

1-15-2021

Real-time Shadow Algorithm Improvements in Digital Earth Scenarios

Defa Zhang

1. *Computer Department, Taizhou Vocational & Technical College, Taizhou 233 100, China; ;*

Yiwei Wang

1. *Computer Department, Taizhou Vocational & Technical College, Taizhou 233 100, China; ;*

Wang Rui

2. *Computer College, Beijing Information Science & Technology University, Beijing 100192, China;;*

Follow this and additional works at: <https://dc-china-simulation.researchcommons.org/journal>



Part of the Artificial Intelligence and Robotics Commons, Computer Engineering Commons, Numerical Analysis and Scientific Computing Commons, Operations Research, Systems Engineering and Industrial Engineering Commons, and the Systems Science Commons

This Paper is brought to you for free and open access by Journal of System Simulation. It has been accepted for inclusion in Journal of System Simulation by an authorized editor of Journal of System Simulation.

Real-time Shadow Algorithm Improvements in Digital Earth Scenarios

Abstract

Abstract: Studying real-time shadow generating and rendering algorithms in digital earth virtual reality scenes, a new method was provided to improve the shadow rendering quality and efficiency of earth scenes, based on the characteristics of huge dataset, huge coordinates and frequent scheduling mechanisms. Comparing with the classics CSM shadow map algorithm, the method proposed focused on the optimization of shadow splitting ranges, *as well as a new pipeline separating the rendering steps of terrain tiles and objects, so to improve both the correctness and efficiency.* analyzes The problems caused by shadow rendering while running dynamic terrain tile scheduler simultaneously were analyzed, and a practicable solution was provided for it, as well as some future plans.

Keywords

digital earth, real-time shadows, dynamic scheduler, shadow map algorithm

Recommended Citation

Zhang Defa, Wang Yiwei, Wang Rui. Real-time Shadow Algorithm Improvements in Digital Earth Scenarios[J]. Journal of System Simulation, 2015, 27(6): 1274-1279.

数字地球虚拟场景中实时阴影生成算法

张德发¹, 王一唯¹, 王锐²

(1. 台州职业技术学院计算机工程系, 浙江台州 233100; 2. 北京信息科技大学计算机学院, 北京 100192)

摘要: 针对数字地球虚拟现实三维场景中的实时阴影生成与渲染方法进行了研究, 并根据数字地球场景本身数据量巨大、场景坐标范围巨大、以及动态调度频繁的特点, 提出了改进渲染质量与渲染效率的方案。与经典的CSM (cascaded shadow map) 阴影图算法相比, 提出的方法虽然同样基于CSM的场景分割理论, 但是对分割的范围进行了进一步的优化和界定, 并且提出了地形-地物分别渲染的流程, 以同时确保渲染的正确性和效率。还对阴影渲染和大场景动态调度策略之间的矛盾进行了分析, 提出了有效的解决方案, 并且对未来的工作进行了展望。

关键词: 数字地球; 实时阴影; 动态调度; 阴影图算法

中图分类号: TP301.6

文献标识码: A

文章编号: 1004-731X (2015) 06-1274-07

DOI:10.16182/j.cnki.joss.2015.06.018

Real-time Shadow Algorithm Improvements in Digital Earth Scenarios

Zhang Defa¹, Wang Yiwei¹, Wang Rui²

(1. Computer Department, Taizhou Vocational & Technical College, Taizhou 233100, China;

2. Computer College, Beijing Information Science & Technology University, Beijing 100192, China)

Abstract: Studying real-time shadow generating and rendering algorithms in digital earth virtual reality scenes, a new method was provided to improve the shadow rendering quality and efficiency of earth scenes, based on the characteristics of huge dataset, huge coordinates and frequent scheduling mechanisms. Comparing with the classics CSM shadow map algorithm, the method proposed focused on the optimization of shadow splitting ranges, as well as a new pipeline separating the rendering steps of terrain tiles and objects, so to improve both the correctness and efficiency. analyzes The problems caused by shadow rendering while running dynamic terrain tile scheduler simultaneously were analyzed, and a practicable solution was provided for it, as well as some future plans.

Keywords: digital earth; real-time shadows; dynamic scheduler; shadow map algorithm

引言

实时阴影的渲染是计算机图形学中一个非常经典而重要的命题。由于计算机显示硬件条件的限制, 人们不可能采用光线跟踪等离线渲染软件中常见的算法来实现逼真和准确的阴影渲染, 而不得不

采取一些模拟和近似的方法, 以确保系统的高实时性和交互性要求, 同时以相对准确和美观的方式来呈现场景中物体的阴影投射情况。

实时阴影渲染的最普遍算法包括阴影图和阴影锥方法。阴影图算法本身具有场景渲染压力可控、算法简洁、可移植性和扩展性都较强的特点, 但是因为它将连续的场景几何信息转化为离散深度纹理像素的形式, 因此大幅度降低了系统的数据采样频率, 场景阴影的细节信息难以有效地表达。为了解决这一问题, 先后衍生出了对同一场景分割并渲染多张阴影图的 CSM 方法(cascaded



收稿日期: 2015-01-28 修回日期: 2015-03-04;
作者简介: 张德发(1962-), 男, 安徽定远人, 副教授, 研究方向为虚拟现实、图形图像、人工智能等; 王一唯(1987-), 女, 浙江台州人, 硕士, 研究方向为图形图像设计、动画等; 王锐(1984-), 男, 山东益都人, 硕士, 研究方向为虚拟现实与人机交互。

<http://www.china-simulation.com>

• 1274 •

shadow map)和对光空间进行扭曲变形以增加深度纹理利用率的 PSM 方法(perspective shadow map)^[1], 以及对阴影叠加的结果进行后处理滤波或者提前滤波的各种方法(例如 PCF, VSM 等)^[2]。这其中, 又以 CSM 方法的应用最为广泛。

本文对最常见的基于视锥体分割的 CSM 方法(又称作 PSSM, parallel-split shadow map)进行了深入分析, 并提出了针对数字地球场景优化的阴影图渲染算法。该方法相比传统的 CSM 算法, 对于大尺度的三维场景的渲染稳定性更高, 渲染效果更好; 并且渲染压力并不会因为渲染次数的增加而线性上升。同时, 本文所述的方法还支持地理信息系统中常见的各种球心坐标系统, 而不只是局限于直角坐标系中的应用。

1 针对数字地球场景的 CSM 算法实现

1.1 阴影图算法与阴影走样

阴影图是 1 种基于纹理图像的采样算法, 它的原理如图 1 所示。

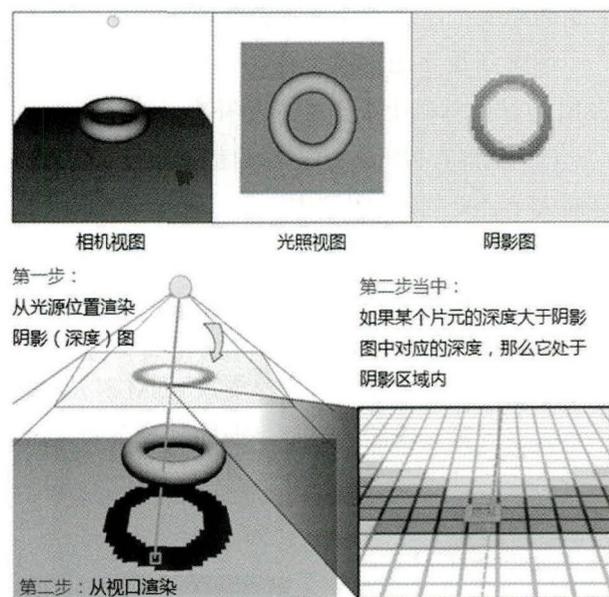


图 1 阴影图算法示意图

这一算法会产生 2 种不同类型的采样问题: 透视走样和投影走样^[3]。其中透视走样的问题对于大

场景而言尤为致命: 由于实时渲染时显卡的深度缓存并非线性计算的, 因此与观察点较远的物体, 其深度数据也更为粗糙, 计算数据的精确性会随着物体与观察点距离的增加呈指数下降——这与人们观察世界的实际感知特点也是相符的。但是, 阴影图算法中要求将场景的深度信息变换到光照空间中进行叠加和比较, 这就造成了远处物体阴影的走样问题。如果对于大场景, 没有设置合适的远近裁切平面的话, 就会导致图 2 所示的严重漏光或者剧烈的闪烁, 无法正常观察出阴影的投射情况。

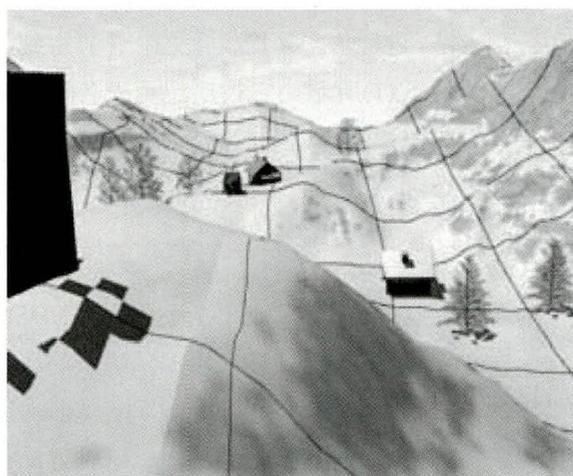


图 2 错误的阴影投射效果

1.2 CSM 阴影算法的原理和局限性

CSM 算法是改善透视走样问题的最直接方法。它的基本原理是将场景划分为多个小的子集, 然后在光空间内对每个子集进行分别渲染和叠加, 最后将结果合并到一起。而 PSSM 算法则是将当前观察范围内的场景按照由远及近的规则划分, 然后分别在不同的深度缓存内计算阴影和叠加, 从而降低了深度缓存精度对结果的影响^[4]。但是, 这种方法会造成我们对同一个场景进行更多次数的渲染, 因而大幅度增加了场景的渲染工作量; 如果划分的子集数目过少, 或者划分各个子集的范围不合理, 那么也不会收到很好的渲染效果, 走样问题依然明显可见, 如图 3 所示。

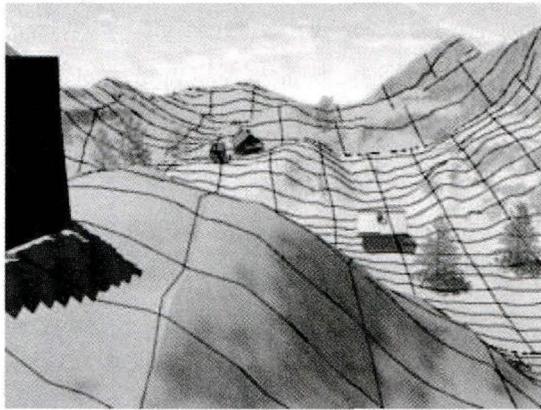


图 3 不同颜色区分的 3 级 CSM 阴影图

经典的视锥体划分方式结合了对数和简单的等距分割方法, 进行一个平均权重之后计算得到分割位置, 如公式(1)所示。它被广泛用于各种游戏的小规模场景阴影渲染, 但是无法适应场景深度范围较大的情形。

$$c_i^{pract} = aC_i^{\log} + (1-a)(z_n + \frac{i}{m(z_n - z_f)}) \quad (1)$$

为了解决这一矛盾, 人们继而提出了 SDSM 算法, 它在 CSM 方法的基础上根据预渲染的场景子集中的光贡献度, 自动调整划分区间, 从而自适应地得到评价分数最高的划分方式^[5]。这种方法被认为是目前最合理的对于 CSM 算法的改进, 但是在我们的实验中, 这种方法依然无法很好地应用到超大场景的阴影渲染当中, 其主要原因有 3 点:

(1) 超大场景的渲染压力本身已经较大, 而 CSM 阴影必须经过至少 2 次或者更多次数的(光空间内)渲染才能得到, 对系统的压力进一步增加^[6]。

(2) 超大场景的远近裁切平面值也是非常大的, 因此即使划分了多个子集分别渲染, 也依然难以弥补深度缓存精度带来的问题。

(3) 超大场景通常会采取一些数据动态调度的方法, 例如四叉树、八叉树等。磁盘上大量的数据瓦片会根据当前观察的位置和方向, 自动进行加载或者卸载。但是 CSM 算法过程中, 光空间的观察位置可能会不断发生变化, 因而导致了大量冗余的调度操作, 而频繁的磁盘 I/O 也会对系统的运行和稳定性产生直接的影响。

1.3 数字地球场景的精确裁切算法

我们限定使用平行光源来投射场景的阴影, 这是因为对于数字地球类的场景, 能够产生阴影的光源主要来自于太阳光, 而它是一种典型的平行光。因此, 光空间的视锥体应当采用正射投影矩阵的形式, 即场景在光空间的深度纹理图是一个囊括了整个场景的立方体, 其投影矩阵的计算公式为:

$$M_{lightProj} = \begin{bmatrix} \frac{2}{r-1} & 0 & 0 & 0 \\ 0 & \frac{2}{t-b} & 0 & 0 \\ 0 & 0 & \frac{2}{n-f} & 0 \\ 0 & 0 & \frac{n+f}{n-f} & 1 \end{bmatrix} \quad (2)$$

其中: n 和 f 分别为近裁切平面和远裁切平面; 1, r , b , t 分别表示立方体的 4 个面的坐标。

而对于 CSM 算法来说, 这个立方体会被切分为多个子集, 并且由于视锥体与光空间立方体之间几乎不会正好平行, 因此这些子集之间总是存在重叠关系。而子立方体的大小, 也就决定了深度缓存的精确程度, 以及最终阴影的走样程度。

考虑到数字地球场景的特性, 我们的算法在 CSM 方法的基础上提出了 3 条补充:

(1) 观察者视锥体远平面的裁切, 以不超过地球最大半径处的地平面为限。

由于地球最大半径处以及更远的地形瓦片的法线方向与观察者负方向的适量夹角将 $\geq 90^\circ$, 即这些地形瓦片都必定是不可见的。那么它们所接受的阴影也是不可见的, 可以提前从 CSM 算法的分割范围中剔除。新的远裁切平面的计算公式为:

$$f = \min(\sqrt{\|e-c\|^2 + r^2}, f_{orig}) \quad (3)$$

$$n = \max(k(1 - \frac{r}{\|e-c\|}) + k_0, 1.0) \quad (4)$$

这里 e 表示观察者所处的位置; c 表示地球球心的位置; r 表示近似得到的地球半径; f_{orig} 表示默认的远平面裁切值; k 和 k_0 表示近平面计算时所用的经验参数。

(2) 观察者视锥体近平面的裁切, 以不超过观察者所处位置的地形瓦片为限。

(3) 投射到地形的阴影, 与投射到其它地物的阴影计算分离处理。

其中第(1)条和第(2)补充可以极大地减小地球场景中远近裁切平面之间的距离, 因而提升了深度缓存的精度。但是, 对于其它地物上的阴影渲染, 例如在地形上飞驰的越野车, 或者在空中飞行的飞机等, 则存在不同的计算规则。尤其它们的自阴影处理, 不应当遵循地球坐标系下的裁切平面的限制。这种情况下, 我们需要对阴影渲染的流水线进行改进, 使之支持 2 种不同的阴影渲染参数设置, 并且可以分别执行和处理, 整个算法流程如下所示。

- 1) 计算当前视角的远近裁切面 P_n 和 P_f
- 2) 计算地球模型的 CSM 层次 N 和分割距离列表 S_n
- 3) *For* $n=0$ *To* $N-1$ {
渲染从 S_n 到 S_{n+1} 地球地形场景
记录场景的深度到深度纹理数组 T_n
}
- 4) 计算其它地物的 CSM 层次 N' 和分割距离列表 S'_n
- 5) *For* $n=0$ *To* $N'-1$ {
渲染从 S'_n 到 S'_{n+1} 地球地形场景
记录场景的深度到深度纹理数组 T'_n
}
- 6) 合并 T_n 与 T'_n 的各层纹理, 用于阴影计算

由此得到的结果保存到 2 个二维深度纹理数组当中, 这样可以避免产生过多的深度纹理, 并占用过多的纹理单元通道。每个纹理数组都要对应一个一致变量的结构体缓存对象, 其中包含的内容有: 实际视角的观察矩阵 M_{viewer} 、光空间的虚拟观察矩阵 $M_{lightView}$ 、光空间的虚拟投影矩阵 $M_{lightProj}$ 、阴影分割时设置的距离值和当前阴影子集在纹理数组中的索引。

最后在渲染观察者视角的视图时, 我们分别计

算每个深度纹理的阴影矩阵, 将当前观察者坐标系的顶点值与这个矩阵相乘, 就可以得到光空间坐标系下的值, 并且通过深度的比较得到最终的光照遮挡情况, 计算和显示阴影。如公式(5)所示:

$$V' = V \times M_{viewer} \times M_{lightView} \times M_{lightProj} \times \begin{bmatrix} 0.5 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0 \\ 0 & 0 & 0.5 & 0 \\ 0.5 & 0.5 & 0.5 & 1 \end{bmatrix} \quad (5)$$

其中: V 为观察者坐标系中的场景顶点, V' 为变换到光空间的深度纹理区间后的顶点坐标, 可以与深度纹理的内容直接进行比较。

2 数据动态调度对阴影算法的影响和修正

2.1 场景绘制过程中的动态调度机制

数字地球场景往往采取某种数据动态调度的方式, 根据当前观察者的位置和方向, 将磁盘中存储的大量数据瓦片动态加载到场景中显示, 将观察者视野范围之外的瓦片从内存中卸载, 从而确保场景的内存和显存占用量在一个相对稳定的数值内。

一个常见的数字地球场景的动态调度方案是四叉树方式, 将地球的 2 个半球视为 2 个四叉树的根节点, 然后分别一分为四, 生成第 2 级 LOD 的地形瓦片数据^[7]。以次类推, 当四叉树层级到达第 20 级时, 对应地形瓦片的空间分辨率已经可以达到 20 cm 左右。而这样的渲染规模显然不可能通过一次性加载所有 20 cm 精度的数据来实现, 如图 4 所示。



图 4 网格形式显示当前层级的地形瓦片数据图

2.2 光空间渲染的数据更新与卸载

因为 CSM 算法的主要流程是在光空间内渲染多张深度纹理图, 然后再从观察者视角渲染深度并与光空间的深度依次叠加, 所以势必在光空间内存在多次数据调度和渲染。这样的调度方式带来的另一个副作用是扰乱了正常的加载和卸载机制, 造成额外的数据瓦片被读入, 因此甚至可能存在数据超出内存限制导致崩溃的情形。

图 5 给出了使用 4 级 CSM 阴影算法对一个二叉树组织的数字地球场景进行动态的漫游和渲染, 并且无视冗余调度时的磁盘 I/O 情况。从图中可以看出, 传统 CSM 产生的磁盘调度过程中, 数据吞吐量表现很不稳定, 不利于调度算法优化和磁盘维护。

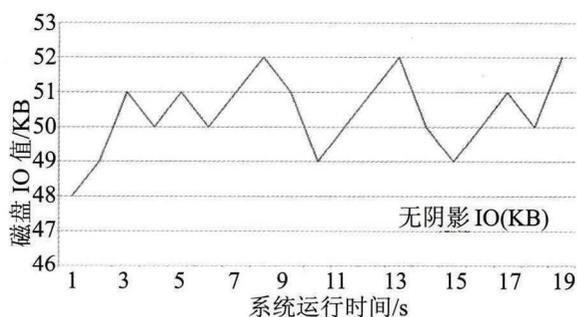


图 5 使用 CSM 阴影的磁盘调度情况图

另一种可行的方案是, 阴影渲染时完全不作数据的调度更新, 只有在观察者视角渲染时才更新一次数据。这样阴影渲染所对应的场景, 就是没有进行下一次调度之前的数据瓦片。这样所带来的一个问题是, 如果下一次数据调度中有明显的瓦片层次变化的话, 那么因为瓦片 LOD 粗糙程度的不同, 会造成阴影深度与场景深度的比较错误, 因而产生抖动或者阴影的漏缝、缺失情形, 影响渲染的质量。

因此, 本文提出了新的适合阴影渲染的数据调度方式, 即预先调度场景瓦片数据, 之后阴影的渲染和场景的实际渲染都使用同一次调度所得到的数据 LOD 结果。其执行流程如下所示

- 1) 开始当前帧的渲染
- 2) 根据当前视角调度地形瓦片数据, 得到加载列表 A_n 和卸载列表 R_n

- 3) IF A_n 不为空 { 加载 A_n 中的数据请求 }
- 4) IF R_n 不为空 { 卸载 R_n 中的现有数据 }
- 5) 计算地球模型的 CSM 层次 N 和分割距离列表 S_n
- 6) For $n=0$ To $N-1$ {
渲染从 S_n 到 S_{n+1} 地球地形场景
记录场景的深度到深度纹理数组 T_n
}
- 7) 合并各层纹理, 计算并渲染阴影
- 8) 返回步骤 1)

这一方案的优势是: 数据调度的工作量和磁盘读取速度相比默认无阴影时的调度方式不会有任何改变, 并且得到的结果不会产生异常。因为数据调度总是以观察者为中心完成的, 地形 LOD 的结果是近处精细而远处粗糙, 而由此得到的阴影同样也是相对观察者来说近处精细而远处粗糙, 这与实际的视觉观感效果相符^[8-9]。

3 实验结果与对比

我们基于开源的三维交互式图形渲染引擎 OpenSceneGraph^[10-11]构建了阴影算法开发和测试平台, 并且在这一平台上基于 GPU 着色语言与延迟渲染的方案^[12]实现了本文所述算法的测试和论证工作。表 1 所示为经典阴影图算法、CSM 以及我们改进的算法在渲染同一个基于数字地球的虚拟三维场景时每帧所消耗的平均时间(ms)。

表 1 每帧阴影渲染所需的时间对比

场景	CrytekSponza (顶点数 49 万)	数字地球与地形动态调度数据 (顶点数约 200 万)
单阴影图	6.4	14.4
CSM(3 级)	13.5	29.6
CSM(4 级)	17.8	35.5
本文算法(3 级)	13.7	19.3
本文算法(4 级)	17.0	23.8

从表中可以看出, 本文的方法在渲染较大数据的数字地球场景时, 所花费的时间明显少于传统的 CSM 方法。而单阴影图的方法虽然耗时更低, 但是在这种超大场景当中它的渲染结果是完全不可用的。场景渲染的部分结果如图 6 所示。

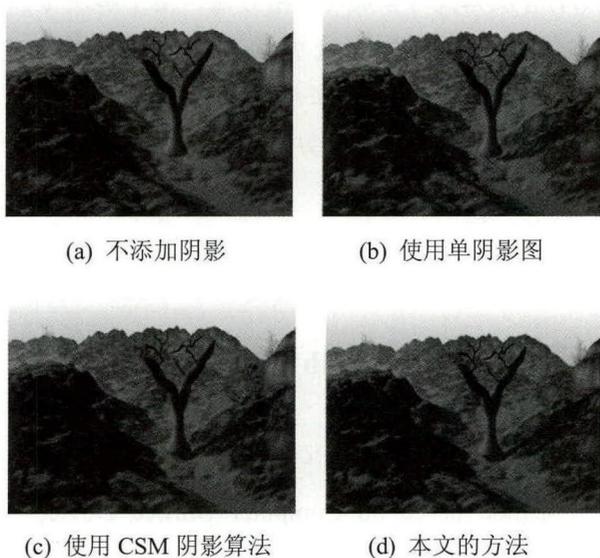


图 6 场景渲染的结果

从图 6 可以看出, 在渲染同一个场景时。我们的算法在处理这种大数据的细节表现上有比较明显的优势。场景渲染过程中的帧速率变化如图 7 所示。

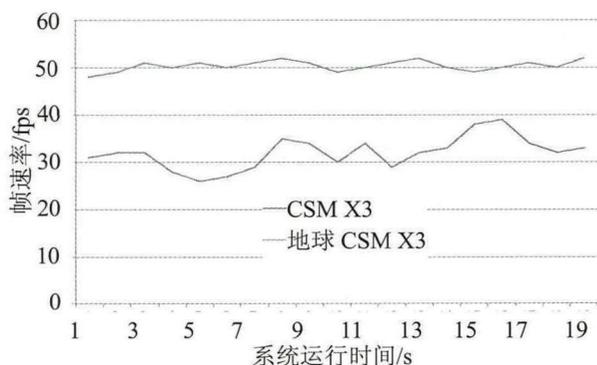


图 7 本文所述的地球 CSM 的算法帧速率与经典的 CSM 方法对比图

由图 7 可知, 本文算法的帧速率要好于经典的 CSM 方法, 具体表现在于帧速率更高, 且帧速率的变化比较平滑。

4 结论

本文介绍了一种针对数字地球级别的超大三维场景的阴影渲染和优化方案, 它在经典的 CSM 算法的基础上, 对远近裁切平面计算的问题进行了专门优化, 并且对于阴影渲染时的大数据调度方案进行了深入的探讨。相比经典的 CSM 算法, 本文所述的算法具有支持地球坐标系、更为合理地分配深度缓存以及分层支持地形和地物渲染这几个特点, 并且在实际应用过程中, 可以明显看出这一方法对于此类场景渲染的改进之处。

此外, 因为阴影图一类的算法具有很强的可扩展性, 所以可将本文所述的算法移植到已有的渲染引擎系统中, 或者为其添加延迟光源以及并行集群光源渲染等较新的渲染特性支持^[13-15], 或者将其它 PSM 类或滤波类的算法结合到本文算法中。这也正是我们在下一步的研究工作中需要考虑的方向。

参考文献:

- [1] Marc Stamminger, George Drettakis. Perspective Shadow Maps [J]. ACM Transactions on Graphics (S0730-0301), 2002, 21(3): 557-562.
- [2] 沈笠, 杨宝光, 冯结青, 等. 高质量快速百分比邻近滤波软影算法 [J]. 解放军理工大学学报(自然科学版), 2014, 26(3): 329-338.
- [3] Elmar Eisemann, Michael Schwarz, Ulf Assarsson, *et al.* Real-Time Shadows [M]. USA: A K Peters/CRC Press, 2011.
- [4] Fan Zhang, Hanqiu Sun, Lei Lei, *et al.* Parallel-Split Shadow Maps for Large-Scale Virtual Environments [C]// Proceedings of ACM International Conference on Virtual Reality Continuum and Its Applications. USA: ACM, 2006: 311-318.
- [5] Andrew Lauritzen, Marco Salvi, Aaron Lefohn, *et al.* Sample Distribution Shadow Maps [C]// ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games, 2011. USA: ACM, 2011.
- [6] Philip Dutre, Kavita Bala, Philippe Bekaert, *et al.* Advanced Global Illumination [M]. USA: A K Peters, 2006.
- [7] 郑新, 刘玮, 吕辰雷, 等. 海量地形实时动态存储与绘制的 GPU 实现算法 [J]. 计算机辅助设计与图形学学报, 2013, 25(8): 1146-1152.

(下转第 1287 页)

<http://www.china-simulation.com>

中选块、布块工作, 将映射计算中表面网格节点着色、三角面片内部点集遍历与着色等工作并行化, 同时转移到 GPU 上进行计算, 使体纹理合成的速度能够满足虚拟手术的交互性的需求, 且具有较强的真实感, 为后续切割显示的工作奠定了基础。开发了基于 OpenGL 的三维显示系统, 该系统能对肝脏体模型进行旋转、平移、缩放, 利用 CUDA 的并行特性, 我们对显示系统的性能进行了优化, 提高的系统的响应速度。但仍然存在一些问题, 如在三角面片数目过高时将导致显存不足以分配而使系统性能降低, 我们相信在具有高配置的医疗设备中进行新方法的合成时, 这一问题将迎刃而解; 另外, 由于尚未进行肝脏体模型切割计算的研究, 因此无法看到肝脏体模型内部的纹理细节, 在后续工作中, 我们将继续对研究肝脏体的切割操作及内部纹理显示的方法进行研究, 并在 CUDA 上进行计算, 使其能够在虚拟手术中实现交互速度的切割操作。

本文实现了肝脏体纹理的快速合成, 对于虚拟肝脏手术的研究有着重要的促进作用。新方法同样适用于纹理结构性较弱且周期性特征不明显的体纹理的合成, 因此在合成某些特定的体纹理时, 新方法也可以获得非常高的合成速度和真实感, 对体纹

理合成领域的发展有一定的指导意义。

参考文献:

- [1] 翟朝亮, 陈国栋, 王娜. 基于体纹理的肝脏可视化仿真方法研究 [J]. 电视技术, 2012, 36(17): 169-172.
- [2] 潘翔. 基于复用计算的肝脏软组织体纹理合成方法研究 [D]. 福州: 福州大学, 2014.
- [3] Johannes Kopf, Chi-Wing Fu, Daniel Cohen-Or, *et al.* Solid texture synthesis from 2D exemplars [J]. ACM Transactions on Graphics (S0730-0301), 2007, 26(3): 21-29.
- [4] Kenshi Takayama, Makoto Okabe, Takashi Ijiri, *et al.* Lapped Solid Textures: Filling a Model with Anisotropic Textures [J]. ACM Transactions on Graphics (S0730-0301), 2008, 27(3): 531-539.
- [5] Yue Dong, Sylvain Lefebvre, Xin Tong, *et al.* Lazy Solid Texture Synthesis [J]. Computer Graphics Forum, (S0167-7055), 2008, 27(4): 1165-1174.
- [6] 陈昕, 王文成. 大尺寸纹理的实时合成 [J]. 软件学报, 2009, 20(S1): 193-201.
- [7] 陈昕, 王文成. 基于复用计算的大纹理实时合成 [J]. 计算机学报, 2010, 33(4): 768-775.
- [8] 王一平, 王文成, 吴恩华. 块纹理合成的优化计算 [J]. 计算机辅助设计与图形学学报, 2006, 18(10): 1502-1507.
- [9] Perez P, Gangnet M, Blake A. Poisson image editing [J]. ACM Transaction on Graphics (S0730-0301), 2003, 22(3): 313-318.
- [10] 张建桥, 王长元. 基于泊松方程的数字图像无缝拼合 [J]. 现代电子技术, 2010 (17): 139-141.
- [11] Rui Wang, Xuelei Qian. OpenSceneGraph 3 Cookbook [M]. UK: Packt Publishing, 2012.
- [12] David Wolff. OpenGL 4 Shading Language Cookbook [M]. UK: Packt Publishing, 2013.
- [13] Sellers G, Obert J. Rendering massive virtual worlds [C]// ACM SIGGRAPH 2013 Courses. Anaheim, California, USA: ACM, 2013: 1-88.
- [14] Olsson O, Billeter M, *et al.* Clustered deferred and forward shading [C]// HPG '12: Proc. of the Conf. on High Performance Graphics, Paris, France, 2012. 2012: 87-96.
- [15] Karras T, Aila T. Fast parallel construction of high-quality bounding volume hierarchies [C]// HPG '13: Proc. of the Conf. on High Performance Graphics, Anaheim, California, USA, 2013. 2013: 89-99.

(上接第 1279 页)