

基于片段的光线投射算法

张怡 孙济洲 张加万

(天津大学计算机科学与技术学院, 天津 300072)

摘要 光线投射算法是最常用的体绘制算法之一,它能够产生高质量的结果图形,但是绘制的时间复杂度很高。提出了一种基于片段的光线投射算法(segment-based ray casting, SRC),以实现加速。同许多加速技术一样, SRC利用体数据的数据一致性,但是却将优化重点放在融合阶段而不是传统的数据预处理阶段。SRC将连续的具有相似属性的重采样点合并成一个片段,然后对片段进行融合而不是对重采样点进行融合,从而减少了融合操作的次数和时间。对SRC从理论和实验两个方面进行验证。实验结果表明,软件实现的光线投射算法使用SRC后性能提高约30%,而基于GPU的光线投射算法使用SRC后性能提升的倍数与片段长度几乎相同, SRC易于与其他体绘制优化算法结合,具有较强的适用性。

关键词 体绘制 光线投射 片段融合 图形加速卡

中图法分类号: TP391.72 **文献标识码:** A **文章编号:** 1006-8961(2007)08-1418-05

Ray Casting Algorithm Using Segment Composition

ZHANG Yi, SUN Ji-zhou, ZHANG Jia-wan

(School of Computer Science and Technology, Tianjin University, Tianjin 300072)

Abstract The traditional ray casting algorithm which is the most accurate volume rendering algorithm suffers from high computational complexity. In order to improve the traditional algorithm, a new ray casting accelerating algorithm, which is called SRC(segment-based ray casting) for short, is proposed in this paper. There are many optimization techniques making use of data coherence to reduce process time in former researches. SRC works the same way, while the optimization is emphasized in the composition stage instead of traditional data preprocessing stage. SRC combines sequential samples which have the similar optical values into a segment. The segments substitute for the samples to be the basic composition units. Therefore, the number of composition operations is reduced. The efficiency of SRC is demonstrated by both theoretical reasoning and experiments results. The experiments results show that software SRC can improve the rendering performance by about 30% compared with traditional ray casting without degrading image quality noticeably and hardware SRC can improve the performance by nearly the same multiple as the length of the segment. In addition, SRC is easy to combine with other successful optimizations.

Keywords volume rendering, ray casting, segment composition, GPU

1 引言

光线投射算法最先由Levoy^[1]于1988年提出,是最常用的体绘制算法之一。由于光线投射算法的绘制结果精确可靠,因此成为医学、工程计算等领

域进行数据可视化的首选算法。可是光线投射算法的计算复杂度大,绘制时间长,很难达到交互式应用的实时绘制要求,限制了它在实践中的普及、应用和发展,因此研究光线投射算法必须和相应的加速技术和策略相结合。

在体绘制算法的加速技术领域,国内外学者已

基金项目:国家自然科学基金项目(60373061)

收稿日期:2006-03-28; 改回日期:2006-05-19

第一作者简介:张怡(1981-),女,天津大学计算机应用技术专业博士研究生。现研究方向为体绘制算法、加速技术、压缩技术、信息可视化等。E-mail: ruochenlingxi@hotmail.com

经取得了诸多成果。现有的加速技术主要分为以下5类:无效体素剔除^[2]、提前不透明度截止、多分辨率绘制、硬件加速^[3-5]、并行加速。

本文提出了一种基于片段的光线投射算法(segment-based ray casting, SRC), SRC利用体数据存在的数据一致性对传统的光线投射算法进行简化从而提高绘制性能。体数据的数据一致性是指在体数据中,相邻体素通常具有相似或者相同的值,因此可以认为相邻的重采样点之间具有相似或者相同的光学属性。SRC将连续的相似重采样点合并成片段,并将融合的基本单位由重采样点扩大为片段,减少了融合次数,提高了绘制性能。实验结果表明,与使用重采样点融合的光投射算法相比,软件实现的SRC(以CPU计算为主)在保持绘制结果质量的前提下,可以提高绘制速度约30%;结合GPU加速技术, SRC可以将绘制速度提高数倍,对中小规模体数据(256³)可以实现高质量的实时绘制。

2 基于片段的绘制方程

2.1 基于重采样点的绘制方程

简要地讲,体绘制算法就是重采样点的光学属性按顺序进行融合的一个过程,重采样点之间的融合规则由绘制方程来规定,而绘制方程是从一定的光学模型推导出来的。吸收与放射模型是体绘制中最常用的模型,该模型认为重采样点是既可以放射光线,也可以吸收光线的粒子,是一种半透明的物质,它们的半透明度定义在区间[0, 1]。基于吸收与放射模型的绘制方程如下所示^[6]:

$$\tilde{C} = \sum_{i=0}^n \left(C_i \prod_{j=0}^{i-1} (1 - A_j) \right) \quad (1)$$

其中, C_i 是第 i 个重采样点经过传递函数映射后得到的光学属性,即 RGBA(红、绿、蓝、透明度)四元组, A_j 是第 j 个重采样点的不透明度, A_j 是 C_i 中的一个分量。依照式(1)进行的融合过程可以用迭代的方法进行计算,根据计算的顺序不同,可以分为从前向后融合和从后向前融合。从前向后融合的迭代公式为

$$\begin{aligned} C'_i &= C'_{i-1} + (1 - A'_{i-1})C_i \\ A'_i &= A'_{i-1} + (1 - A'_{i-1})A_i \end{aligned} \quad (2)$$

式(2)的初始条件为 $C'_0 = 0, A'_0 = 0$ 。

从后向前融合的迭代公式为

$$C'_i = C_i + (1 - A_i)C'_{i+1} \quad (3)$$

式(3)的初始条件为 $C'_n = 0$ 。

2.2 基于片段的绘制方程

体数据中能够体现其结构特征和成分组成的重采样点只是其中的一部分,甚至是一小部分,这部分重采样点是用户在绘制过程中所关心的,辨别这些重采样点的任务在分类阶段通过传递函数映射完成。由于重采样点是由体素插值得到的,因此可以先对体素进行筛选,只留下用户认为对绘制结构有意义的体素,重新构造一个更精简的体数据,并以此作为重采样点插值的基础。许多优化数据结构如八叉树、游程编码等都是将无效体素剔除掉以节省存储空间和绘制时间。但是分类是一项很困难的工作,尤其是对于没有前导经验的体数据,一般只能简单地将背景视为无效体素,因此在实际应用中,无效体素在体数据中所占的比例要比理论上小得多。在经过无效体素剔除后,有效体素一般不予处理,直接用于绘制,事实上有效体素中仍然存在着很高的一致性。另外无效体素的判别是依赖于传递函数的,每当传递函数改变的时候,就需要将体素重新划分,更新数据结构,这时候一部分有效体素和无效体素会发生角色转换。

基于对体数据的数据一致性分析,可以做如下假设:在一定范围内连续的重采样点通常具有某些特征上的相似,如具有相似的密度或者梯度,那么可以认为它们也具有相似的光学属性,更近似地认为,它们具有相同的光学属性。将连续的具有相似特征的重采样点组成一个片段,则一条光线被分为多个片段,经过推导,式(1)有如下形式:

$$C = \sum_{i=0}^M \left(c(t_i) \prod_{j=0}^{i-1} (1 - A_j) \right) \sum_{k=0}^{n_i-1} (1 - A_i)^k \quad (4)$$

其中, M 是一条光线中片段的个数, t_i 是第 i 个片段的最后一个重采样点; $c(t_i)$ 为 t_i 点的光学属性;令 $n_i = (t_i - t_{i-1})/\Delta t$, n_i 为第 i 个片段中包含的重采样点的个数,定义 n_i 为第 i 个片段的长度; A_i 表示第 i 个片段的不透明度。仍然可以用迭代的方法计算式(4),依照从前向后的顺序有:

$$C'_i = C'_{i-1} + (1 - A'_{i-1}) \sum_{k=0}^{n_i-1} (1 - A_i)^k C_i \quad (5)$$

$$A'_i = A'_{i-1} + (1 - A'_{i-1}) \sum_{k=0}^{n_i-1} (1 - A_i)^k A_i$$

式(5)的初始条件为 $C'_0 = 0, A'_0 = 0$ 。

从后向前融合的迭代公式为

$$C'_i = \sum_{k=0}^{n_i-1} (1 - A_i)^k C_i + (1 - A_i)^{n_i} C'_{i+1} \quad (6)$$

式(6)的初始条件是 $C'_n = 0$ 。且 $\sum_{k=0}^{n_i-1} (1 - A_i)^k$ 与 $(1 - A_i)^{n_i}$ 有如下关系:

$$(1 - A_i)^{n_i} = 1 - A_i \sum_{k=0}^{n_i-1} (1 - A_i)^k \quad (7)$$

式(5)和式(6)是 SRC 的绘制方程,比较式(2)和式(3):基于片段的绘制方程与原绘制方程具有相似的形式和计算过程,其区别在于光学属性 RGBA

各分量的前面均多一个因子 $\sum_{k=0}^{n_i-1} (1 - A_i)^k$ 。从物理意义上解释,由于光学模型仍是作用于重采样点,而融合的单位从重采样点变为光线片段,因此在每次融合须加权片段中所有重采样点的光学属性, $\sum_{k=0}^{n_i-1} (1 - A_i)^k$ 就是产生加权作用的因子。 A_i 是第 i 个片段的不透明度即片段内重采样点的不透明度,是已知量, n_i 是第 i 个片段的长度,可以由用户提前定义,确定了 A_i 和 n_i ,由式(7),可以提前计算出 $\sum_{k=0}^{n_i-1} (1 - A_i)^k$ 。

定义

$$SC_i = \sum_{k=0}^{n_i-1} (1 - A_i)^k \cdot C_i \quad (8)$$

$$SA_i = \sum_{k=0}^{n_i-1} (1 - A_i)^k \cdot A_i$$

则式(5)可以改写为

$$SC'_i = SC'_{i-1} + (1 - SA'_{i-1}) \cdot SC_i \quad (9)$$

$$SA'_i = SA'_{i-1} + (1 - SA'_{i-1}) \cdot SA_i$$

式(9)的初始条件是 $SC'_0 = 0, SA'_0 = 0$ 。

式(6)可以改写为

$$SC'_i = SC_i + (1 - SA_i) SC'_{i+1} \quad (10)$$

式(10)的初始条件是 $SC'_n = 0$ 。

对比基于重采样点的绘制方程式(2)、式(3)和相应的基于片段的绘制方程式(9)、式(10),可以发现,两者具有相同的形式。

3 SRC 的参数

将基于片段的的思想应用于光线投射算法,提出基于片段的光线投射算法(SRC)。SRC 是在简化条件下实现的一个优化算法,它认为体数据的数据一

致性总是存在的,并且密度值相近的体数据总是具有相同的光学属性,虽然这种假设在大部分情况下都是成立的,但是在某些特殊的情形下会丢失一些细节,例如传递函数在某一点处发生跳跃有可能就会被忽略,因此 SRC 在一定程度上以牺牲图像质量来提高绘制性能。

SRC 中有两个重要参数,一个是判定片段相似度的阈值 S (Similarity),另一个是片段长度的最大值 M (maximum length),这两个参数是决定图像质量和绘制性能的关键。

片段相似度阈值 S 是判定连续重采样点是否属于同一个片段的依据,以每个片段开始的第一个重采样点的值 V 为标准,该片段内所有的重采样点的值都在区间 $[V - S, V + S]$ 内,如果不限制片段的最大长度,则 S 是决定片段长度的唯一因素,这个参数和体数据的一致性相关的。

片段长度的最大值决定了一个片段可能存在的最大长度,设置片段的最大长度可以适度制约不合理的片段相似度造成的图像走样。

在 GPU 得到广泛应用之后,纯软件的体绘制算法在绘制性能方面已经明显处于劣势,但是使用软件方法对算法进行实验仍然具有其独特的优势。与基于 GPU 的算法相比,软件算法在数据结构和算法流程方面控制非常灵活,可以如实准确地反映算法设计者的思想,因此本文首先使用软件方法实现 SRC,得到上述两个绘制参数的推荐值,并将其用于基于 GPU 的 SRC 绘制过程。

图 1 是密度片段相似度(S)、梯度片段相似度(G)和片段最大长度(M)这 3 个参数的绘制性能曲线图。对 3 个不同的体数据(3 个数据的分辨率在表 1 中给出)进行绘制,绘制曲线表现出相似的趋势。

表 1 基于重采样点的光线投射算法 (TRC) 和 SRC 的绘制性能比较 ($S = 3, M = 3$)

Tab.1 Performance comparison of traditional resample-based ray casting (TRC) and segment-based ray casting (SRC) with rendering parameters $S = 3, M = 3$

数据	分辨率	TRC (s)	SRC (s)	加速比
Bonsai	256 × 256 × 128	18.11	10.922	1.658
Engine	256 × 256 × 110	15.078	8.485	1.777
Head	256 × 256 × 256	20.625	14.609	1.412

图 1(a)表明随着 S 的增大, SRC 的绘制速度不断提高,图中的 3 条曲线分别对应 3 个数据的绘制

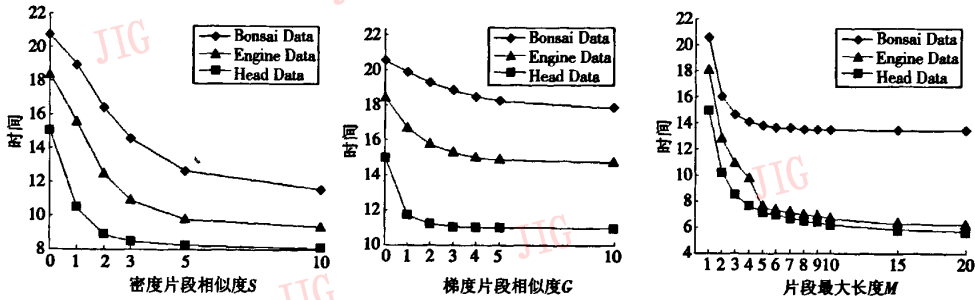


图 1 SRC 绘制曲线图

Fig.1 Performance curves of SRC

速度曲线,这里是以重采样点的密度作为唯一的片段相似度判断标准。绘制结果表明,对于 3 个类型的数据,当 $S > 5$ 时,绘制性能的提升幅度很小,当 $S > 3$ 的时候,图像质量的下降相对明显,因此选择 $S = 3$ 是比较合适的。

图 1(b)是受梯度片段相似度影响的绘制性能曲线,这组实验中在确定 $S = 3$ 的基础上,加入梯度 (G)作为片段相似度判定的标准。绘制结果表明,加入梯度虽然对图像质量有一定的提高,但是影响不是很大,因此可以选择只用密度作为相似度判定标准。如果使用梯度作为联合相似度判定标准,选择 $G = 2$ 和 $S = 3$ 是比较合适的。

图 1(c)是当 $S = 3$ 时,随着片段最大长度的增大,3 个数据的绘制性能曲线。曲线的拐点基本在区间 $[3, 5]$ 之间,由此可知,线段长度一般不会超过 5,只有少量的背景片段长度会比较大,选择 $M = 5$ 比较合适,虽然 M 对绘制结果没有决定性的作用,但是随着 M 的增大,绘制结果质量会逐步降低。在后面的实验中,为了保证绘制图像的质量,选择 $M = 3$ 作为绘制参数。

表 1 给出 3 个数据使用 SRC 的绘制加速比,相对于基于重采样点的光线投射算法,绘制性能提高幅度在 30% ~ 40% 之间。绘制采用的参数为 $S = 3$, $M = 3$ 。

4 基于 GPU 的 SRC 实现和结果

GPU 作为一种体绘制加速技术具有强大的性能优势,因此将新的体绘制加速技术放在 GPU 上进行讨论和验证,是符合体绘制技术的发展趋势和实际需求的。

GPU 流水线的不可逆性限制了程序设计师在

GPU 上精确实现 SRC,而只能实现 SRC 的简化方案。GPU 每轮绘制都要进行一次片段融合,但无法通过多个重采样点之间的比较确定片段长度,所以在实现时采用固定的片段长度,在绘制过程中每个片段只采样一次。片段长度的确定可以借鉴上一节中确定的最大片段长度,即处于区间 $[3, 5]$,后面的实验选用 3 作为固定的片段长度。

下面就基于 GPU 的 SRC 在不同采样率下的绘制结果质量和绘制性能进行总结和分析。以下所有的实验中, SRC 的固定片段长度为 3,使用的传递函数是线性传递函数,重采样间隔和绘制时间在每个绘制结果处标明。

(1) 重采样率低于体数据分辨率情形下的绘制结果

当绘制使用的重采样率低于体数据的分辨率时,使用传统的光线投射算法 (TRC) 产生的绘制结果不饱满,更严重的会缺失部分结构,如图 2 中的 (b)、(e),而使用 SRC 在同样的重采样率下,绘制结果很大程度上有所改善,图像比较清晰、饱满,结构也更加丰富,如图 2 中的 (a)、(f),因此在相同的传递函数设置规则下, SRC 使用较低的重采样率可以得到较好的绘制结果。

(2) 重采样率等于或者高于体数据分辨率情形下的绘制结果

当绘制使用的重采样率高于体数据的分辨率时,使用 TRC 可以得到质量很高的结果图像,如图 2(c),提高重采样率会造成绘制性能下降,有的时候过高的重采样率也会使结果图像变得模糊或者某些结构过于尖锐突出,如图 2(h)。使用 SRC 可以在保持一定结果图像质量的基础上,提高绘制性能,同时还降低过高的重采样率对结果图像产生的不良影响,如图 2 中的 (d)、(g)。

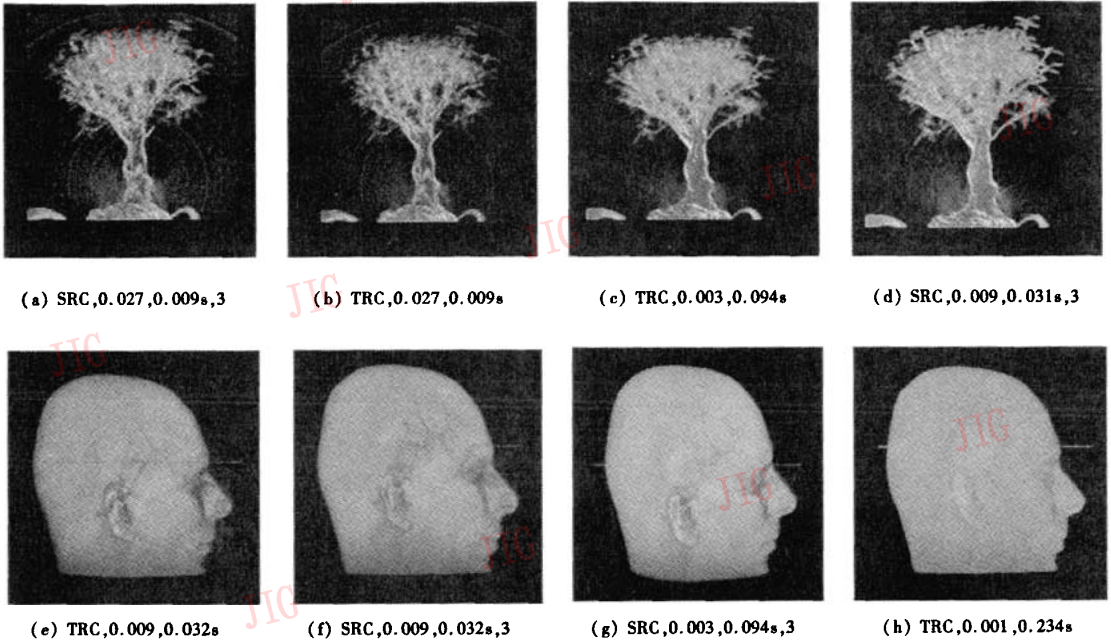


图 2 基于 GPU 的 SRC 绘制结果(算法(SRC/TRC),重采样间隔,绘制时间(s),片段长度)

Fig.2 Results of SRC on GPU

(3) 绘制性能

软件的 SRC 由于没有简化重采样操作,对绘制性能的提升十分有限,而基于 GPU 的 SRC 不仅简化了融合过程也减少了重采样的次数,其对绘制性能的提升是和重采样率降低的速度成正比的,重采样率减少的越多,绘制速度提高的越多。另外,基于 GPU 的 SRC 绘制速度与体数据规模无关,只与绘制窗口大小和重采样率相关。图 2 中各绘制结果都标注了绘制时间,即使重采样间隔是 0.003,相当于 333^3 分辨率的数据,只使用 GPU 和 SRC 两种加速技术,绘制速度也可以达到 10fps 以上。

5 结 论

从分析体数据的数据一致性出发,指出了在采用无效体素剔除优化技术基础上,体数据仍存在的数据一致性潜力,提出基于片段的光线投射算法(SRC)的简化条件,并依据简化条件推导基于片段的绘制方程。实验结果表明, SRC 可以在保持绘制图像质量的基础上,提高绘制速度,与 GPU 加速技

术结合, SRC 可以实现中小规模体数据的高质量实时绘制。

参考文献 (References)

- 1 Levoy M. Display of surfaces from volume data[J]. IEEE Computer Graphics & Application, 1988, 8(3): 29 ~ 37.
- 2 Crawfis R A, Max N. Texture splats for 3D scalar and vector field visualization [A]. In: Proceedings of IEEE Conference on Visualization [C], San Jose, CA, USA, 1993: 261 ~ 266.
- 3 Kruger J, Westermann R. Acceleration techniques for GPU-based volume rendering [A]. In: Proceedings of IEEE Conference on Visualization [C], Seattle, Washington, USA, 2003: 287 ~ 292.
- 4 Westover L A. Footprint evaluation for volume rendering [J]. Computer Graphic, 1990, 24(4): 367 ~ 376.
- 5 Wilson O, van Gelder A, Wilhelms J. Direct Volume Rendering via 3D Textures [R]. UCSC-CRL-94-19, Santa Cruz, California: Jack Baskin School of Engineering, University of California, 1994.
- 6 Klaus Engel, Markus Hadwiger, Joe Kniss, et al. Real-time volume graphics [A]. In: Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH [C], Los Angeles, California, 2004, Siggraph 2004 Course Notes#28.