利用球谐方法分散计算场景的全局光照

王 辉[1,2] 刘学慧[1] 吴恩华[1,3]

[1](中国科学院软件研究所计算机科学国家重点实验室 北京 100080)

[2](中国科学院研究生院 北京 100049)

[3](澳门大学科学技术学院电脑与资讯科学系 澳门)

(huihui4@sina.com)

摘 要 作为一种特殊的光照模型,辐射度在图形学中有着重要的应用. 然而,由于其计算的复杂性,辐射度方法一直得不到广泛的应用,特别是在动态场景中. 提出一种基于辐射度算法的全局光照实时加速算法,在原有基于点采样进行场景的全局光照近似的基础上,提出了不同的点采样计算方法,以提高辐射度方法对动画场景的适应性. 算法对场景中的物体建立六面体包围盒,并在包围盒各顶点设置采样点,这样就可以在环境变化对其光照影响不大的情况下采用原有的计算结果,实现辐射度算法在动态场景中的计算加速. 同时,对场景中的每个物体的光照,利用立方体映射计算直接光照;而对物体间的能量交换采用点采样方法进行近似计算并利用附近采样点的光照插值作为物体的光照. 实验结果表明,该算法可以减少大量的间接光照计算,提高了辐射度算法的效率.

关键词 辐射度;立方体映射;球谐方法

中图法分类号 TP391

Dispersive Calculation of Global Illumination Based on Spherical Harmonics Method

Wang Hui[1,2] Liu Xuehui[1] Wu Enhua[1,3]

[1](*State Key Laboratory of Computer Science, Institute of software, Chinese Academy of Sciences, Beijing* 100080)

[2](*Graduate University of Chinese Academy of Sciences, Beijing* 100049)

[3](*Department of Computer and Information Science, Faculty of Science and Technology, University of Macau, Macao*)

Abstract The radiosity illumination model plays an important role in computer graphics. As the calculation in the traditional radiosity method is time consuming, it is not suitable for dynamic scenes. In the paper we present a system based on the radiosity approach to speedup the calculation of global illumination. The algorithm follows the thoughts of using point sampling to simulate global illumination, while the sampling process is special so that the radiosity solution could be more adaptive in a dynamic scene. To every object inside the scene in the method, reference points are setup just at the vertices of the bounding box. When the scene is only slightly changed, the sampling result can be reused to reduce computation. The direct light is computed by cube-mapping and the energy exchange between objects is simulated by a trilinear-interpolation among the reference points. Experiments show that the method can accelerate rendering with less indirect illumination calculation required.

Key words radiosity; cubemap; spherical harmonicsemail

收稿日期:2006-06-24;修回日期:2007-01-29. 基金项目:国家"九七三"重点基础研究发展规划项目(2002CB312102);国家自然科学基金(60573155). 王 辉,男,1980 年生,硕士,主要研究方向为真实感图形、计算机动画. 刘学慧,女,1968 年生,博士,副研究员,CCF 高级会员,主要研究方向为计算机图形学、虚拟现实、计算机动画. 吴恩华,男,1947 年生,博士,研究员,博士生导师,CCF 高级会员,主要研究方向为计算机图形学、虚拟现实、科学计算可视化、计算机动画.

利用辐射度进行光能计算一直是图形绘制中的一项重要技术,使用目前的绘制技术可以绘制出漂亮的图形并表现出多种有趣的效果,如相近面片的颜色混合、强烈的间接光照等. 要表现这些效果,需要计算光能量是如何在场景里传播和反射的,辐射度算法正是用于此类计算的. 经过 20 年的研究,辐射度算法在光照效果表现及快速计算方面得到了很大的改善. 然而由于辐射度的计算量非常巨大,不仅需要多次迭代计算各个面片入射和出射的光能量,而且每次迭代过程都是大计算量的积分,所以目前还无法做到人机交互,特别是动态场景中的交互.

本文提出一种基于辐射度算法的全局光照实时加速算法. 对于场景内部的每一个物体,首先在其包围盒的 8 个顶点处用立方体映射技术得到光照情况;然后通过点采样和球谐函数将这 8 个顶点处的光照情况分别用若干个球谐系数来表示,用三线性插值的方法得到物体面片的球谐系数;最后结合球谐函数近似得到物体面片的光能量. 由于相邻 2 次绘制同一个物体的变化非常微小,每个物体的上述过程是相互独立的,因此一次绘制只要绘制场景内的部分物体即可. 这样就大大地降低了移动物体和改变光源时的计算量.

1 相关工作

在一个稳定的辐射度光照环境中,物体每个面片上的光照的情况可以用

$$L = Le + TL \tag{1}$$

来表示. 其中,L 是最终要求的光照值,Le 是直接从光源照射过来的光,T 表示光照传播算子,TL 表示从其他面片传播来的光照. 从式(1)可以看出,光照的传播是一个无限循环最终达到平衡的过程,这个过程是非常耗时. 研究人员提出了许多方法提高整体计算速度. Keller[1] 提出了 instant radiosity 的概念,实际上是将辐射度的计算与 photon mapping 思想[2]结合起来,将由光源发出的光粒看作光源,然后进行这些光源到物体的光照. 进一步,Nijasure 等利用球谐函数[3],通过计算均匀分布在空间中的点光照来近似全局光照,任意位置的点的光照由这些点的光照线性插值的方法来取得. 该方法以那些采样点上的能量估计作为所有面片出射能量和接收能量的中间过程,避免了面片之间两两相互计算辐射度的复杂过程,从绘制出的图形上看可以近似达到用辐射度的效果.

然而在实践中我们发现,由于文献[1]方法所计算的点光源是均匀分布的,会带来一些问题,如得到的计算画面被抹去了许多阴影而显得比较亮,更严重的是运动中物体的阴影出现间断现象. 对此本文提出了点采样计算方法,其基础来源于近年来在光照采样方面的进展. 下面简要介绍本文所借鉴的几种采样方法.

在预计算的光照传递算法[4]的基础上,Kristensen 等[5]提出了一种称为光团的采样点设置方法. 在空间中均匀地放上虚拟的光源点,对于场景中的每个点,看某 2 个光源对其贡献的差值. 差值足够小,则可以将这 2 个虚拟光源合并为一个,如此循环,原先的那些虚拟光源就变成了若干个团,一团用一个虚拟光源代表. 这时加进来一个光源则加亮其所属的团,光源移动则可以用某一团变暗、某一团变亮来表达. 同时,对场景中的物体上的点也可以用上述的方法聚类,一类中的点用相同的光照. 文献[5]估算了采样的误差,论证了其采样方法的合理性.

Walter 等[6]设想并论证了一种将光源点聚类的方法,将光源按照其位置组织成树状结构,叶节点是光源点,枝节点代表合并后的虚拟光源,还可以将这些虚拟光源合并成更高层次的虚拟光源. 比较枝节点与枝下各节点对场景贡献之和的误差,决定是否可以用枝节点替代枝下各节点.

Arikan 等[7]采用最终汇集的方法来累加计算环境光照,该方法对光照进行了区分,以一定的范围为界分为近处光和远处光. 远处光视为变化不大的、在空间中通过考察计算得出了一系列的采样点,计算采样点上的远处光的球谐系数,某点的远处光以邻近的采样点的球谐系数插值得到;近处光视为变化较大的,认为在上述所指范围以内的对向面都是可见的,利用这些面的辐射度和相对于绘制点的形状因子计算出近处光. 最终点上的光照 = 远处光 + 近处光 − 被近处面片遮挡的远处光.

上述几种采样方法中,采样点代表了对场景光照的模拟,因而相当重要. 文献[1]中所有采样点都设置在固定位置,这种方法存在以下不足:1)如果采样点比较稀疏则对场景的采样就粗略,影响绘制效果,设置得密集则每帧的采样工作计算量大,影响整体计算速度;2)对于动态场景较容易出现采样点失效的情况;3)每帧都必须对所有采样点建立立方体并计算该立方体的球谐系数,在此基础上插值计算所有的面片的光照. 另外,文献[1]方法中每帧都对所有面片作计算,没有利用到相邻帧的相似性.

本文针对这些问题进行改进:1)采样点随物体的运动而更新,大大降低了采样点失效的概率;2)每帧选用合适的采样点进行计算,既保证采样效率又减少了采样工作的计算量;3)利用相邻帧的变化较小、每帧可以只更新场景中的一部分物体的光照,将原本大量的计算分散到各帧,提高了效率. 虽然文中未能论证由采样点选取而导致的误差,但从实验结果来看,视觉效果良好.

2　本文的计算模型

2.1　算法思想

本文算法中各个物体都以其包围盒的 8 个顶点作为采样点,这就需要将场景中的物体分为 2 类:一类是场景内部物体,这类物体面片的法向量指向物体外部,物体本身和它的包围盒都在场景内部,便于使用其包围盒的顶点进行周围环境光照的采样;另一类是场景边界物体,如墙壁、地面、天花板,这类物体已经是场景的边界,法向量指向物体内部,其包围盒是场景的最外围边界,而且空间跨度大,不便于使用其包围盒的顶点采样周围的环境光. 第一类物体吸收 2 类物体发射出的能量,第二类物体则近似为它只吸收第一类物体发射出的能量,因此计算时只需分别独立地考虑每一个第一类的物体即可. 首先计算该物体从周围的 2 类物体吸收的能量,再计算它出射到第二类物体上的能量. 对于它出射到其他第一类物体上的能量,由于在计算相应物体从周围吸收的能量时能够得到,所以在此不必计算. 对于第二类物体的每个面片专门设置一个数据结构,记录光源直接照射的直接光照和各个物体向其发射的间接光能量. 传统的计算方法中,由于每帧都计算所有面片的能量,因此计算量非常大. 而实际上,即使是动态的场景,相邻 2 帧间的区别仅仅是物体稍微移动了一点点,这个差别是极其微小的,因此可以考虑不必每帧都重新计算所有的面片能量以降低计算量. 联想到当操作系统在处理多个进程时,所采取的调度方案是细分时间片,将时间片轮流分配给多个进程,这样从整体上感觉就是多个进程在并行处理. 与此类似,我们分别独立地考虑每一个第一类物体,在计算时,一帧就只针对少数的物体进行计算,从而将所有物体的计算分散到几帧中完成. 这样,减少了每帧的计算量,提高了绘制速度,而从整体上的感觉还是同时在处理所有的物体.

2.2　计算步骤简述

首先对场景进行初始化计算,包括细分面片、建立面片列表和相互对应关系、建立立方体形状因子查找表、设置物体包围盒的 8 个顶点为入射参考点、设置出射参考点等,而后的每一帧的计算都需要经过以下步骤:

Step1. 计算到场景边界物体的直接光照. 直接光照是具有自我发光能力的物体向外发射出的直接照射到其他物体的光照,为了避免与后续计算重复,此处不计算光源照射到场景内部物体的直接光照,只计算光源到场景边界上的直接光照,并将其记录在每个边界面片的数据结构中.

Step2. 选择一个场景内部的物体,计算它的光照. 与以往的方法不同,每一帧都不是计算所有物体的光照,而是有所选择地同时对接收到的直接光照和间接光照进行计算,采用球谐方法并插值.

Step3. 计算所选择的物体对边界的间接光照. 场景内部的物体从周围环境吸收能量,再将部分能量反射到环境中去,这部分就是间接光照. 将物体视为一个整体发射间接光能量,这个过程和直接光照非常类似. 由于计算时每帧所选择的物体一般不同,所以维护每个边界面片接收到的间接光照能量时只更新当前物体对它的影响.

Step4. 绘制场景面片. 场景每个面片的光能量包括物体自身拥有的光能量和从其他物体处吸收来的光能量 2 个部分. 将能量值转化为颜色值并绘制.

2.3　计算过程详述

2.3.1　场景初始化

在绘制计算中,由于一部分参数的取值是恒定不变的或者可以由一个取值经过简单变换得到,因而没有必要每帧都做重复计算,可以在初始化中预计算好以加快每帧的绘制速度. 实现中,初始化工作除了所必需的读入模型文件、建立物体链表和物体的面片链表、记录面片各顶点的位置和法向量、计算物体包围盒、细分场景的面片外,结合本文算法还需要:

1) 设置球谐函数的采样点. 计算球谐系数在理论上是要原函数与基函数对所有方位角进行积分,而在实际实现中只能对方位角采样,这一步的初始化先计算出在每个采样点处的球谐基函数的值.

2) 每个面片预计算一个立方体形状因子查找表和一个半立方体的形状因子查找表. 面片的形状因子对于面片的作用在于,面片实际接收到的能量是从各个方向上其他物体发出的能量和物体所在方向上的形状因子乘积的积分. 球谐方法的特性之一是 2 个函数乘积的积分可以转化为表示这 2 个函数

的球谐系数乘积的求和,所以可以利用这个特性简化计算. 此时预计算中形状因子视为一个以方向为变量的函数,投射到球谐基函数上,就得到了表示形状因子的一组球谐系数. 各面片的朝向各不相同,不同面片形状因子的查找表也不同. 但对于固定的某一个面片,它在物体的移动过程中只是位置发生了移动而方向没有变化,则它的形状因子查找表也就不发生变化,可以看作是面片的一个固定的属性,所以这一组表示形状因子的球谐系数也是不变化的.

3) 每个面片预计算它相对于物体包围盒顶点的三线性插值. 先期的球谐方法中,每帧都需要花费一定的时间搜索面片邻近的参考点,计算三线性插值系数. 本文中,由于计算的是面片相对于物体包围盒的 8 个顶点的三线性插值系数,这是不随物体的运动而变化的,因而也将这部分工作放在预处理中,绘制计算时引用即可,降低了绘制的计算量.

2.3.2 计算光源对边界的直接光照

光源对边界的直接光照是场景直接光照的一部分,计算了场景边界面片的直接光能量. 计算分为 2 个步骤,先以光源为中心绘制半立方体,再计算半立方体各个方向上相应面片光能量.

2.3.3 计算选定的场景内部物体的光照

一个场景从只有光源有光能量到最终场景内各个物体的光能量保持平衡,这中间的过程是通过许许多多帧,一帧绘制一个场景内部物体并更新场景边界. 假设有 n 个场景内部的物体,初始状态只有光源有能量. 第一帧只计算第一个内部物体的光能量并只绘制第一个内部物体和更新场景边界,第二帧在第一帧的基础上只计算第二个内部物体的光能量并只绘制第二个内部物体和更新场景边界,接下来的其他帧都是同样地在上一帧的基础上只绘制下一个物体和更新场景边界;绘制了 n 帧以后,每个物体的能量都计算了一遍. 这时,第 $n+1$ 帧在第 n 帧的基础上再绘制第一个物体,此时的场景已经与第一帧相比发生了很大的变化,所绘制的第一个物体更加接近平衡的状态. 不断地重复上述计算,就使得每个物体都一次次地不断接近平衡状态. 在这个基本方法的基础上,还可以对物体的绘制顺序有灵活的排序方式,如将几个物体作为一组,在一帧的计算和绘制中同时处理这几个物体,这种分组还可以是灵活变动的,可以对于移动较频繁的物体加大计算和绘制的频率. 这种分散计算的想法类似于操作系统中对进程的调度,当同时有多个进程时,操作系统采取将时间切成小片的方式,将各个小片的时间分派给各个进程,这样可以做到感觉上各个进程同时在被处理着,同时还可以通过不同的分派方法控制进程处理的进度. 在每一帧具体计算和绘制中,都包含以下 4 个步骤:

Step1. 场景物体选择. 由于每帧都只计算一部分的场景物体,而不是所有物体,因而每帧计算之初先确定将要计算场景的哪个物体,选择的物体必须是场景内部的物体,即第一类物体. 最简单的选择方法就是将场景内部的物体进行编号,各帧按编号顺序依次选择绘制. 在此帧其他的场景内部物体不作更新.

Step2. 在选定物体的包围盒顶点绘制 cubemap. 文献[1]方法在空间中设置固定采样点,与此方式相比,物体的包围盒顶点是紧随着物体的,采样时可以直接使用,不需要采样点定位的工作. 运动中的物体只需要随着运动更新自身包围盒顶点的位置和采样,而不是更新所有采样点处的采样,因而本文对采样点的设置方法更适合于对每个物体都相对独立地进行计算与绘制,同时物体的包围盒的顶点与物体本身较为贴近,采样的效果更好.

物体的包围盒是一个长方体,有 8 个顶点. 在各个顶点绘制立方体获得 8 个顶点处的环境光照,以此为基础近似计算物体接收到的环境光照. 任何一个顶点处绘制的立方体都有 3 个面来自于物体本身,但由于绘制立方体的作用是采集环境光照,因而可以不考虑来自于物体本身的光,将这部分置为 0.

Step3. 用球谐系数表示 cubemap. 球谐函数在整个球面定义了一系列正交基 $Y_l^m(\omega)$,用这些正交基,结合分布于整个球面的光照函数 $L(\omega)$,可以得到一系列的球谐系数 $L_l^m=\int_0^{\pi}\int_0^{2\pi}L(\omega)Y_l^m(\omega)\mathrm{d}\omega$. 而原来的光照函数是可以用这些系数和正交基还原出来的,$L(\omega)=\sum_l\sum_{m=-l}^{l}L_l^mY_l^m(\omega)$. 其中,$l$ 一个大于 0 的、可以人为调节的数,用来调节还原精度,低频光 l 值小些就可以达到较好的还原效果,有高光的话 l 值就需要大些. 辐射度的场景是很好的低频光的场景,所以是球谐函数理想的用武之地[8].

我们得到 cubemap 6 个面上的光照函数的一个离散表示,将理论上用积分求得的球谐系数用密集采样的方法来计算得到

$$L_l^m\approx\sum_{face=1}^{6}\sum_{i=1}^{size}\sum_{j=1}^{size}L_{\mathrm{face}}(i,j)Y_l^m(\omega)A(\omega),$$

其中,$size$ 是 cubemap 每个面的维度,$L_{\mathrm{face}}(i,j)$ 是 cubemap 中的一个面在(i,j)位置的取值,ω 表示像素的方向,$A(\omega)$ 表示这个像素的方位角在单位球面上所占的面积.

Step4. 插值计算每个面片的球谐系数,得到入射能量.

对于物体面片上的光照计算,由于已经预计算了面片相对于 8 个顶点的三线形插值的系数,用这些插值的系数和相应顶点的球谐系数进行插值计算得到面片的球谐系数,这样就可以还原光照函数了. 由于在预计算中已经用球谐系数表示面片的形状因子,根据球谐函数的性质,这 2 个系数向量的点乘结果实际上就是入射到面片上的光照. 根据入射光照结合面片的材质还可以计算面片出射的能量值.

2.3.4　计算所选择的物体对边界的间接光照

物体对边界的间接光照是场景间接光照的一部分,它计算了场景边界面片的间接光能量. 计算分为 3 个步骤:

Step1. 计算每个面片对出射球谐系数的贡献. 实现中将一个物体视为一个整体向外出射光能量,使用一组球谐系数来表示物体向各个方向出射的能量,这个球的中心就设置为与物体的中心相同的位置. 物体向外出射的光能量本质上是由物体的每个面片向外出射的光能量累加而成的,所以对物体的每个面片计算它对出射球谐系数的贡献. 由于球谐函数表示的光出射是以物体中心为出射原点的球,而面片的光出射是以面片中心为原点、以面片法向量为出射主方向的,因而如果将面片对各个方向光能的贡献当作物体出射的一部分则误差较大. 为了减小些误差,实现时进行了折中,将面片的出射方向取为物体中心指向面片中心的向量和面片法向量的中间值. 这样,近处的面片接收能量时能大大地减小误差,稍远的物体接收到的能量较弱,所以影响不大.

Step2. 以物体中心为立方体的中心绘制物体间接光照能量出射 cubemap. 由于在初始化的过程中已经建立了半立方体的形状因子的查找表,因此用光源面片的能量和该方向上的形状因子就能够计算在该方向上出射的光能量.

Step3. 更新相应面片光能量. 物体出射的间接光照只对场景边界上的面片作计算.

2.3.5　绘制

面片接收到的直接光照能量和间接光照能量构成了面片的总能量. 能量最大的面片是场景中最亮的,颜色值也最大;能量最小的面片是场景中最暗的,颜色值也最小;其余面片按照其能量值介于最大值与最小值之间的比例,对应出颜色值介于最大值和最小值的比例,从而将能量值转化为面片的颜色值进行绘制.

3　实　　验

本文实验环境为 P Ⅳ 3.0 GHz CPU,1.0 GB 内存. 实验场景为漫射场景.

首先根据对文献[1]方法的理解进行了模拟实验,选择 16 个球谐系数,立方体映射的分辨率为 128×128.

当网格点数为 $4\times4\times4$ 时,实验数据如表 1 所示.

表 1　文献[1]方法在 $4\times4\times4$ 网格点时的耗时

场景面片数	计算耗时/ms
12 246	1 069
26 257	4 258
37 087	5 693
62 601	8 632
76 831	10 421
129 057	20 659

当网格点数为 $5\times5\times5$ 时,实验数据如表 2 所示.

表 2　文献[1]方法在 $5\times5\times5$ 网格点时的耗时

场景面片数	计算耗时/ms
12 246	1 234
26 257	4 667
37 087	6 240
62 601	9 427
76 831	11 451
129 057	22 200

文献[1]方法中,如果网格点位于物体内部,则会造成该网格采样点的失效,如图 1 所示.

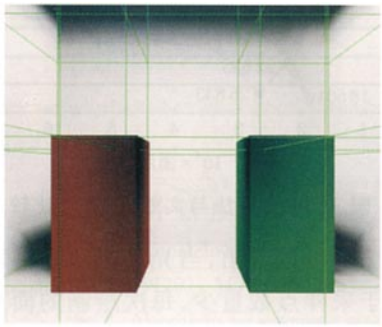

图 1　采样点失效

本文实验中对各个规模场景内物体的面片数量、每个物体采样点都是它包围盒的 8 个顶点、球谐系数的个数都是一个平方数,当选择 16 个球谐系数时相比选择 9 个球谐系数时的绘制结果改善较大,而选择 25 个球谐系数时的绘制结果改善不明显,所以最终选定计算 16 个球谐系数值;立方体映射中每个面上的分辨率上,通过对各种分辨率的实验比较看出,精度为 128×128 的分辨率已满足需要.

表 3,图 2 所示为绘制不同面片数量场景的时耗情况,以及各个步骤所花费的时间.

表 3 本文算法耗时统计

场景面片数	场景内部物体面片数最大值	绘制面片最多物体/ms				绘制一帧最大耗时
		光源对边界的直接光照耗时	场景内部物体光照耗时	所选物体对边界的间接光照	总耗时	
18 501	280	391	312	70	773	773
26 833	5 068	503	374	1 856	2 733	2 733
22 635	14 286	641	437	2 413	3 491	3 491
32 919	24 570	797	578	2 959	4 334	4 334
42 752	34 406	890	704	3 476	5 070	5 070
53 247	44 811	1 031	891	4 022	5 944	5 944
62 106	53 670	1 156	1 015	4 507	6 678	6 678
75 427	63 684	1 297	1 187	5 016	7 500	7 500

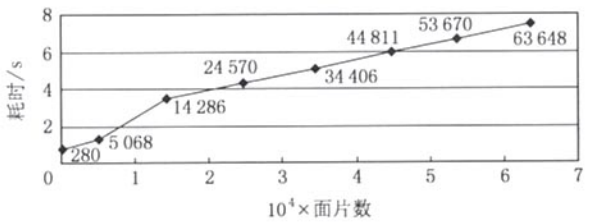

图 2 本文算法耗时

在绘制速度上,本文算法与文献[1]方法比较如图 3 所示.

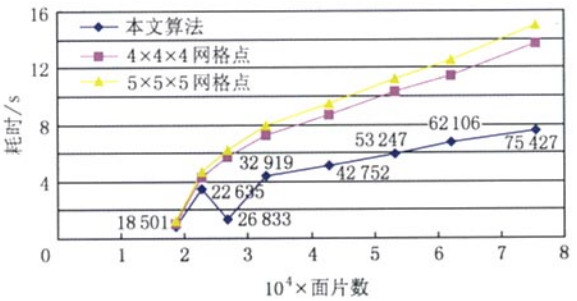

图 3 本文算法与文献[1]方法比较

从图 2,3 可以看出,当光源面片数量增多时,本文算法由于采样点数量少,每次绘制的面片数量少,所以耗时较少.

本文算法的实验结果如图 4～7 所示. 图 4 中,在物体包围盒顶点处的蓝色粗点处计算球谐系数;图 5 中,面片数为 2 万,物体面片上的光照经由包围盒顶点的球谐系数插值后再还原而得到.

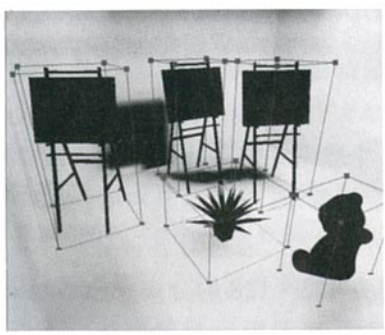

图 4 内部物体未绘制时

图 5 物体绘制后

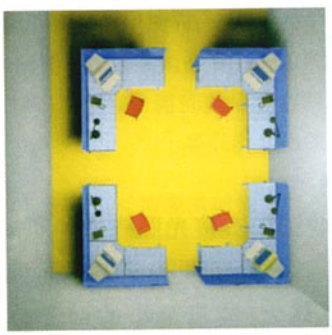

图 6 7 万面片场景俯视图

图 7 7 万面片场景平视图

图 4～7 中,细分场景面片时将墙面及地面划分为小正方形,由此导致物体阴影有一定的锯齿.

4　结论和下一步的工作

本文在辐射度场景的计算中利用了球谐方法的一些性质,基于每个物体独立计算,大大地加速了场景绘制. 该算法中,采样点的选择是很重要的环节. 针对场景中的每个物体,如何更进一步地优化采样点的选取,以及估算采样点选取的不同而带来的误差等,是值得深入探讨的问题 还可以考虑根据球面谐波的差异自适应地增加球面谐波的采样密度 ;随着场景的复杂度的提高,尤其是光源数量较多的情况下,可以考虑对光源进行合并和简化. 另外,目前的工作是在 CPU 上的实现的,而时下 GPU 发展迅速,如果利用 GPU 的可编程的特性,则可以将绘制速度进一步提高到可以进行实时计算.

参　考　文　献

[1] Keller Alexander. Instant radiosity [C] //Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH, Atlanta, Georgia, 1997:49-56

[2] Purcell Timothy J, Donner Craig, Cammarano Mike, *et al*. Photon mapping on programmable graphics hardware [C] // Proceedings of the ACM SIGGRAPH/Eurographics Conference on Graphics Hardware, San Diego, California, 2003:41-50

[3] Nijasure Mangesh, Pattanaik Sumanta, Goel Vineet. Interactive global illumination in dynamic environments using commodity graphics hardware [C] //Proceedings of the 11th Pacific Conference on Computer Graphics and Applications, Canmore, 2003:450

[4] Sloan Peter-Pike, Kautz Jan, Snyder John. Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments [C] //Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH, San Antonio, Texas, 2002:526-536

[5] Kristensen Anders Wang, Akenine-Moller Tomas, Jesen Henrik Wann. Precomputed local radiance transfer for real-time lighting design [C] //Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH, Los Angeles, California, 2005:1208-1215

[6] Walter Bruce, Fernandez Sebastian, Arbree Adam, *et al*. Lightcuts: a scalable approach to illumination [C] //Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH, Los Angeles, California, 2005:1098-1107

[7] Arikan Okan, Forsyth David A, O'Brien James F. Fast and detailed approximate global illumination by irradiance decomposition [C] //Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH, Los Angeles, California, 2005:1108-1114

[8] Ramamoorthi Ravi, Hanrahan Pat. Frequency space environment map rendering [C] //Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH, San Antonio, Texas, 2002:517-526

(上接第 608 页)

[4] Dou Yikang. An automatic adaptive full quadrilateral mesh generation with a point-by point insertional method [J]. Chinese Journal of Computational Mechanics, 1997, 18(3):317-323 (in Chinese)
(窦一康. 用逐点插入法造成全四边形自适应有限元网格 [J]. 计算力学学报, 1997, 18(03):69-75)

[5] Bowyer A. Computing dirichlet tessellations [J]. The Computer Journal, 1981, 24(2):162-166

[6] Watson D F. Computing the *N*-Dimensional Delaunay Triangulation with Application to Voronoi Polytopes [J]. The Computer Journal, 1981, 24(2):167-172

[7] Lawson C L. Software for C^1 Interpolation [M] //Rice J. Mathematical Software Ⅲ, New York:Academic Press, 1977:161-194

[8] Borouchaki H, George P L, Lo S H. Optimal Delaunay point insertion [J]. International Journal for Numerical Methods in Engineering, 1996, 39(20):3407-3437

[9] Li S S, Shi John Z. Algorithms for automatic generating interior nodal points and Delaunay triangulation using advancing front technique [J]. Communications in Numerical Methods in Engineering. 2006, 22(5):467-474

[10] Shewchuk Jonathan Richard. Triangle:engineering a 2D quality mesh generator and Delaunay triangulator [C] //Proceedings of the 1996 FCRC Workshop on Applied Computational Geometry, Philadelphia, PA, 1996:203-212

[11] Sarrate J, Palau J, Huerta A. Numerical representation of the quality measures of triangles and triangular meshes [J]. Communications in Numerical Methods in Engineering, 2003, 19(7):551-561