

启发式探查最佳分割平面的快速 KD-Tree 构建方法

范文山^{1),2),3),4)} 王 斌^{2),3),4)}

¹⁾(清华大学计算机科学与技术系 北京 100084)

²⁾(清华大学软件学院 北京 100084)

³⁾(信息系统安全教育国家重点实验室 北京 100084)

⁴⁾(清华信息科学与技术国家实验室 北京 100084)

摘 要 在基于光线跟踪方法的真实感绘制中,kd-tree 是一种重要的加速结构. 文章对 kd-tree 的构建方法进行了研究,提出了一种基于分区(binning)算法的快速构建方法. 首先,通过分析 kd-tree 的成本函数,启发式地定位了当前节点的分割平面所在的子区间;其次,对探查到的子区间进行进一步的细化采样(sub-sampling),使得到的分割平面更好地逼近最优分割位置;同时,文章分析了现有方法在处理分割终止时存在的问题,提出了更加合理的分割终止条件. 与以往方法相比,新方法用更小的计算成本生成了质量更好的 kd-tree,构建过程更加鲁棒. 实验数据验证了文中方法的有效性.

关键词 光线跟踪;kd-tree;SAH;分区算法;细化采样

中图法分类号 TP391 **DOI 号**: 10.3724/SP.J.1016.2009.00185

A Fast KD-Tree Construction Method by Probing the Optimal Splitting Plane Heuristically

FAN Wen-Shan^{1),2),3),4)} WANG Bin^{2),3),4)}

¹⁾(Department of Computer Science and Technology, Tsinghua University, Beijing 100084)

²⁾(School of Software, Tsinghua University, Beijing 100084)

³⁾(Key Laboratory for Information System Security, Ministry of Education of China, Beijing 100084)

⁴⁾(Tsinghua National Laboratory for Information Science and Technology, Beijing 100084)

Abstract In the field of ray-tracing based photo-realistic rendering, kd-tree is used as an important acceleration structure. This paper focuses on the effective construction of kd-tree, and proposes a novel and fast construction method which is based on the binning algorithm. The method is composed of two main steps. Firstly, by analyzing the SAH cost function, the method determines the most-likely sub-span which holds the splitting plane. Secondly, sub-sampling is used on the resulted span to get much better approximation to the optimal splitting plane. Moreover, the paper discusses the exiting schemes on binning termination condition, points out their problems, and proposes a more reasonable termination condition. The experimental results show that the novel approach is effective. Compared with the previous works, it decreases the construction overhead, improves the quality of generated kd-tree, and the construction procedure is more robust as well.

Keywords ray tracing; kd-tree; SAH; binning algorithm; sub-sampling

收稿日期:2008-08-09;最终修改稿收到日期:2008-12-12. 本课题得到国家自然科学基金(90715043,90818011,60773143,60533070)、国家“九七三”重点基础研究发展规划项目基金(2004CB719400)、国家“八六三”高技术研究发展计划项目基金(2007AA040401)、高等学校全国优秀博士学位论文作者专项资金(200342)和霍英东教育基金会(111070)资助. 范文山,男,1975年生,博士研究生,主要研究方向为实时渲染技术、真实感渲染、曲面离散技术. E-mail: fws06@mails.tsinghua.edu.cn. 王 斌,男,1976年生,博士,讲师,主要研究方向为计算机图形学、计算机辅助设计.

1 引 言

在高质量真实感图像渲染领域,光线跟踪技术近年来得到了长足的发展,这一方面得益于硬件性能的提高,另一方面是由于高质量的加速数据结构的提出及改进,以及基于这些加速结构的遍历算法^[1-3]的应用.这些进步使得目前在普通微机上就可以实现复杂场景的交互式的光线跟踪应用.

作为一种全局光照技术,光线跟踪方法需要访问全部的场景空间,因此选择并构造高效的空间表示方法是加速光线跟踪应用的重要因素.常用的空间表示方法有均匀网格、Oct-tree、kd-tree、BVH等^[4].这些方法各具特点,适用不同的场景,因此不能用一种标准定义各种数据结构的性能优劣.但综合近来研究及应用情况,kd-tree被越来越多地研究人员青睐^[5],它的空间适应性好,被形象地描述为可以解决诸如“Teapot in the stadium”之类的空间表示问题^[6].

在光线跟踪中使用 kd-tree 主要涉及 kd-tree 的构建与 kd-tree 的遍历两个问题,本文主要讨论前一个问题.Kd-tree 的构建过程是一个计算强度大而耗时的过程.良好的 kd-tree 构建方法能够生成优化的树结构,同时能够降低构建过程自身的复杂度,提高构建速度.

kd-tree 的构建过程是对节点空间进行自上而下划分的递归过程.伪码描述如下:

```
Procedure BuildKDTree(Primitives  $T$ , Voxel  $V$ )
    //输入图元对象集合  $T$  及
    //场景包围盒  $V$ 
Begin
    if Terminate( $T$ ,  $V$ )           //判断是否满足终止条件
        return LeafNode( $T$ , $V$ );
    else
         $P=FindSplitPlane(T,V)$ ; //选择分割平面
         $(V_l, V_r)=Split\ V\ with\ P$ ; //空间分割
         $T_l=\{t\in T\ and\ t\cap V_l\neq\emptyset\}$ ; //将图元对象划分到分割
            //平面左侧
         $T_r=\{t\in T\ and\ t\cap V_r\neq\emptyset\}$ ; //将图元对象划分到分割
            //平面右侧
        return Node( $P$ ,BuildKDTree( $T_l,V_l$ ),
            BuildKDTree( $T_r,V_r$ )); //递归分割
End
```

选择分割平面是构建 kd-tree 的核心工作.有多种选择分割平面的标准,表面积启发(Surface Area

Heuristic,SAH)^[7-8]是当前普遍使用的选择分割平面的处理方法.该方法基于几何概率理论^[9],考虑了影响光线遍历过程的两个因素:空间尺寸和图元对象分布.实际应用表明,SAH 方法是一种合理、高效的 kd-tree 构建方法.

本文所做的工作对基于分区的 kd-tree 构建算法进行了改进,提出了一种新的探查最优分割平面的方法.主要思想是通过分析成本函数,以节点的中点位置为参考点,启发式地定位最佳分割平面所在的子区间,避免对全部区间的遍历.然后,对探查到的子区间进一步地细化采样,得到最佳分割平面的更好的逼近;文章对终止条件的选择进行了讨论,分析了现有方法存在的问题,提出了更合理的分区终止条件.实验数据表明,本文提出的方法提高了 kd-tree 的构建速度,改善了 kd-tree 的质量,提高了系统的鲁棒性.同时本算法具有良好的并行扩展性.

2 相关工作

提高 kd-tree 的质量,以加速光线遍历空间的过程是 kd-tree 构建方法的重点研究内容之一.文献[10]对 kd-tree 构建过程的终止标准和候选分割平面的选取方法进行了研究,提出了一种自动终止标准(Automatic Termination Criteria,ATC),该文还使用了分割剪裁(split clipping)方法,对图元对象进行分割,得到更优的候选分割平面,使 kd-tree 的质量得以优化(10%~35%的提高).但分割剪裁方法需要平面与多边形的求交计算,降低了构建速度.

SAH 算法构建 kd-tree 的复杂度理论下限为 $O(n\log n)$,但大多数实际使用的算法复杂度为 $O(n\log^2 n)$,文献[5]对 3 种复杂度的构建方法进行了详细说明,并重点描述了一种 $O(n\log n)$ 复杂度的 kd-tree 构建方法,使 kd-tree 的构建速度得到了提高.这些算法都依赖于对场景中全部对象的边界位置进行排序,通常也被称为传统的 kd-tree 构建方法.

为满足快速、交互式的应用需要,近年来,kd-tree 的构建方法研究重点聚焦到了不需要排序的基于分区(binning)的构建方法上.这类方法不是使用图元对象的代理 AABB(Axis Aligned Bounding Box)的边界作为候选分割平面,而是对场景空间进行划分,然后在这些划分的边界位置处探查分割平面,得到 kd-tree 最小成本的近似.为提高速度,通常分割的区域数目远小于场景中的对象数.

传统的基于分区的构建方法存在两个主要问题. 第一是对分割平面两侧的图元对象的数目统计不准确; 另一个问题是探查的候选分割位置数目少, 得到的近似成本与最优的成本误差大.

文献[11]提出了最小-最大分区(min-max binning)算法, 改善了第一个问题. 最小-最大分区算法使用 min-bin 和 max-bin 两个数据结构分别计数映射到各个区间的图元代理 AABB 的起点数和终点数, 然后根据 min-bin 计算分割平面左侧的图元对象数, 根据 max-bin 计算分割平面右侧的图元对象数, 但文献[11]实现的是一种均匀的分区方法, 无法对第二个问题给出有效的解决方案.

简单地增加探查位置数目也不能很好地解决第二个问题. 为此, 一些自适应的方法被提出来. 文献[12]用给定误差限的分段二次函数来近似 SAH 成本函数, 进行非均匀的自适应采样. 本文通过对 SAH 成本函数进行分析, 得到启发信息, 对基于分区的方法的第二个问题给出了一种新的解决方案, 提出了非均匀、自适应地探查近似最优分割位置的方法.

基于分区的 kd-tree 构建算法是一个两阶段的过程, 在生成 kd-tree 的高层阶段应用分区方法, 当构建到一定的深度, 要终止第一阶段的算法, 转到传统的 kd-tree 构建方法完成树的构建. 文献[12]简单地提及了选择终止条件的方法. 终止条件的选择是影响构建效率的重要因素. 本文对终止条件的设定进行了分析, 改进了终止条件设定的合理性.

为了提高构建 kd-tree 的速度, 研究人员对构建方法的并行化进行了研究. 文献[11]使用多线程机制实现在多内核 CPU 平台上的 kd-tree 的并行构建. 文献[12-13]通过 SIMD 流指令实现了基于分区的并行构建 kd-tree 方法. 他们的实验数据表明, 并行化可以显著地提高 kd-tree 的构建速度. 作为一种分区方法, 本文提出的算法具有同样的并行扩展性.

3 改进的分区算法

本文的算法从分析 SAH 的成本函数入手, 以中点为出发点. 中点位置对 SAH 是一个特别的位置, SAH 方法理论下限的推导就是假设每次分割都发生在中点处^[5].

3.1 以中点为参考点的 SAH 成本函数

SAH 方法定义的 kd-tree 的成本函数为

$$C(V) = K_t + P_l C(V_l) + P_r C(V_r) \quad (1)$$

其中, V, V_l, V_r 为当前节点及左、右孩子节点的 AABB, K_t 为遍历 kd-tree 一步的成本, P_l, P_r 为光线进入左、右子树的概率, 用节点 AABB 的面积比表示: $P_l = SA(V_l)/SA(V)$, $P_r = SA(V_r)/SA(V)$. $C(V_l), C(V_r)$ 为左、右子树的成本. 展开式(1)得到如下形式的成本函数

$$C(V) = \sum_{n \in \text{nodes}} \frac{SA(V_n)}{SA(V_s)} K_t + \sum_{l \in \text{leaves}} \frac{SA(V_l)}{SA(V_s)} K_i \quad (2)$$

其中, V_s 为整个场景的 AABB, V_n 为中间节点的 AABB, V_l 为叶节点的 AABB, K_i 为求交成本.

根据 SAH 原理, 最优的 kd-tree 是具有最小成本的树. 然而要得到 kd-tree 中所有层次的最优划分位置的精确解, 使式(2)的成本最小, 对一个现实的场景计算代价太高. 因此实际应用中使用局部贪心(local greedy)算法近似求解. 局部贪心算法在计算成本时, 将子结点近似看作叶子节点, 这样可以很容易地以自上而下的方式近似估算出当前划分的成本. 基于局部贪心算法的成本公式表示为

$$C(V) \approx K_t + K_i (P_l N_l + P_r N_r) \quad (3)$$

N_l, N_r 为左、右孩子节点中的对象数. 由于 K_t 和 K_i 为常数, 因此本文对成本函数的分析只关注 $(P_l N_l + P_r N_r)$ 部分.

如图 1 所示, P_l, P_r 可分别表示为

$$P_l = SA(V_l)/SA(V)$$

$$= 2(ab\omega + ac\omega + bc)/2(ab + ac + bc)$$

$$P_r = SA(V_r)/SA(V)$$

$$= 2(ab(1-\omega) + ac(1-\omega) + bc)/2(ab + ac + bc) \quad (4)$$

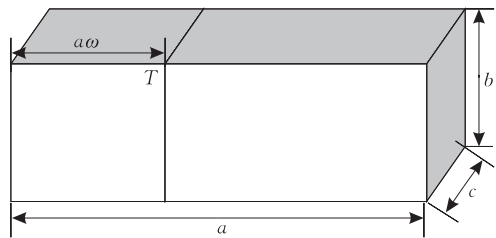


图 1 节点被分割平面 T 分割为左右两个子节点

ω 为分割平面 T 到左端点的长度与整条边长的比例. 如图 2 所示, 我们考虑中间点处的分割平面 T_m ($\omega=1/2$), 定义有符号变量 Δ , Δ 的绝对值表示 T 到 T_m 的长度与整条边长的比例, 并令 T 在 T_m 的左侧时, $\Delta > 0$; T 在 T_m 右侧时, $\Delta < 0$, 则 T 位置处 $\omega = 1/2 - \Delta$.

根据式(4)可以得到

$$P_l = 2(ab(1/2 - \Delta) + ac(1/2 - \Delta) + bc)/$$

$$\begin{aligned}
& 2(ab+ac+bc) \\
& = (ab+ac+2bc)/2(ab+ac+bc) - \\
& \quad 2(ab\Delta+ac\Delta)/2(ab+ac+bc), \\
P_r & = 2(ab(1/2+\Delta)+ac(1/2+\Delta)+bc)/ \\
& \quad 2(ab+ac+bc) \\
& = (ab+ac+2bc)/2(ab+ac+bc) + \\
& \quad 2(ab\Delta+ac\Delta)/2(ab+ac+bc) \quad (5)
\end{aligned}$$

用 P_m 表示 $(ab+ac+2bc)/2(ab+ac+bc)$, P_Δ 表示 $2(ab\Delta+ac\Delta)/2(ab+ac+bc)$, 简化式(5)的表达形式, 则

$$\begin{aligned}
P_l & = P_m - P_\Delta, \\
P_r & = P_m + P_\Delta \quad (6)
\end{aligned}$$

显然, 当 T 在 T_m 左侧时, $P_\Delta > 0$, T 在 T_m 右侧时, $P_\Delta < 0$.

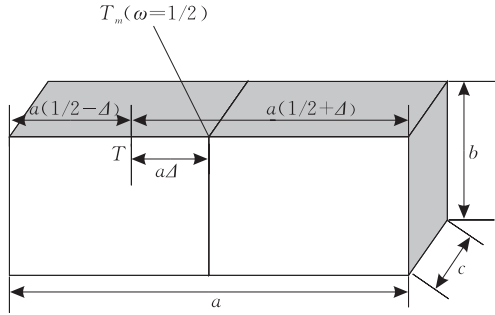


图 2 以中点位置为参考点表示分割平面 T

定义中点处分割平面 T_m 左侧节点中对象数目为 N_{ml} , 右侧对象数目为 N_{mr} , 分割平面从中点处移动到 T 处时, T 左侧节点的对象数减少了 $N_{l\Delta}$, 则左侧节点中对象数 $N_l = N_{ml} - N_{l\Delta}$; 右侧节点的对象数目增加了 $N_{r\Delta}$, 则右侧节点中对象数 $N_r = N_{mr} + N_{r\Delta}$. 显然当分割平面从中点向左移动到 T 时, $N_{l\Delta} > 0$, $N_{r\Delta} > 0$, 从中点向右侧移动到 T 时, $N_{l\Delta} < 0$, $N_{r\Delta} < 0$. 重写 $(P_l N_l + P_r N_r)$ 得到以中点为参考点的 SAH 成本公式

$$\begin{aligned}
& P_l N_l + P_r N_r \\
& = (P_m - P_\Delta)(N_{ml} - N_{l\Delta}) + (P_m + P_\Delta)(N_{mr} + N_{r\Delta}) \\
& = P_m N_{ml} + P_m N_{mr} + P_m (N_{r\Delta} - N_{l\Delta}) + \\
& \quad P_\Delta (N_{mr} - N_{ml}) + P_\Delta (N_{r\Delta} + N_{l\Delta}) \quad (7)
\end{aligned}$$

3.2 启发式探查最佳分割平面所在子区间

我们对式(7)进行分析, 得到启发线索. 由于 $P_m N_{ml} + P_m N_{mr}$ 项为常量, 因此只关注式(7)中 $P_m (N_{r\Delta} - N_{l\Delta}) + P_\Delta (N_{mr} - N_{ml}) + P_\Delta (N_{r\Delta} + N_{l\Delta})$ 部分.

先分析 $P_m (N_{r\Delta} - N_{l\Delta})$ 项. 如图 3 所示. 实线框中的图元集合表示从 T_m 到 T 时, 退出分割平面左侧的部分, 其数目即为 $N_{l\Delta}$; 虚线框中的图元集合表

示从 T_m 到 T 时, 增加到分割平面右侧的部分, 其数目即为 $N_{r\Delta}$. 由图示可知, 这两个集合分别包括两部分: 实线框中的集合由白色图元以及与 T_m 相交的竖线填充模式表示的图元组成; 虚线框中的集合由白色图元以及 T 相交的网格填充模式表示的图元组成. 二者的区别在于不同的分割平面相交的部分. 根据文献[14], 理论上, 分别与不同的分割平面相交的图元集合包含的图元数值之差为一小值. 因此 $N_{l\Delta} - N_{r\Delta}$ 为一小数值, 又由于 $0 < P_m < 1$, 因此我们可以忽略 $P_m (N_{r\Delta} - N_{l\Delta})$ 项对成本的影响.

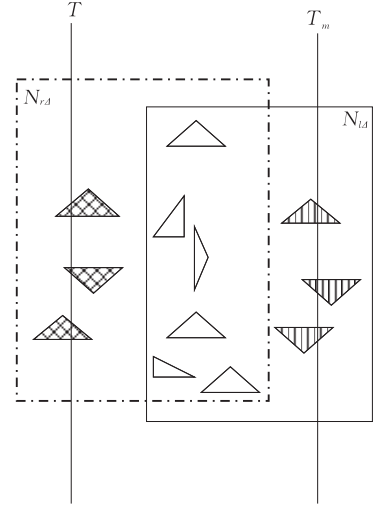


图 3 分割平面从 T_m 变化到 T 时, 左右子结点中图元集合的变化情况

对 $P_\Delta (N_{mr} - N_{ml}) + P_\Delta (N_{r\Delta} + N_{l\Delta})$ 部分, 我们根据 N_{ml} 和 N_{mr} , 分情况讨论.

(1) $N_{mr} > N_{ml}$. 如果在这种情况下, 向左探查分割平面, 由于此时变量 $P_\Delta, N_{l\Delta}, N_{r\Delta}$ 均大于 0, 分割平面越向左远离中点, $P_\Delta, N_{l\Delta}, N_{r\Delta}$ 值越大, 相应地, $P_\Delta (N_{mr} - N_{ml}) + P_\Delta (N_{r\Delta} + N_{l\Delta})$ 值越大, SAH 成本越高. 因此在 $N_{mr} > N_{ml}$ 情况下, 不应该到中点左侧探查最佳分割位置. 最佳分割位置最有可能在中点右侧取得. 在中点右侧探查时, $P_\Delta, N_{l\Delta}, N_{r\Delta}$ 均小于 0. 因此 $P_\Delta (N_{mr} - N_{ml})$ 小于 0 且单调线性递减, $P_\Delta (N_{r\Delta} + N_{l\Delta})$ 大于 0 且单调递增. 而且, 分割位置越向右远离中点, $N_{r\Delta} + N_{l\Delta}$ 绝对值越大, 在某个位置 $T_k (\omega = x)$ 处, $N_{r\Delta} + N_{l\Delta}$ 绝对值会大于或等于 $N_{mr} - N_{ml}$, 此时 $P_\Delta (N_{r\Delta} + N_{l\Delta}) + P_\Delta (N_{mr} - N_{ml}) \geq 0$, 之后 $P_\Delta (N_{mr} - N_{ml}) + P_\Delta (N_{mr} - N_{ml})$ 保持大于 0 并递增. $P_\Delta (N_{mr} - N_{ml}), P_\Delta (N_{r\Delta} + N_{l\Delta})$ 和 $P_\Delta (N_{r\Delta} + N_{l\Delta}) + P_\Delta (N_{mr} - N_{ml})$ 各项变化趋势如图 4 所示.

因此最佳分割位置一定在中点 T_m 与 T_k 之间, 即 ω 在子区间 $[1/2, x]$ 上.

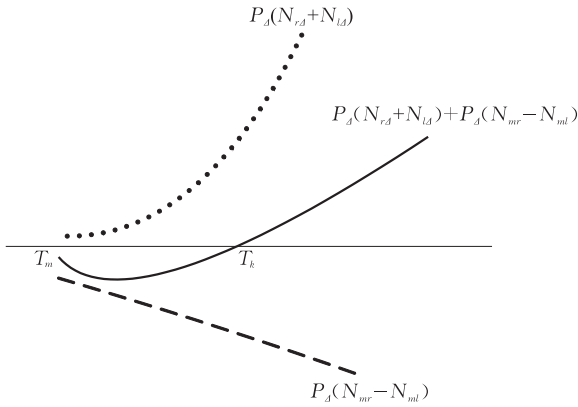


图 4 各成本项的变化趋势图

(2) $N_{mr} < N_{ml}$. 我们可以对称地处理 $N_{mr} < N_{ml}$ 情形. 此时, 应该向左侧移动分割平面, 探查的 ω 所在的子区间为 $[x, 1/2]$.

(3) $N_{mr} = N_{ml}$. 此时无论向左探查还是向右探查, $P_d(N_{r_d} + N_{l_d}) + P_d(N_{mr} - N_{ml})$ 均大于 0, 因此, 最佳分割位置就在中点. 这与推导 SAH 方法复杂度下限时所做的假设是吻合的.

3.3 细化采样得到近似的最佳分割平面

我们对中点位置处的全部 3 种情况进行了讨论, 即当 $N_{mr} > N_{ml}$ 时, 在 $[1/2, x]$ 上探查最佳分割平面的近似; 当 $N_{mr} < N_{ml}$ 时, 在 $[x, 1/2]$ 上探查最佳分割平面的近似; 当 $N_{mr} = N_{ml}$ 时, 中点位置处的分割平面就是最佳分割平面.

确定子区间边界 x 的方法是: 在初始的区域划分序列上, 从中间位置开始, 沿分析得到的探查方向, 对各划分区域边界进行成本函数的求值, 将满足 $P_d(N_{r_d} + N_{l_d}) + P_d(N_{mr} - N_{ml}) \geq 0$ 的第一个边界位置作为前面讨论的 T_k , 通过该值即可简单地求得 x .

在确定了子区间范围后, 接下来对该子区间进行细化采样, 建立新的分区方案. 最后对重分区后的子区间遍历各个分区边界, 查找最小成本函数对应的位置. 由于新建立的分区方案比初始划分的离散间隙更小, 因此查找得到的位置更加靠近最佳分割平面, 保证了对最佳分割平面更好的逼近.

与遍历查找方法相比, 自适应的方法需要面对选择的子区间内不包含实际的最佳分割平面的问题. 在前面的分析中, 我们从理论上忽略了 $P_m(N_{r_d} - N_{l_d})$ 项对成本函数的贡献, 但对某个具体的场景而言, 可能会由于这一假设导致上述问题的发生.

当出现这种情形时, 本文认为, 尽管由于 $P_m(N_{r_d} - N_{l_d})$ 项的影响使我们的探查“失效”, 但根

据图 4 的成本曲线图, 即使得到的子区间内不包括使 SAH 成本最小的最佳分割平面, 该子区间内仍包含有使 SAH 成本取得准最小值的准最佳分割平面. 由于分区算法本身就是在少数的、有限的候选位置集合中挑选一个具有最小成本函数值的位置来代替(近似)真正的最佳分割位置, 因此即使是逐一遍历查找的方法, 这样的“代替”也面对有发生大误差的可能. 此外, 本算法在子区间上的细化采样操作提高了近似的精度. 综合以上考虑, 即使发生“失效”, 对区域分割算法而言, 本方法选出的分割位置也是可接受的理想位置.

3.4 对终止条件的分析

基于分区的 kd-tree 构建过程在进行到一定程度时, 需要终止第一阶段的处理, 转到第二阶段, 使用传统的 SAH 算法完成深层次树的生成. 之前的研究工作通常以节点内对象数目作为判断依据. 文献[12]中当节点内的对象数低于 36 时, 终止第一阶段构建, 转换到传统的 SAH 算法. 文献[11]中当节点内对象数小于划分的区间数时, 终止第一阶段过程.

本文实验发现, 构建 kd-tree 的第一阶段, 在树的高层进行节点分割时, 完全分布在分割平面两侧的图元占大多数, 与分割平面相交的图元占少数. 当构建到树的深层次时, 由于节点空间的缩小, 出现图元对象充满节点空间的情形越来越多. 对这样的节点继续划分会造成大量冗余节点(包含基本相同对象集合的节点)的产生, 增加了 kd-tree 的深度和节点数, 损害 kd-tree 的质量, 同时也增加了构建时间.

更严重的是, 当发生图 5 所示的情形时, 对节点的划分将处于一种稳定状态. 图 5 以二维情形说明此问题. 图中的矩形框代表节点空间, 其中的所有图元在各维度上都充满了该节点. 此时, 划分当前节点将得到两个包含与父节点完全相同的图元集合的子节点, 如果此时集合中的图元对象数目不小于设定的终止数值, 划分会无休止地继续进行下去.

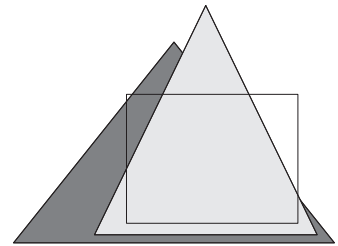


图 5 一个各维度被所有的图元充满的节点

之前只考虑节点内图元数目的终止条件不能避

免这种情形发生,划分过程中只要出现一个这样的节点,kd-tree 的构建过程就会失败. 如同 SAH 算法选择分割平面时考虑图元对象与节点空间两个因素一样,本文在设计终止条件时,除了使用节点数目外,也将节点的空间尺寸作为判断终止的条件. 当节点内对象数目小于阈值或者当前节点待分割维度的尺寸小于阈值时,终止第一阶段过程. 节点尺寸的阈值通过统计场景中全部图元在各维度上的尺寸信息得到.

本文的终止条件,既考虑到了将小数目的图元集合传递给第二阶段的要求,也使得第一阶段操作不会由于节点空间尺寸过小产生过多的冗余节点. 此终止条件更趋合理,降低了 kd-tree 构建过程发生失败的可能.

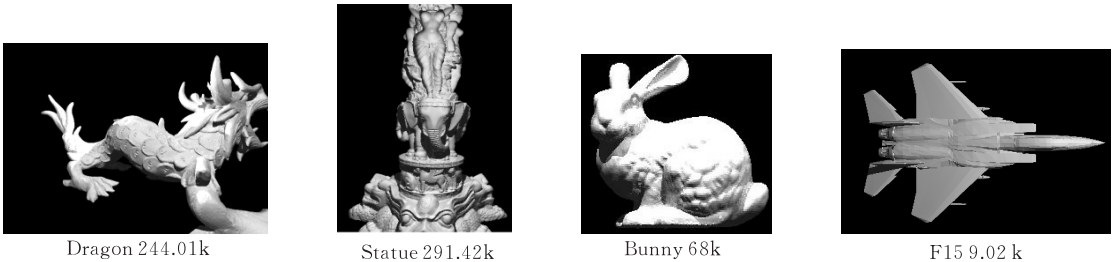


图 6 本文使用的模型及模型所含的三角面片数目

我们对比了改进的算法和最小最大分区算法构建 kd-tree 的时间. 为了测试本方法的空间适应性,我们组合多个模型来模拟空间分布不均匀的场景.

4.1 未应用改进的终止条件的构建时间比较

实验中,用本文方法进行的初始分区使用大的、

在本文的实验中,将尺寸信息纳入到终止条件的考量中来. 通过统计场景中图元在各维度上的尺寸信息来设定终止划分的尺寸阈值,在目前的工作中,还是作为一种提示性的、手工的方式. 未来的工作,设计自动的终止条件是我们所要考虑的.

4 结果和讨论

本文实现了单线程的改进算法,第二阶段使用了 $O(n \log^2 n)$ 复杂度算法. 使用的微机基本配置: PIV 2.8GHz 双核 CPU, 2GB 主存, 编译环境为 VS2005. 使用的模型及模型中所含的三角面元数如图 6 所示.

粗糙的分割粒度,细化分区的粒度与最小最大分区算法保持一致. 在终止条件上,二者使用相同的最小终止对象数,均不应用改进的终止条件. 实验数据如表 1 所示.

表 1 启发式探查并细化采样的构建方法与最小最大分区算法构建时间的比较

场景	终止划分的对象数目	本文算法		最小最大分区算法	
		构建时间/ms	渲染时间/ms	构建时间/ms	渲染时间/ms
Dragon+Bunny+F15	55	7125	2031	10735	2000
Statue+8 个 F15	75	8240	1312	11000	1359
Dragon+8 个 F15	75	7109	2234	9547	2360

数据表明,两种方法在最终划分粒度相同的情形下,使用本文的方法提高了 kd-tree 的构建速度,并得到了与最小最大分区算法大致相同的 kd-tree 质量(使用所生成的 kd-tree 进行渲染的时间数值基本相同).

在终止划分的对象数目的选取上,目前尚未有自动的、场景无关的机制. 实验发现,若划分终止条件设定的对象数目过多,会造成第一阶段的快速构建过程提早结束,使得输入到第二阶段传统的构建过程的对象数目偏大,导致构建的时间加长;反之,第二阶段的传统的构建过程会加快,但是第一阶段的构建过程容易产生更多的冗余节点,并容易造成

构建过程失败. 因此研究智能的终止条件对提高 kd-tree 构建过程的效能是有意义的.

4.2 应用改进的终止条件的构建时间比较

分区设置与前一节的方法相同,本文的方法使用两种终止条件:最小对象数和最小划分区间尺寸,满足其中之一即终止划分,转入第二阶段的构建过程. 最小最大分区算法使用相同的最小对象数目作为终止条件. 实验发现当最小对象数目减低至一定值时,最小最大分区算法失败. 通过设置最小尺寸值,在构建时间基本相同的情形下,本文实验中得到了更高质量的 kd-tree(使用所生成的 kd-tree 进行渲染的时间更短).

表 2 应用改进终止条件的构建方法与最小最大分区算法构建时间比较

场景	本文算法				最小最大分区算法		
	最小终止 对象数	最小区间 尺寸	构建时间/ ms	渲染时间/ ms	最小终止 对象数	构建时间/ ms	渲染时间/ ms
Dragon+Bunny+F15	25	1E-3	8625	1734	25	构建失败	
					55	10735	2000
Statue+8 个 F15	25	1E-3	10204	1062	40	构建失败	
					75	11000	1359
Dragon+8 个 F15	45	1E-4	7609	1954	45	构建失败	
					75	9547	2360

实验数据表明,与之前的方法相比,本文提出的方案使 kd-tree 的构建速度及所生成的树的质量都得到了改进。

作为一种基于分区的 kd-tree 构建方法,与传统的方法相比,其优势在于构建速度的显著提升,由此造成的副作用是使生成的 kd-tree 的质量下降. 一些研究人员开始尝试使用高级的遍历技术来补偿分区构建算法产生的 kd-tree 的质量下降造成的影响. 文献[11]使用 MLRTA^[1]算法进行 kd-tree 的遍历,在实验场景中达到了交互的水平. 基于分区的快速构建方法加之诸如 MLRTA 的等高级遍历算法是解决快速、交互甚至实时渲染应用的可行选择。

5 结论及未来工作

本文的主要贡献是提出了一种基于分区的快速 kd-tree 构建算法,通过对 kd-tree 成本函数的分析,本算法可以启发式地快速确定包含潜在最佳分割平面的子区间,提高了构建速度. 通过对子区间进一步地细化采样,得到了对最小成本的更好的逼近. 文章也对分区方法构建 kd-tree 的终止条件进行了分析,指出了已有方法存在的问题,提出了更合理的终止条件,使构建过程更加鲁棒。

文献[11-13,15]通过并行化实现使 kd-tree 的构建速度得到了极为显著的提高. 当前的工作中,我们实现的是算法的单线程版本,在第二阶段使用了 $O(n \log^2 n)$ 复杂度算法,未对内存管理进行优化. 本文的方法与其它的基于分区的算法一样,具有天然的并行属性,因此进行并行化实现及其访存优化是我们下一步要做的工作。

针对不同的场景,如何设置基于分区的 kd-tree 构建方法的各类参数,是一个开放的问题,需要进行进一步的理论研究和实验分析。

由于基于分区的 kd-tree 构建方法是两阶段过程,终止条件的设计对性能产生了重要影响. 过早结

束第一阶段过程,造成了子结点中对象数目多,使第二阶段的 SAH 算法耗时长. 另一方面,第一阶段构建的层次过深,会产生过多冗余节点,并影响最终的 kd-tree 质量,因此对终止条件做进一步的分析有重要意义。

参 考 文 献

[1] Reshetov A, Soupikov A, Hruley J. Multi-level ray tracing algorithm//Proceedings of ACM SIGGRAPH2005. Los Angeles, USA, 2005: 1176-1185

[2] Wald I, Slusallek P, Benthin C, Wagner M. Interactive rendering with coherent ray tracing. Computer Graphics Forum, 2001, 20(3):153-164

[3] Wald I. Realtime ray tracing and interactive global illumination [Ph. D. dissertation]. Saarland University, Saarbrücken, German, 2004

[4] Glassner A. An Introduction to Ray Tracing. San Francisco, USA: Morgan Kaufmann, 1989

[5] Wald I, Havran V. On building fast kd-trees for ray tracing, and on doing that in $O(n \log n)$ //Proceedings of the IEEE Symposium on Interactive Ray Tracing 2006. Salt Lake City, USA, 2006: 61-69

[6] Lext J, Assarsson U, Moller T. A benchmark for animated ray tracing. IEEE Computer Graphics and Application, 2001, 21(2): 22-31

[7] Macdonald J D, Booth K S. Heuristics for ray tracing using space subdivision. The Visual Computer, 1990, 6(3): 153-166

[8] Havran V. Heuristic ray shooting algorithms[Ph. D. dissertation]. Czech Technical University, Prague, Czech Republic, 2001

[9] Santalo L. Integral Geometry and Geometric Probability. Cambridge, UK: Cambridge University Press, 2004

[10] Havran V, Bittner J. On improving kd tree for ray shooting//Proceedings of the WSCG' 2002 Conference. Plzen, Czech Republic, 2002: 209-216

[11] Shevtsov M, Soupikov A, Kapustin A. Highly parallel fast kd-tree construction for interactive ray tracing of dynamic scenes. Computer Graphics Forum, 2007, 26(3): 395-404

- [12] Hunt W, Mark W R, Stoll G. Fast kd-tree construction with an adaptive error-bounded heuristic//Proceedings of the IEEE Symposium on Interactive Ray Tracing. Salt Lake City, USA, 2006; 81-88
- [13] Benthin C. Realtime ray tracing on current CPU architectures [Ph. D. dissertation]. Saarland University, Saarbrücken, German, 2006
- [14] de Berg M T, Katz M J, Van der Stappen A F, Vleugels J.

Realistic input models for geometric algorithm//Proceedings of the 13th ACM Symposium on Computational Geometry. Nice, France, 1997; 294-303

- [15] Popov S, Gunther J, Seidel H-P, Slusallek P. Experiences with streaming construction of SAH kd-trees//Proceedings of the IEEE Symposium on Interactive Ray Tracing 2006. Salt Lake City, USA, 2006; 89-94



FAN Wen-Shan, born in 1975, Ph. D. candidate. His research interests include real-time rendering, photo-realistic rendering, surface tessellation technique.

WANG Bin, born in 1976, Ph. D. , assistant professor. His research interests include computer graphics, computer-aided design.

Background

As one of the most important acceleration structure, kd-tree is used widely by most of the ray-tracing based photo-realistic rendering applications. There are two problems in relation to the use of kd-tree. One is how to traverse it as rapidly as possible. Much effort has been done to solve it, such as MLRTA. Another one is how to build high quality kd-tree in as less time as possible. The problem has been investigated just in recent years. This paper focuses on the latter.

The building procedure of kd-tree is a calculation intensive and tactical task. With the increase of complexity of the scene, the overhead paid for kd-tree construction has grown tremendously. So it is significant to present a fast kd-tree building method.

One of the problems of existing approaches on quick kd-tree construction is that they use fixed and uniform binning style, which incurs the dilemma that the used binning scheme can't accommodate to the random distribution of the nature scene. Another problem is that the fixed and uniform binning style needs the unnecessary iteration on the whole span. To address the problems, the authors propose a novel, adaptive

binning method by digging out the heuristic information from the SAH cost function. On the first step, the method tries to find the most-likely sub-span, which holds the splitting plane of current node. Doing as that can avoid the unnecessary iteration and speedup the building procedure. On the second step, the sub-sampling is used on the found sub-span to get the much better approximating splitting plane. After both of that, the more reasonable termination condition is used, which is helpful to improve the quality of resulted kd-tree and stability of construction procedure. The proposed method can be parallelized in multi-thread style easily as well. The experimental results show the effectiveness of the novel, fast kd-tree construction method.

The research was supported by the National High Technology Research and Develop Program (863 Program) of China (Grant No.2007AA040401), the National Basic Research Program (973 Program) of China (Grant No.2004CB719400), National Natural Science Foundation of China (grant Nos.90715043, 90818011, 60773143, 60533070).