, 2006: 153-160.

[6] Wyman C. Hierarchical caustic maps [C]//The 2008 Symposium on Interactive 3D Graphics and Games. New York, USA: ACM, 2008: 163-171.

[7] Wyman C, Nichols G. Adaptive caustic maps using deferred shading [J]. Computer Graphics Forum, 2009, 28(2): 309-318.

[8] Heckbert P S, Hanrahan P. Beam tracing polygon objects [J]. Computer Graphics, 1984, 18:119-127.

[9] Ernst M, Akenine-Moller T, Wann Jensen H. Inter-active rendering of caustics using interpolated warped volumes [C]//Proceedings of Graphics Interface. Ontario, Canada: ACM, 2005:87-96.

[10] Wallace J R, Cohen M F, Greenberg D P. A tow-pass solution to the rendering equation: a synthesis of ray tracing and radiosity methods [J]. Computer Graphics, 1987, 21:311-320.

[11] Hu W, Qin K H, Wang H W, et al. Reflections and refraction on implicit surfaces [J]. Journal of Computer Science and Technology, 2006, 21(2): 166-172.

[12] Qin K H, Chang Z Y, Li Y F, et al. Real-time ray tracing for planar scenes with non-linear refractions and reflections [J]. Chinese Journal of Computers, 2002, 25:923-929. [秦开怀, 常正义, 李亚峰, 等. 非线性折射和反射平面场景的实时光线跟踪 [J]. 计算机学报, 2002, 25:923-929.]

[13] Li Y F, Qin K H. Rendering reflections and refractions in object space [J]. Journal of Tsinghua University: Science and Technology, 2005, 45(7): 956-961. [李亚峰, 秦开怀. 景物空间的镜面反射(折射)现象绘制方法 [J]. 清华大学学报(自然科学版), 2005, 45(7):956-961.]

[14] Nishita, T, Nakamae E. Method of displaying optical effects within water using accumulation buffer [C]//Proceedings of Siggraph. New York, USA: ACM, 1994: 373-379.

[15] Iwasaki, K, Dobashi Y, Nishita T. An efficient method for rendering underwater optical effects using graphics hardware [J]. Computer Graphics Forum, 2002, 21(4):701-711.

[16] Iwasaki K, Dobashi Y, Nishita T. A fast rendering method for refractive and reflective caustics due to water surfaces [J]. Computer Graphics Forum, 2003, 22(3): 601-609.

[17] Kalra A. Rendering grass with instancing in DirectX 10 [EB/OL]. [2013-11-10] <http://software.intel.com/en-us/articles/rendering-grass-with-instancing-in-directx-10>, 2011.

[18] Szirmay-Kalos L, Umenhoffer T, Patow G, et al. Specular effects on the GPU: state of the art [J]. Computer Graphics forum, 2009, 26(6):1586-1617.