

中图法分类号: TP391 文献标志码: A 文章编号: 1006-8961(2011)11-1957-10

论文索引信息: 吴佳泽, 郑昌文, 胡晓惠, 欧阳冠军, 张利强. 景深效果绘制技术综述[J]. 中国图象图形学报, 2011, 16(11): 1957-1966

景深效果绘制技术综述

吴佳泽^{1),2)}, 郑昌文¹⁾, 胡晓惠¹⁾, 欧阳冠军¹⁾, 张利强¹⁾

¹⁾ (中国科学院软件研究所综合信息系统技术国家级重点实验室, 北京 100190) ²⁾ (中国科学院研究生院, 北京 100049)

摘要: 景深效果能够增强计算机生成图像的真实感, 因此关于这类效果的绘制已成为计算机图形学中的重要研究方向。针对景深效果的绘制问题, 首先研究和分析各种相机镜头模型的光学成像原理, 比较各类镜头模型的特点和模拟景深效果的能力; 然后分类总结景深效果绘制的基本原理和典型算法, 比较各类景深绘制方法的优缺点。最后指出景深效果绘制技术的应用价值, 并展望未来的研究方向。

关键词: 景深效果; 相机镜头模型; 景深效果绘制方法; 真实感绘制; 实时绘制

Survey of rendering techniques for depth-of-field effect

Wu Jiaze^{1),2)}, Zheng Changwen¹⁾, Hu Xiaohui¹⁾, Ouyang Guanjun¹⁾, Zhang Liqiang¹⁾

¹⁾ (National Key Laboratory of Integrated Information System Technology, Institute of Software, Chinese Academy of Sciences, Beijing 100190 China)

²⁾ (Graduate University of Chinese Academy of Science, Beijing 100049 China)

Abstract: The depth-of-field effect can improve the realistic sense of a computer-generated image, and thus rendering of this effect has become one of the important research topics in computer graphics. Aiming at the problem for rendering the depth-of-field effect, in this paper we first study optical imaging principles which various kinds of camera lens models feature, and compare their characteristics and capabilities of modeling the depth-of-field effect; then the taxonomy of rendering techniques for the depth-of-field effect is presented. Relevant fundamental principles and typical algorithms are concluded, and their advantages and disadvantages are compared in detail. Lastly applications of rendering techniques for the depth-of-field effect are presented and future research trends are foreseen.

Keywords: depth-of-field; camera lens model; depth-of-field rendering approach; realistic rendering; real-time rendering

0 引言

景深效果是相机镜头的重要光学成像特性, 在摄影中作为一种非常重要的艺术工具用于强调所拍摄的对象。在计算机合成的图像中加入景深效果有如下益处: 1) 更加精确地模拟相机镜头的光学成像过程, 增强计算机合成图像的真实感; 2) 增加图像的深度暗示, 有助于理解场景的几何结构和物体的

相对位置; 3) 吸引观察者关注清晰成像的物体而忽略图像中的其他模糊部分, 缓解眼睛疲劳。

自 Potmesil 和 Chakravarty^[1] 在 20 世纪 80 年代初发表关于景深效果绘制的研究论文以来, 景深效果的绘制已成为计算机图形学的重要研究方向, 并出现大量的研究成果, 这些成果已被广泛应用于影视娱乐、信息可视化、眼科治疗与校正等领域。

针对景深效果的绘制问题, 对国内外在该领域的研究成果进行系统研究, 为景深效果绘制技术的

收稿日期: 2010-11-10; 修回日期: 2010-12-15

基金项目: 国家高技术研究发展计划(863)项目(2009AA01Z303)。

第一作者简介: 吴佳泽(1982—), 男。现为中国科学院软件研究所计算机应用技术专业博士研究生, 主要研究方向为计算机图形学、虚拟现实、计算机仿真。E-mail: jiaze08@iscas.ac.cn。

研究提供一个统一的理论框架,为开展进一步的研究提供有价值的参考。首先对现有各种相机镜头模型进行分类,分析和比较它们的优缺点。然后分析和比较各种景深效果绘制技术的基本原理、典型算法和优缺点。最后分析景深效果绘制技术的应用价值,并对未来的研究方向进行了讨论。

1 景深效果的光学原理

景深效果是指相机镜头能够对物体成清晰像的物距范围,在该物距范围之外(包括前景和背景),相机镜头对物体成模糊像^[2]。在物理本质上,景深效果是由于相机镜头的孔径具有一定的尺寸,其光学原理如图 1 所示。图中,只有位于聚焦平面 O 上的点才能在像平面 I 上成清晰的像点,而在聚焦平面之前或之后的点在像平面上成一定大小的弥散圈,其大小(直径)可通过下式计算得到^[3]

$$c = \frac{|U - U_f|f}{U(U_f - f)}D$$

(1)

式中, D 为透镜的直径, f 为透镜的有效焦距, U_f 为聚焦平面到透镜的距离, U 为物距。可以看出,弥散圈的大小随着镜头孔径、有效焦距或物距的增加而增加。当弥散圈的大小足够小,而不能被相机所感知时,依然可以认为是成清晰像。现有的景深效果绘制算法大多基于式(1),假定弥散圈为圆形,且弥散圈内的光强均匀分布,因而绘制出的散景效果相对简单。而实际相机镜头的景深效果非常复杂,其弥散圈通常不是圆形,且弥散圈内的光强分布不均匀。

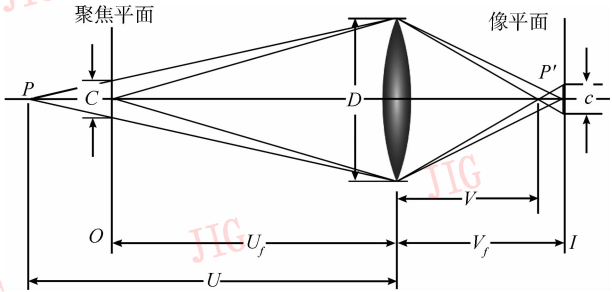


图 1 景深效果的光学原理

Fig. 1 Optical principle of depth of field effect

2 相机镜头模型

计算机图形学领域中,相机镜头模型用于模拟相机镜头的光学成像特性,以描述 3 维场景到 2 维图像的转换过程。目前,存在多种不同的相机镜头模型,包括针孔模型、薄透镜模型、厚透镜模型和几何镜头模型^[4-6],这些模型以不同的精度模拟相机镜头的光学成像特性,具有不同的复杂度,因此可以绘制不同精度的景深效果。

针孔模型(图 2(a))是计算机图形学中使用较为广泛的相机镜头模型之一。该模型假定镜头孔径和厚度无限小,即通过单点成像,该点也称为投影中心,这使得场景中所有物体能同时成清晰像。针孔模型的成像过程可通过一个透视变换矩阵描述^[7]。该模型的优点是原理简单,易于实现,缺点是不能模拟景深效果。

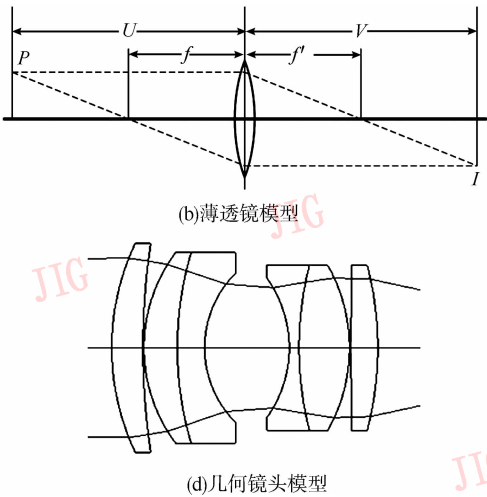
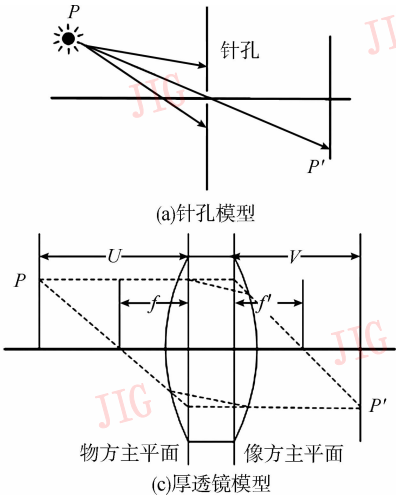


图 2 各种相机镜头模型的光学成像过程

Fig. 2 Optical imaging principle of camera lens models

薄透镜模型(图 2(b))也是计算机图形学中常用的一种相机镜头模型。该模型具有一定大小的孔径和无限小的厚度,是真实镜头的近似模拟。薄透镜模型首次由 Potmesil 等人^[1]在绘制景深效果时引入,随后在真实感绘制和实时绘制领域中得到广泛使用。薄透镜模型能模拟简单的景深效果,而不能模拟真实镜头所导致的复杂的景深效果。厚透镜模型(图 2(c))具有一定大小的孔径和厚度,与薄透镜模型相比,厚透镜模型能够更加精确地模拟相机镜头的光学成像性质,当透镜厚度与焦距相比不可忽略时,常使用该模型。厚透镜模型的光学成像原理如图 2(c)所示,该模型有两个主平面:物方主平面和像方主平面,这两个主平面之间的距离称为厚透镜的厚度。薄透镜和厚透镜模型的成像过程可以用投影矩阵^[8]来描述,这使得折射光线的计算可以用矩阵变换完成,而不是计算量更大的折射定律。

Kolb 等人^[8]结合镜头设计和计算机图形学技术,首次引入几何镜头模型,利用镜头设计领域中标准的相机镜头信息建立镜头模型,使用光线追踪技术模拟相机镜头的光学成像性质。几何镜头模型(图 2(d))需要相机镜头的详细参数,包括各个镜片的相对位置、几何形状、材质属性(折射或反射材料)等。该模型首先获取相机镜头的详细参数,然后利用各个镜片的几何形状和相对位置参数建立相机镜头的空间结构,再结合各个镜片的材质参数以正确模拟光通过各个镜片的过程。通过在镜头内进行光线追踪。该模型能够模拟真实相机镜头的景深效果,由于涉及较大的计算量,常用于真实感绘制,在实时绘制中使用该模型存在不少困难。吴佳泽等人^[3,9]对该模型进行了改进,以绘制出复杂的景深效果(散景效果)。

上述 4 种镜头模型具有不同的复杂度,以不同的精度模拟真实镜头的光学成像特性。这些模型以几何光学理论为基础,支持对景深效果的模拟。实际应用中不同的场合对这些模型有不同的需求,如真实感绘制中常需要非常精确的几何镜头模型,而在实时绘制中,基于性能的考虑,倾向于选择较为简单的针孔模型或薄透镜模型。

3 景深效果绘制方法

Potmesil 等人^[1]1981 年首次提出景深效果绘制算法以来,国内外学者已经提出众多的景深效果绘

制算法。这些绘制算法大致可以分为 5 类^[10]:基于分布式光线追踪的绘制方法、基于图像累积的绘制方法、基于深度分层的绘制方法、基于前向映射的绘制方法和基于逆向映射的绘制方法。分布式光线追踪方法、图像累积方法和深度分层方法都可以看成是组合技术,难以与现有的图形引擎集成。前向映射方法和逆向映射方法都是基于 Z 缓存(深度缓存)的方法,能够充分利用图形硬件的加速能力。深度分层方法是一类特殊的方法,其本质上是一种组合技术,但常在对深度进行分层时利用 Z 缓存中的信息。

景深效果绘制方法在图像质量方面主要围绕强度渗漏、部分遮挡和离散化 3 个问题展开,力争全部或部分解决这些问题以获得高质量的景深效果。强度渗漏问题指模糊的前景或背景融合到聚焦物体中,或聚焦物体融合到模糊的前景或背景中,导致不一致的深度暗示,干扰观测者对图像的正确理解;部分遮挡指物体通过镜头孔径面上不同点成像时具有不同的可见性;离散化则是深度分层方法中的特有问题,对跨越多层的物体进行模糊时,由于子层之间深度的不连续性而出现的块状或带状效果。

3.1 分布式光线追踪

基于分布式光线追踪的景深效果绘制方法是一种多通道绘制方法,需要在像平面和镜头孔径平面上进行多次采样,以获得多个光线样本。该方法的关键技术包括光线追踪技术和随机采样技术^[11],光线追踪技术精确模拟光能在 3 维场景的传输过程以及和物体的交互过程,而随机采样技术用于改进光线追踪技术所绘制图像的质量。

Cook 等人^[12]首次提出基于分布式光线追踪的景深效果绘制算法,但采用的是薄透镜模型,不能真实模拟相机镜头的光学成像过程。因此, Kolb 等人^[8]利用镜头设计领域中标准的相机镜头参数建立几何透镜模型,通过在镜头内部和 3 维场景中进行分布式光线追踪,绘制出更加真实的景深效果。以上算法都采用光线追踪技术的逆向形式,即从像平面到场景对光线进行跟踪。逆向光线追踪的最大缺点是:每次调焦时需要重新进行光线与场景的相交计算。针对这一缺点,吴向阳等人^[13]提出基于正向光线追踪技术的景深效果绘制算法,从场景到像平面进行光线追踪,在重新调整相机参数时,利用光线连贯性和参数改变前的相关数据,加快景深效果的绘制。

上述算法为获得高质量的景深效果,在像平面和镜头孔径面上采样较多的点,计算量巨大。针对这一问题,Soler 等人^[14]提出一种自适应的景深效果绘制算法。绘制聚焦区域时,由于图像清晰,含有较多的细节,在像素上采样较多的点,而通过镜头孔径的辐射度来自场景的一个较小的局部区域,变化较小,在镜头孔径上只需采样较少的点;绘制模糊区域时,由于图像模糊,细节较少,在像素上只需采样较少的点,而通过镜头孔径的辐射度来自场景的范围较大的区域,变化较大,在镜头孔径上需采样较多的点。与一般的分布式光线追踪景深效果绘制算法相比,该算法的绘制效率得到了极大提升。

为了模拟更加复杂的景深效果(即散景效果,此时弥散圈形状不再是圆形的,而弥散圈内的光强分布不再是均匀的),Buhler 和 Wexler^[15]使用一个可以任意指定的概率密度函数对镜头孔径面上的样本进行抖动,以模拟弥散圈内的任意光强分布,但该方法是一种基于现象的模拟方法。吴佳泽等人^[3,9]对几何镜头模型进行改进,并结合分布式光线追踪技术,提出一种基于物理的复杂景深效果真实感绘制算法。该算法考虑镜头光阑和光学像差的影响,精确模拟相机镜头的光学成像特性,绘制出具有不同弥散圈形状和光强分布的景深效果。

分布式光线追踪方法需要在像平面和镜头孔径平面上随机采样,因此绘制的图像不存在强度渗漏和离散化问题,能够真实表现部分遮挡效果,绘制的景深效果最为精确,但其计算量巨大,不适合实时绘制,常用于离线场合,如电影特效等。当光线样本较少时,绘制的图像中会含有较多的噪声,但不会出现其他方法中常见的条带和重影现象(这类现象比噪声更加影响图像质量)。此外,该类方法可以在时间和空间上抖动光线样本,因此比其他方法更加容易绘制出运动模糊、软体阴影等其他真实感效果。

3.2 图像累积

基于图像累积的景深效果绘制方法可看成是基于累积缓存的分布式光线追踪方法,主要针对分布式光线追踪方法计算量大的缺点而提出。该方法的基本原理是:在镜头孔径面上选定若干个投影中心,每个投影中心对应一个投影变换,因此可以绘制多副图像,然后利用累积缓存精度相对较高的特点,累积多幅图像,以绘制最终的图像。

Haeberli 和 Akeley^[16]最先提出基于图像累积的景深效果绘制算法。该算法使用薄透镜模型多次绘

制场景,每次在镜头孔径面上选取多个固定的样本点作为投影中心,每个投影中心构成一个投影变换,而每个投影变换可以绘制一副图像,然后利用累积缓存对多次绘制的结果进行累积,即可得到具有景深效果的图像。为获得可以接受的景深效果,该算法所需的绘制次数与弥散圈的大小成正比。当绘制次数较少时,在图像中最模糊的地方就会出现重影、对象复制等不期望的效果。Lee 等人^[17]利用基于图像的场景表示法近似不同的视图,以避免 Haeberli 所提算法对场景进行多次绘制的缺点,同时利用早期裁剪、近似圆锥跟踪以及抖动采样技术提高绘制速度和改进图像质量。

Heidrich 等人^[4]对 Haeberli 算法进行改进,引入几何镜头模型,在镜头孔径面上选择多个近似的投影中心,将每个投影变换绘制的图像看成是 4D 光场^[18]的一个 2D 切片,将光场绘制技术用于景深效果绘制,同时利用硬件加速透视投影过程,利用光线空间的相关性加速光场采样,以获得较高的交互性。Isaksen 等人^[19]提出新的动态光场参数化技术,以支持可控的景深效果。新的动态光场参数化技术支持非平面的聚焦平面,能够绘制出不受真实相机成像过程限制的景深效果。Yu 等人^[20]使用 CUDA 在 GPU 上实现了 Isaksen 所提出的景深算法,改进了实时性能。

与分布式光线追踪方法相比,该类方法牺牲部分图像质量,以提高绘制速度,如果结合图形硬件的快速处理能力,则可以获得实时或近实时的绘制性能。在性能提升的同时,该方法还可以与几何透镜模型结合,以获得更加精确的景深效果。此外,该类方法可以正确处理部分遮挡效果,能够绘制出精确的景深效果。但是为绘制出高质量的景深效果,尤其是当弥散圈较大时,为避免重影现象,常需要较多的绘制次数,既容易导致累积缓存的精度问题,又难以达到实时性能。

3.3 前向映射

基于前向映射的景深绘制方法的基本思想是:遍历源图像(针孔模型绘制的图像),将源像素按某种方式映射(散射)到目标图像中。该类方法大致可分为 4 个基本步骤:1)利用针孔模型绘制源图像,源图像中包含颜色、深度和其他必要的信息;2)根据镜头孔径尺寸、镜头焦距和源图像中的深度信息计算每个源像素(源图像中的像素)的弥散圈大小,以及确定合适的强度分布函数,如均匀分布函

数、Lommel 分布函数、高斯分布函数等;3)根据弥散圈大小和强度分布函数,对每个源像素模糊处理;4)在目标图像中以某种方式进行混合以累积源图像的模糊处理结果,最终生成含有景深效果的图像。

Potmesil 和 Chakravarty^[1]首次提出基于前向映射的景深效果绘制算法。针对源图像中的每个像素进行模糊处理,弥散圈的大小由该像素的深度决定。利用菲涅耳衍射理论推导出 Lommel 函数,以计算弥散圈内的强度分布。为加速景深效果的生成,并不直接计算 Lommel 函数值,而是通过表查询获取 Lommel 函数值。Chen^[21]根据光粒子理论^[22]和相邻像素相互扩散的现象,以均匀强度分布函数取代 Lommel 函数。

上述算法均涉及大量的计算,不能用于实时绘制,也不能正确解决部分遮挡问题。为在绘制景深效果时考虑部分遮挡效果,Shiya^[6]提出光线分布缓存技术,光线分布缓存本质上是一个子像素结构,每个像素含有若干个 RDB 单元,每个 RDB 单元存储颜色和深度信息。该算法解决 Potmesil 算法所不能解决的部分遮挡问题,且速度远快于 Potmesil 算法。然而,该算法的计算量和内存消耗仍然很大,且缺乏图形硬件的支持,不能达到交互性能。

为了降低计算量,Fearing^[23]扩展 Potmesil^[1]的工作,赋予每个像素一个重要性值,以表明像素更新的优先级,利用连续帧之间的相关性,仅更新那些优先级较高的像素。当场景中只有少量物体运动时,该算法避免每帧都重新处理图像的全部或大部分像素,大大加速景深效果的生成,使得在实时绘制中应用景深效果成为可能。然而,当相机视点或聚焦位置频繁改变时,该算法需要重新生成每帧图像的全部像素,导致实时绘制性能大大降低。

Křivánek 等人^[24]对 EWA surface splatting 技术^[25]进行扩展,提出用于点模型的景深效果绘制算法。利用高斯过滤核对单个重采样核进行模糊以获得景深效果,高斯过滤核的标准差与弥散圈大小成正比。该算法能够较好地处理部分遮挡问题,而且在绘制景深效果时,能够根据弥散圈大小选择相应的 LOD,使得绘制速度不依赖于弥散圈的大小,即使在模糊量很大时也不会导致性能的显著降低。

与逆向映射方法相比较(3.4 节介绍),该类方法不会将清晰像素散射到相邻像素,因此不会导致强度渗漏现象,这是该方法一个显著优点。但该方法需要非常耗时的深度排序技术,在最后的累积

阶段需要图形硬件具有较高的精度,且不能很好地解决部分遮挡问题。

3.4 逆向映射

基于逆向映射的景深效果绘制方法是目前被研究最多的一类方法。该方法的基本原理是:遍历目标图像,计算每个目标像素的弥散圈大小,收集该弥散圈内所有源像素的信息。该方法利用深度缓存中的深度信息计算出弥散圈的大小,然后对针孔模型绘制出的图像进行模糊处理,即可模拟出景深效果。

针对 Potmesil 算法^[1]速度较慢的缺点,Rokita^[26-27]最先提出基于逆向映射的景深效果绘制算法。该算法使用多个较小的过滤核(3×3)对针孔模型绘制的图像进行过滤处理,每个像素的弥散圈大小决定在该像素上过滤核的数目。使用较小的过滤核可以充分利用硬件过滤器的快速计算能力,使得景深效果可以用于实时领域中。

针对单纯使用过滤技术存在的缺点,Riguer 等人^[28]在绘制景深效果时,提出预模糊技术和可分离的高斯过滤技术,其中预模糊技术是对源图像进行亚采样以获得压缩的模糊图像,而可分离的高斯过滤技术则是用两个 1 维的高斯过滤核替代 2 维高斯过滤核以提高过滤速度。由于两个 1 维卷积的计算速度要远快于一个 2 维卷积,因此该算法的绘制速度得到极大的提升,但未能很好地解决强度渗漏问题。

针对强度渗漏问题,Bertalmio 等人^[29]将对源图像的模糊过程看成是用二阶偏微分方程描述的非均匀热扩散过程,提出用数值方法求解偏微分方程,解决大部分强度渗漏问题和部分遮挡问题。由于偏微分方程的数值求解能够并行进行,因此可以充分利用 GPU 的并行加速能力获得实时性能。但该算法的计算复杂度与弥散圈大小成正比,在实时绘制中这是应该避免的,且不能解决背景的去遮挡问题。为进一步加快热扩散偏微分方程的求解过程,Kass 等人^[30]提出用交替方向隐式方法求解偏微分方程,并在 GPU 上实现该方程的并行求解。交替方向隐式求解方法将模糊过程看成是可分离的空间变化递归过滤过程,使得大过滤核的卷积能在常数时间内完成。该算法考虑了聚焦物体和离焦物体的边界,可以得到高质量的景深效果,在静态场景中实现景深效果的交互级生成,但在动态场景中难以获得交互级的景深效果。

Lee 等人^[31]结合非均匀扩散技术和预模糊技

术的优点,提出非均匀过滤的 mipmap 插值技术。针对预模糊技术难以解决强度渗漏问题的缺点,使用非均匀高斯过滤核构造 mipmap 图像,大量减少了强度渗漏现象。该算法对模糊度而不是深度进行平滑,以减少前景模糊的不连续性现象。

该类方法在模糊过程中充分利用 GPU 快速的图像插值和纹理查找能力,可实现景深效果的实时绘制,其绘制速度是所有方法中最快的。尽管该类方法在解决强度渗漏(清晰的像素与背景中的像素混合)问题上取得一些进步,但与前向映射方法相比,仍有不少不足,图像质量相对较差,且与前向映射方法一样,不能很好地考虑部分遮挡效果。

3.5 深度分层

基于深度分层的景深效果绘制方法是一种基于 2D(由于分层的使用,有时也称为 2.5D^[32])的近似深度排序方法。这类方法的基本思路可分为 3 步:1)按照深度将场景或图像分成若干层;2)对各层进行模糊处理;3)将各层的模糊处理结果进行合成,即可得到具有景深效果的图像。

基于深度分层的景深效果绘制算法最初由 Scofield^[32]提出,该算法采用基于物方空间的深度分层技术。将场景对象按深度划分到前景组和背景组中,独立进行绘制,然后使用低通滤波器对前景和背景图像进行模糊,最后进行合成以生成景深效果。该算法是一种对象级粒度的物方空间分层技术,不能对对象的一部分进行模糊处理,也不能对对象的表面进行平滑的模糊处理。当图像中含有在深度方向上很长的物体时,绘制的图像会出现离散化问题。当 3 维场景非常复杂时,景深效果的绘制速度非常慢。针对这一问题,Mulder 和 Lieke^[33]提出基于像方空间深度分层的景深效果绘制算法,将针孔模型绘制的图像按深度分为前景和背景两层。与基于场景对象分层的算法相比,该算法的计算复杂度与 3 维场景无关,极大地提高了景深效果的生成速度。

Barsky 等人^[34]指出,人眼对模糊的分辨能力与深度成非线性关系,而与屈光度近似成线性关系,在对深度分层时,各层之间的深度差应考虑这一因素,因此提出一种基于人眼镜头模型的景深效果绘制算法。该算法能够很好地解决强度渗漏问题,但未能解决离散化问题。针对这一缺点,Barsky 等人^[35]使用边缘检测技术和邻接像素差分技术识别图像中的物体,辅助子图像的分层,更好地解决部分遮挡问题和离散化问题。尽管 Barsky 提出的一系列算法改

进了图像质量,但其最大的缺点是计算量较大,难以达到实时绘制的要求。

为进一步改进图像质量,提高绘制速度,Kraus 和 Strengert^[36]对基于 GPU 的金塔形图像处理技术^[37]进行扩展,以加速 GPU 上的关键图像操作,如去遮挡操作、模糊操作等。该算法通过对相邻子图像间的平滑处理有效地减少离散化问题,且当图像和弥散圈的尺寸都不太大时,可以用于实时绘制。

Barsky^[34]和 Kraus^[36]等提出的算法都是通过对被遮挡区域的外部边界进行外推来模拟部分遮挡效果,外推技术只能获得被遮挡区域的近似颜色,而不能获得精确的颜色值,尤其当弥散圈很大时,外推技术不再获得较佳的部分遮挡效果。而且这些算法都采用基于整个图像的全局分层技术,不能很好地解决因长物体跨越多层而导致的离散化问题。针对这些缺点,Lee 等人^[38]提出单像素的局部分层技术和绘制隐藏图像的特殊着色器技术,单像素的局部分层技术进一步减少离散化问题,特殊的着色器技术用于绘制被遮挡的场景,以获得非常精确的部分遮挡效果。

该类方法主要是针对基于单层的后处理方法(前向映射和逆向映射方法)不能很好处理部分遮挡效果而提出的,常采用两种技术获得被遮挡区域的场景信息(颜色、深度等):第 1 种是对被遮挡区域的外边界进行外推,该技术可以获得被遮挡区域的近似颜色,较好地解决部分遮挡问题,对大部分场景来说,不会导致明显的问题;第 2 种是使用特殊的着色器绘制被遮挡的场景信息,该技术可以获得被遮挡区域的精确颜色,更好地解决部分遮挡问题。当场景中,尤其是在前景中,存在跨越较大深度范围的物体时,这些物体不会表现出明显的景深效果,且在物体之间的模糊量变化很快。当场景中的物体在深度上互相重叠时,分层方法难以实现。尽管可以沿着深度拆分场景,但是这样会导致更加难以解决的离散化问题。与单层的方法相比,该类方法能够更好地解决强度渗漏,获得更好的前景模糊质量,但是在层之间得到平滑的模糊过渡,需要的层数较多,实时性能明显不如单层方法。

3.6 各类方法的比较

表 1 总结了各类景深效果绘制方法在输入信息、算法实现、运行性能和输出图像质量等方面的典型特点。可以看出,不同的景深绘制方法在镜头模型的选择上有所不同,对光传输模拟的精度各有取

舍,在真实性和实时性方面作出不同的权衡,与 GPU 加速能力的匹配性不尽相同。总而言之,由于分布式光线追踪和图像累积方法属于多遍绘制方法,因此能够绘制出非常精确的景深效果,通常适用于电影等真实感要求较高的场合。而前向映射、逆向映射和深度分层方法都是对单幅针孔图像进行某种图像后处理操作以获得景深效果,因此缺乏足够精确的场景信息,其绘制结果是近似的。即使采用许多技术以增强图像质量,依然无法比拟前两种方法的绘制结果,但具有很高的实时性能,常用于计算机游戏、动画、虚拟现实等实时领域。随着绘制技术和硬件计算能力的进步,以及人们对真实感的无止境追求,分布式光线追踪和图像累积方法在未来的研究中将占有更加重要的地位,例如, Lee 等人^[17]首次在 GPU 上实现基于图像累积方法的景深效果实时绘制,即是朝着这方面前进的有益尝试,而分布式光线追踪方法因其巨大的计算复杂度,将成为未来需要突破的研究热点。

表 1 景深绘制方法的比较

Tab.1 Comparison of depth of field rendering approaches

比较指标	光线追踪	图像累积	前向映射	逆向映射	深度分层
深度	精确	精确	不精确	不精确	不精确
GPU 适配性	差	较佳	差	最佳	较佳
场景复杂度	依赖	依赖	不依赖	不依赖	不依赖
实时性	差	差	较佳	最佳	较佳
强度泄露	不存在	不存在	存在	严重	存在
部分遮挡	最佳	较佳	差	差	较佳
离散化	不存在	不存在	不存在	不存在	存在
最终结果	最佳	较佳	差	差	较佳

4 应用价值

4.1 影视娱乐

真实感绘制和实时绘制的结果中都需要加入景深效果以增强结果的真实感,这些结果既可用于电影、动画等需要高度真实感的场合,也可用于计算机游戏、虚拟现实等需要高度实时性的场合。在电影与动画中,通过加入景深效果,不断调整聚焦位置,使得观众对场景的注意力按导演的意愿不断的转移。而且复杂的景深效果也常作为一种特效在电影和动画中被广泛使用以增强视觉效果。在游戏中,为了使玩家关注于场景特定部分,以便能够集中精

力完成特定的操作,可以对其他不重要的部分引入景深效果,这样还可以缓解玩家的视觉疲劳。在增强现实和虚拟工作室中,通过尽可能地模拟真实镜头的光学成像特性,获得真实的景深效果,使得真实世界和虚拟世界的对象能够被无缝地合成到单一的图像中。

4.2 信息可视化

在信息可视化领域中,景深效果可以使得所关注的重要数据被清晰地可视化,而其他不重要的部分则被模糊处理。传统景深效果绘制算法的主要目标是尽可能真实地模拟相机拍摄的景深效果,相机的景深效果受成像过程和其他物理条件的约束,如镜头焦距、镜头孔径或物距等,不能实现任意的景深效果,即对图像中的任意部分进行聚焦控制。而在信息可视化领域中,这是一种常见的需求。Kosara 等人^[39]首次提出语义景深的绘制,将景深效果用于信息可视化领域。对 Scofield^[32]提出的景深绘制算法进行扩展,使得场景中的对象根据相关性而不是深度进行可视化。语义景深技术使得用户可以关注场景中任何感兴趣的部分,而不影响可视化的几何和其他特征。由于语义景深技术是从模拟人眼成像过程的景深效果绘制技术发展而来,因此在可视化中使用该技术非常符合人眼的视觉特点,容易被用户理解与接受。然而,与 Scofield 算法一样,该技术只支持对象级的景深控制,不能够仅聚焦对象的一部分,而模糊对象的其他部分。针对上述问题, Kosloff 和 Barsky^[40]对 Barsky 等人^[34-35,41]提出的深度分层技术和 Bertalmío^[29]等人提出的非均匀热扩散技术进行扩展,提出通用化的景深效果绘制算法,以致景深效果不再受到焦距、物距和孔径的限制,而是由场景所生成的 3 维模糊场所控制,场景中的任意点都可以被独立地模糊。

4.3 眼科治疗与校正

景深效果绘制技术能够正确模拟人眼的光学成像过程,真实绘制出人眼观察世界的结果,在眼科治疗领域中具有重要的应用价值。Mostafawy 等人^[42]利用分布式光线追踪技术和几何镜头模型评估人眼的屈光误差。Barsky 等人^[34]提出基于视觉真实感的绘制技术,将基于深度分层的景深效果绘制技术应用到眼科手术中,眼科专家可通过查看用该技术绘制的图像,来了解病人的视觉缺陷。如在对人眼进行的激光治疗手术中,通过使用光学像差测量设备获得病人眼睛的成像特性数据,再利用这些数据

建立病人手术前后的镜头模型,并使用建立的模型绘制出手术前后的视觉图像,可辅助眼科医生对病人进行有针对性的治疗。Kakimoto 等人^[43]将景深效果绘制技术应用到眼镜镜片设计中,首先模拟人眼的景深效果,以及近视眼和远视眼的像散效果,再模拟在人眼前加入眼镜镜片后的成像效果,以检验眼镜的校正作用。

5 结 论

从几何光学出发,系统归纳和总结了与景深效果绘制相关的研究文献,从相机镜头模型、景深效果绘制技术和应用领域等多方面进行了探讨。尽管景深效果绘制技术已经被应用到多个领域,但在真实性和实时性方面还需要进一步的改进。为促进景深效果绘制技术的进一步实用化,未来的研究工作应该包括以下两个方面:

1) 更加精确的相机镜头模型 随着硬件技术和绘制技术的进步,实时绘制和真实感绘制对镜头模型的精度要求将会越来越高,因而几何镜头模型具有更大的研究和应用前景。然而,现有的几何模型依然存在一些不足:在镜头光学表面的建模方面,现有的几何镜头模型只考虑了折射球面,没有考虑反射镜面,更没有考虑二次或高次曲面,这些曲面常出现在复杂光学镜头中;在光学镜头内的光学现象建模方面,现有几何镜头模型仅考虑光线经过分界面时发生的折射,没有考虑与折射同时发生的反射和散射现象,也没有考虑与波长相关的色散现象;在模拟人眼时,没有从光学和生物解剖学两方面进行精细建模。因此,未来的镜头模型研究需要从表面形状、光学、生物解剖学等多个方面进行更加精细的建模,这种精细的镜头模型能够模拟更加精确的景深效果,可用于支持图形学领域所追求的照片级或视觉级真实感;

2) GPU 加速的景深效果绘制 近年来,GPU 的并行计算能力与可编程能力得到大幅度的提升,为提高景深绘制技术的真实性和实时性提供了良好的硬件平台。在景深效果绘制算法中,部分或全部地使用 GPU 加速能极大提高算法的效率,甚至使得较复杂的绘制算法可达到实时性能。许多研究人员对传统的基于 CPU 的绘制算法进行改进,提出了许多利用 GPU 加速的绘制算法,主要集中在前向映射、逆向映射和深度分层 3 类方法上。以上这些研

究成果都是使用标准的图形编程接口(DirectX 或 OpenGL)和着色语言(HLSL、cg 或 GLSL 等)联合实现的,在性能上得到很大的提升。然而,基于效率的考虑,图形绘制管线的一些阶段仍然是可配置而不是可编程的,GPU 的并行计算能力并没有得到完全的释放,因而几乎没有基于分布式光线追踪或图像累积方法的研究工作。唯一的例外是 2009 年 SIGGRAPH Asia 大会上 Lee 等人^[17]提出基于片元着色器的镜头光线采样技术,首次在 GPU 上实现了图像累积方法。最近,以 NVIDIA 的 CUDA 架构和 ATI 的 Stream SDK 架构为标志,GPU 完全可编程变成现实,GPU 的计算能力得到彻底释放。这些架构使得基于分布式光线追踪的景深效果实时绘制成为可能,这是因为光线追踪具有天然的并行特性,能够充分利用 GPU 的并行计算能力。NVIDIA 公司利用 CUDA 开发出首个光线追踪引擎 OptiX,该引擎提供了可扩展的框架以支持基于光线追踪的交互式应用。此外,Apple 提出了异构计算语言 OpenCL,更进一步将多核 CPU 和 GPU 无缝地集成为一个强大的并行计算平台。开源渲染器 LuxRender 的开发团队利用 OpenCL 开发出一个基于 GPU 的光线追踪库,并在此基础上开发了一个基于 GPU 加速的渲染器原型系统。因此,可以预见未来利用 GPU 加速的分布式光线追踪景深效果实时和非实时绘制将是一个研究热点。

参考文献(References)

- [1] Potmesil M, Chakravarty I. A lens and aperture camera model for synthetic image generation[C]//Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH. New York: ACM, 1981:297-305.
- [2] Zhang Yimo. Applied Optics[M]. 3rd ed. Beijing: Publishing House of Electronics Industry, 2008. [张以谟. 应用光学[M]. 3 版. 北京:电子工业出版社, 2008.]
- [3] Wu J Z, Zheng C W, Hu X H, et al. Realistic rendering of bokeh effect based on optical aberrations[J]. The Visual Computer, 2010, 26(6):555-563.
- [4] Heidrich W, Slusallek P, Seidel H P. An image-based model for realistic lens systems in interactive computer graphics[C]//Proceedings of Graphics Interface. Toronto, Canada: Canadian Information Processing Society, 1997:68-75.
- [5] Barsky B A, Horn D R, Klein S A, et al. Camera models and optical systems used in computer graphics (part i): object-based techniques[C]//Proceedings of the International Conference on Computational Science and Its Applications. Berlin: Springer,

- 2003;246-255.
- [6] Shinya M. Post-filtering for depth of field simulation with ray distribution buffer [C] // Proceedings of Graphics Interface. Toronto, Canada; Canadian Information Processing Society, 1994; 59-66.
- [7] Möller T A, Haines E. Real-time Rendering [M]. 2nd ed. Natick, Massachusetts; A K Peters, 2004. [Möller T A, Haines E. 实时计算机图形学 [M]. 2 版. 普建涛, 译. 北京: 北京大学出版社, 2004.]
- [8] Kolb C, Mitchell D, Hanrahan P. A realistic camera model for computer graphics [C] // Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH. New York; ACM, 1995; 317-324.
- [9] Wu Jiaze, Zheng Changwen, Hu Xiaohui, et al. Realistic rendering of bokeh effect [J]. Journal of Computer Aided Design & Computer Graphics, 2010, 22 (5): 746-752, 761. [吴佳泽, 郑昌文, 胡晓惠, 等. 散景效果的真实感绘制 [J]. 计算机辅助设计与图形学学报, 2010, 22 (5): 746-752, 761.]
- [10] Hammon J E. Practical Post-Process Depth of Field [M]. Nguyen H. GPU Gems 3. Boston; Addison-Wesley, 2007; 583-606.
- [11] Cook R L. Stochastic sampling in computer graphics [C] // Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH. New York; ACM, 1986; 51-72.
- [12] Cook R L, Porter T, Carpenter L. Distributed ray tracing [C] // Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH. New York; ACM, 1984; 137-145.
- [13] Wu Xiangyang, Bao Hujun, Chen Wei, et al. Real-time simulation of camera imaging by forward ray tracing [J]. Journal of Computer Aided Design & Computer Graphics, 2005, 17 (7): 1427-1433. [吴向阳, 鲍虎军, 陈为, 等. 采用正向光线跟踪的照相机成像实时模拟 [J]. 计算机辅助设计与图形学学报, 2005, 17 (7): 1427-1433.]
- [14] Soler C, Subr K, Durand F, et al. Fourier depth of field [J]. ACM Transactions on Graphics, 2009, 28 (2): 18.
- [15] Buhler J, Wexler D. A phenomenological model for bokeh rendering [C] // Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH Abstracts and Applications. New York; ACM, 2002; 142.
- [16] Haeberli P, Akeley K. The accumulation buffer; hardware support for high-quality rendering [C] // Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH. New York; ACM, 1990; 309-318.
- [17] Lee S, Eisemann E, Seidel H P. Depth-of-field rendering with multiview synthesis [J]. ACM Transactions on Graphics (Proceedings of the ACM SIGGRAPH Asia conference), 2009, 28 (5): 1-6.
- [18] Levoy M, Hanrahan P. Light field rendering [C] // Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH. New York; ACM, 1996; 31-42.
- [19] Isaksen A, McMillan L, Gortler S J. Dynamically reparameterized light fields [C] // Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH. New York; ACM, 2000; 297-306.
- [20] Yu X, Wang R, Yu J. Real-time depth of field rendering via dynamic light field generation and filtering [J]. Computer Graphics Forum, 2010, 29 (7): 2099-2107.
- [21] Chen Y C. Lens effect on synthetic image generation based on light particle theory [J]. The Visual Computer, 1987, 3 (3): 125-136.
- [22] Born M, Wolf E. Principles of Optics [M]. 7th ed. Cambridge; Cambridge University Press, 1999.
- [23] Fearing P. Importance ordering for real-time depth of field [C] // Proceedings of the Third International Conference on Computer Science on Image Analysis Applications and Computer Graphics. Berlin; Springer, 1995; 372-380.
- [24] K rivánek J, Žár J, Bouatouch K. Fast depth of field rendering with surface splatting [C] // Proceedings of Computer Graphics International. New York; ACM, 2003.
- [25] Zwicker M, Pfister H, Baar J V, et al. Surface splatting [C] // Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH. New York; ACM, 2001; 371-378.
- [26] Rokita P. Fast generation of depth-of-field effects in computer graphics [J]. Computers & Graphics, 1993, 17 (5): 593-595.
- [27] Rokita P. Generating depth-of-field effects in virtual reality applications [J]. IEEE Computer Graphics and its Application, 1996, 16 (2): 18-21.
- [28] Riguer G, Tatarchuk N, Isidoro J. Real-time Depth of Field Simulation [M] // Engel W F. Shader X²; Shader Programming Tips and Tricks with DirectX 9. Plano; Wordware, 2003; 529-556.
- [29] Bertalmío M, Fort P, Sánchez-Crespo D. Real-time, accurate depth of field using anisotropic diffusion and programmable graphics cards [C] // Proceedings of Second International Symposium 3D Data Processing, Visualization, and Transmission. Washington; IEEE, 2004; 767-773.
- [30] Kass M, Lefohn A, Owens J. Interactive Depth of Field Using Simulated Diffusion on a GPU [R]. Emeryville; Pixar Animation Studios, 2006.
- [31] Lee S, Kim G J, Choi S. Real-time depth-of-field rendering using anisotropically filtered mipmap interpolation [J]. IEEE Transactions on Visualization and Computer Graphics, 2009, 15 (3): 453-464.
- [32] Scofield C. 2.5d Depth-of-Field Simulation for Computer Animation [M] // Kirk D. Graphics Gems III. San Francisco; Morgan Kaufmann, 1992; 36-38.
- [33] Mulder J D, Liere R V. Fast perception-based depth of field rendering [C] // Proceedings of the ACM Symposium on Virtual Reality Software and Technology. New York; ACM, 2000; 129-133.
- [34] Barsky B A. Vision-realistic rendering; simulation of the scanned foveal image from wavefront data of human subjects [C] // Proceedings of the First Symposium on Applied Perception in

- Graphics and Visualization. New York: ACM, 2004: 73-81.
- [35] Barsky B A, Tobias M J, Chu D P, et al. Elimination of artifacts due to occlusion and discretization problems in image space blurring techniques [J]. Graphical Models, 2005, 67(6): 584-599.
- [36] Kraus M, Strengert M. Depth-of-field rendering by pyramidal image processing [J]. Computer Graphics Forum, 2007, 26(3): 645-654.
- [37] Strengert M, Kraus M, Ertl T. Pyramid methods in GPU-based image processing [C] // Workshop on Vision, Modelling, and Visualization. Aire-la-Ville, Switzerland: Eurographics Association, 2006: 169-176.
- [38] Lee S, Kim G J, Choi S. Real-time depth-of-field rendering using point splatting on per-pixel layers [J]. Computer Graphics Forum, 2008, 27(7): 1955-1962.
- [39] Kosara R, Miksch S, Hauser H. Semantic depth of field [C] // Proceedings of the IEEE Symposium on Information Visualization. Washington: IEEE, 2001: 97-104.
- [40] Kosloff T J, Tao M W, Barsky B A. Depth of field postprocessing for layered scenes using constant-time rectangle spreading [C] // Proceedings of Graphics Interface. Toronto, Canada: Canadian Information Processing Society, 2009: 39-46.
- [41] Barsky B A, Bargteil A W, Garcia D D, et al. Introducing vision-realistic rendering [C] // Proceedings of Eurographics Rendering Workshop. Aire-la-Ville. Switzerland: Eurographics, 2002: 26-28.
- [42] Mostafawy S, Kermani O, Lubatschowski H. Virtual eye: retinal image visualization of the human eye [J]. IEEE Computer Graphics and Applications, 1997, 17(1): 8-12.
- [43] Kakimoto M, Tatsukawa T, Mukai Y, et al. Interactive simulation of the human eye depth of field and its correction by spectacle lenses [J]. Computer Graphics Forum, 2007, 26(3): 627-630.