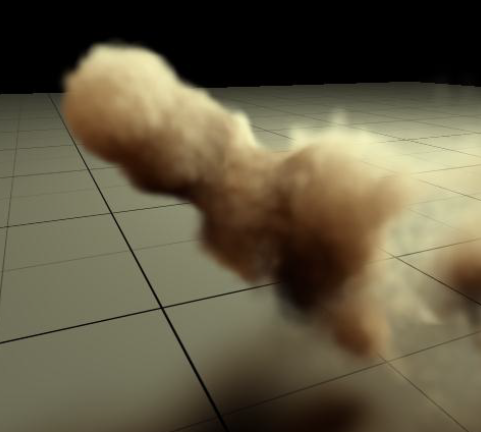
第三章 粒子表示的介质体碎片云效果渲染算法

3.1概述和相关工作

由第一章所述，以SPH，MPM等方法为代表的无网格法最主要的一些用途就是处理一些大变形、流固耦合、冲击破碎、爆炸等物理过程。本文所设计的后处理系统显然应该能够很好地渲染这些物理过程的仿真结果。这类问题当中，我们经常需要渲染比如冲击侵彻后的碎片云以及爆炸后的烟雾火光效果，不同于上一章所讲的表面重构，这类效果的物质通常没有明确的表面信息，而且一般是半透明材质，所以不能使用传统的基于三角形面片的渲染框架进行渲染，这类物质一般称为介质体。

对于这类物体，Reeves于1983年创立了粒子系统【1】，该系统后来被广泛应用于烟雾、火、爆炸等自然现象的模拟；其特点是能够以很高的效率实现较为真实的视觉效果；然而，该方法的粒子是通过人为的规则控制其产生、运动和消亡的，并没有遵从真实物理过程。而且其渲染过程也是更多地使用了人工着色再结合半透明混合算法，难以实现真实感的视觉效果。

为了给粒子系统引入更加具有真实感的视觉效果，人们提出了很多体阴影算法来实现更真实的光照效果，有代表性的如深度图算法【2】以及如【3，4】等的一些列改进算法，这类算法大多数都是对待渲染物体进行切片，并对每一片计算阴影图，最终使用混合这类传统渲染方式将图形显示到屏幕上。这类方法已经可以实现效果比较理想的介质体渲染，且一般效率比较高，如【4】中使用GPU实时渲染的烟雾：



但是这类方法的不足之处是其大多针对单光源的情况，很难使用固定方向上的阴影图来实现不同方向上的多光源照明。而且这类方法处理的问题也多数是简单照明，处理复杂光照环境及全局照明上存在一定困难，而且其普遍采用的切片处理方式导致了处理和其它物体相交的情况比较复杂。另外，由于像【4】中算法主要用于渲染烟，而由于烟的照明系数较低，全局照明对其效果影响不显著，所以也没有必要考虑此因素。

同时，由于近些年基于物理的仿真在图形学界的流行，人们提出了很多基于流体动力学的烟雾仿真算法【5,6】，这类算法大多使用欧拉网格进行仿真计算和结果的渲染，而在空间体素网格结构上可以很容易地使用体光子映射【7】等全局照明算法，且能够处理复杂多光源及非均匀介质的情况，所以能够生成质量非常高的真实感图形，如图【】



结合本文后处理系统的需求及前人工作，本章做了以下工作：

1. 分析碎片云的特点，选定适合本后处理系统的算法类型
2. 提出一种基于测试光子的自适应的光子映射增密算法
3. 提出一种可以体现模型动态效果的运动模糊算法，以便使生成的动画更加真实
   1. 算法类型的选定及算法框架

3.2.1 算法类型选定

本文的后处理系统所要渲染的无明确表面物体主要有以下几种形式：爆炸产生的烟雾和火光、高速冲击侵彻产生的固体碎片云以及液体的雾气效果。其中碎片云中可能会有一些体积稍大的小碎块，这部分可以用第二章的表面重构算法处理；而体积更小的碎片通常由于其已经远远小于单个像素大小且数量极其巨大，而且在无网格法的计算结果中通常已经是当个粒子，所以这种情况已经不适合于进行表面重构，它在本质上更接近于烟，于是这些效果都适合于使用粒子系统进行渲染。由于本文中的后处理系统更主要的目标是获得高质量的真实感图形，而对渲染实时性的要求并不高，而且有些介质体的材料具有较高的照明系数，这时若不考虑全局光照，便不能获得正确的光照效果，如高速冲击形成的水雾等效果，即便是照明度低的材料，其碎片云的浓度也一般会比较高，如图【】，这时如果不考虑全局光照，也会造成结果的不准确，所以以阴影图为代表的方法并不适合本文。

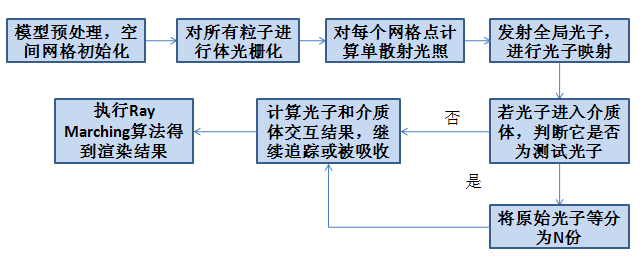


本图放在绪论

于是，基于欧拉网格的体渲染方法就更能够符合本文的要求。而本文针对的是无网格法结果的渲染，所以为了适应欧拉网格的渲染框架，需要将粒子数据进行体光栅化。虽然【6】中提到了一种粒子数据的体光栅化方法，即利用了粒子半径信息进行采样的一种算法，但是并没有讨论粒子的半径对渲染结果的影响。实际情况是，很多无网格法并没有粒子的半径信息，而且很多情形下，比如高速冲击的碎片云，其粒子间距差异可能较大，这不同于烟雾粒子的相对均匀的分布情况，这时如果使用单一半径进行体光栅化，容易出现以下两种情况：当粒子半径过大时，渲染结果会出现不连续的球状颗粒；当粒子半径过小时（特别地，小于空间网格尺寸很多）会出现大量粒子没有被采样到的情况，造成渲染结果的失真。

基于以上分析，本章采用了对模型数据进行预分析，计算得到每个粒子的合理半径并进行体光栅化，然后使用光子映射法模拟全局光照，最后使用光线步进算法对模型进行光照和透射率的采样和累加并最终生成结果图形。

3.2.2算法框架



算法流程图如图【】，由于需要处理的碎片云介质体一般是浓密且非均匀的，所以为了保证足够的渲染精度，在光子映射阶段需要用到的光子数比二维表面要大很多，于是对介质体需要专门地多投射光子，于是引入了测试光子来判断哪些地方需要增加投射光子的密度。

3.3 体光栅化及渲染方法

首先根据第4章的内容，我们可以计算得到每个粒子的邻近粒子平均半径，然后定义被采样粒子半径为



其中d为空间网格的长度

采样粒子密度分布选取以粒子位置为中心的高斯分布：



，其中为采样网格点到粒子中心的矢径，对每一个采样粒子求其AABB包围盒，并选取包围盒内的网格点进行采样，将采样后的密度场值在该网格点上累加，遍历所有粒子，完成体光栅化过程。

渲染方法采用从前到后（front to back）的光线步进算法对透射率（不透明度）和光照信息进行更新，这样做的优点是当透射率已经接近0时可以停止步进而不用计算接下来的对总光照几乎没有影响的部分。步进采样时，步进长度为网格尺寸d，在采样点对立方体网格的8个结点进行插值，插值方式采取立方体的三线性插值；插值后的透射率和不透明以及体光源值度需要乘上步进长度d。

* 1. 体光子映射算法的实现

3.3.1算法概述

与前人工作相同，本文的光子映射过程也只针对间接光照，即只存储与场景中的物体碰撞（或交互）次数大于一次的光子。

当光子进入介质体后进行路径追踪时，使用和光线步进算法一样的距离d步进。每个步长计算光子位置处的密度场值，用上一节的方法插值得到，以密度场值为概率使用轮盘赌算法决定光子与介质体交互还是不受影响径直穿过；若光子与介质体交互，则以散射率为概率决定光子被散射还是吸收；若光子被散射，则根据由散射相位函数



预先计算好的概率函数生成光子的散射方向，并沿该方向继续追踪光子；

若光子被吸收，如果其碰撞次数大于1，则将其存入k-d树中，用于空间位置的光照强度计算。

光照强度的计算，使用基于k-d树的BBF-KNN算法搜索邻近的50个光子，光强计算公式如下：



其中是一个不同于公式3-1的相位函数，因为对于碎片云这种介质体，其内部的颗粒并非足够小，光线穿过这样的介质体时并非只有散射成分，而还有相当一部分反射成分，所以本文中采用了如下定义的：

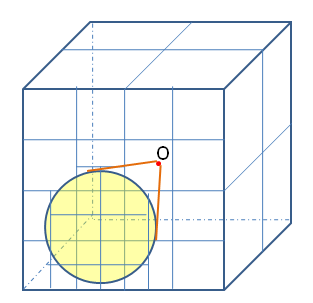


其中为散射率，这样一来，照明度高的介质（如水汽）光线主要被散射，否则（如煤烟等）主要被反射。

3.3.2 发射光子的自适应增密

由于光子映射算法需要使用大量的光子来保证渲染收敛到正确的结果。然而，由于场景中不同位置的物体分布可能不均匀，而不同类型的物体需要的光照计算精度差异也很大，这导致了某些区域需要大量光子才能够实现足够的光照计算精度，而另外一些区域只需要很少的光子就可实现要求的精度，有些地方甚至无需计算光照（如渲染视野之外或者天空及距离非常远的场景）。这就导致了如果整体增加发射光子的密度，势必造成总体光子数量过于庞大，浪费内存并严重降低渲染效率；而如果光子发射密度太低则会导致很多光照细节无法实现。于是我们必须考虑向场景中的不同区域自适应地发射不同密度的光子。本章采用两层增密的算法：第一层针对整体场景，使得光源只在有物体的大致区域发射较高密度的光子，而其它区域只发射极少数的光子；第二层针对需要实现高精度全局光照的部分，如介质体，在上一层增密的基础上，对介质体的区域进一步增密。

为了实现第一层增密，本文引入一个空间立方体盒结构来生成发射光子，如图【】：



其中O点为光源，在光源外构造一个立方体盒，使光源位于盒子中心，然后将盒子的每个面按照特定尺寸剖分为一系列正方形网格，将光源和每个正方形面片连成一个以光源为顶点的四棱锥，然后向每个四棱锥的中轴线方向发射一个光子。记光源的总光通量为，记某个四棱锥所所张的立体角为，则每个光子的能量为



对于不需要加密的区域，只需要构造比较大的正方形网格即可；对于需要加密的区域，可以从光源出发，向需要加密的区域引一个圆锥，计算合适的张角，使该圆锥能够将所有需要加密的区域囊括在内；遍历立方体盒表面的正方形网格，将位于圆锥内的网格进行细分，细分的尺寸根据增密的精度要求确定，之后对每个细分后的网格发射一枚光子，如此便实现了第一层增密。

为了实现介质体区域发射光子的自适应加密，可以在光子的结构中增加一个是否为测试光子的字段，并将所有从光源处发射的光子都置为测试光子，当对一个光子进行路径追踪时，若光子进入介质体，且在步进到某位置时密度场值不为0，则不继续对该光子进行追踪，追踪函数直接返回一个代表进入介质体的值，此时，在测试光子的发射方向上新增N个非测试光子，每个光子的能量为原光子的，各非测试光子的方向在原光子所张立体角范围内平均分布，此时再对各非测试光子进行正常追踪。如果测试光子不经过介质体，则其是否为测试光子对光子其它行为没有任何影响，可以正常实现二维表面上的光照计算。

* 1. 碎片云的运动模糊效果的实现

碎片云不同于普通的烟雾和云等自然现象，碎片云一般是告诉冲击或爆炸产生的，其中的物质粒子会以很高的速度运动，由于人眼的视觉暂留、照相机的曝光视觉等因素，碎片云的影像都会在粒子的速度方向上有一个运动模糊，从图【】中可以看到这一点。

实现运动模糊一般有以下两种策略：

1. 将运动的物体的若干帧画面进行混合
2. 在一帧内根据物质不同部分的速度信息在速度方向上进行一定的拉伸及模糊操作

其中方法1实现起来比较简单且效率较高，但是对于无网格法仿真数据来说，并不合适。首先因为若对渲染出的若干帧图形整体进行模糊的话，因为一般情况下模型各部分速度方向不同，不能将其当做单个物体，这样直接混合有可能会导致整个图形的混乱。其次由于碎片云中的粒子运动速度一般很快，两帧之间的粒子位置可能差异极大，甚至碎片云部分完全不是同一批粒子，直接混合已经没有意义。所以此类问题更适合使用第二种方法。

前面提到的，在对粒子进行体光栅化的过程中，我们把粒子看做了一个密度场沿半径衰减的球体，如果将球体换成椭球体，使其长轴沿着速度矢量的方向，这样便会使碎片云产生速度方向上的模糊。

取两根短轴的长度均为粒子半径，长轴长度为短轴长度的一个大于1的倍数，有



其中

定义以粒子位置为原点，速度方向为z轴正方向的局部坐标架，即，的方向可随意选取，只需保证各轴正交，这里使用任一不与平行的向量与作向量叉积并单位化得到。

将该坐标架的三根轴向量合并为一个正交旋转矩阵：



并定义对角阵



于是便可以定义描述该粒子对应的椭球体的矩阵



于是对每个粒子对应的椭球体进行体光栅化时，可将公式3-2替换为



3.5结果与讨论

3.6 总结与展望

本章通过对本文后处理系统的需求和碎片云特点的分析，选定了先将粒子体光栅化，随后使用光子映射法实现全局照明，最后使用光纤步进发进行体渲染的算法。并且为了满足渲染精度要求，提出了一种基于测试光子的投射光子增密算法。为了使碎片云效果看起来更加真实，使用了粒子的速度信息实现了碎片云的运动模糊效果。

但是以上算法存在内存消耗较大，需要的预处理较长等缺点。尤其是体光栅化的过程，由于粒子数目庞大，且增加运动模糊效果后，由于每个粒子的采样椭球体较长，包围盒中囊括的网格点数目过大，导致了体光栅化的过程非常缓慢。这些地方都还有很大的改进空间，但是由于时间所限，本文并未针对这些问题去仔细优化，可以考虑的优化方法包括，将全空间的欧拉网格改为空间哈希网格，这样可以节省部分内存；将椭球体的AABB包围盒改为任意平面包围盒，这样可以极大地减小包围盒的体积；最后，由于该方法仍具有高度的并行特点，以后的工作中可以考虑对其进行GPU并行化。

【2】LOKOVIC, T., AND VEACH, E. 2000. Deep shadow maps. In Proceedings

of the 27th annual conference on Computer graphics

and interactive techniques, 385–392.

|  |  |  |
| --- | --- | --- |
| 【3】Fourier opacity mapping  Jon Jansen, Louis Bavoil | | |
| February 2010 |  | I3D '10: Proceedings of the 2010 ACM SIGGRAPH symposium on Interactive 3D Graphics and Games |

【4】

Volumetric Particle Shadows

Simon Green

NVIDIA CUDA Samples Doc July 2012

【5】

|  |  |  |
| --- | --- | --- |
| [Visual simulation of smoke](http://dl.acm.org/citation.cfm?id=383259.383260&coll=DL&dl=ACM&CFID=668643590&CFTOKEN=35085075)  [Ronald Fedkiw](http://dl.acm.org/author_page.cfm?id=81100612327&coll=DL&dl=ACM&CFID=668643590&CFTOKEN=35085075), [Jos Stam](http://dl.acm.org/author_page.cfm?id=81100148921&coll=DL&dl=ACM&CFID=668643590&CFTOKEN=35085075), [Henrik Wann Jensen](http://dl.acm.org/author_page.cfm?id=81100640205&coll=DL&dl=ACM&CFID=668643590&CFTOKEN=35085075) | | |
| August 2001 |  | **SIGGRAPH '01:**Proceedings of the 28th annual conference on Computer graphics and interactive techniques |

【6】

|  |  |  |
| --- | --- | --- |
| [Smoke simulation for large scale phenomena](http://dl.acm.org/citation.cfm?id=1201775.882335&coll=DL&dl=ACM&CFID=668643590&CFTOKEN=35085075)  [Nick Rasmussen](http://dl.acm.org/author_page.cfm?id=81100172208&coll=DL&dl=ACM&CFID=668643590&CFTOKEN=35085075), [Duc Quang Nguyen](http://dl.acm.org/author_page.cfm?id=81100511689&coll=DL&dl=ACM&CFID=668643590&CFTOKEN=35085075), [Willi Geiger](http://dl.acm.org/author_page.cfm?id=81100325446&coll=DL&dl=ACM&CFID=668643590&CFTOKEN=35085075), [Ronald Fedkiw](http://dl.acm.org/author_page.cfm?id=81100612327&coll=DL&dl=ACM&CFID=668643590&CFTOKEN=35085075) | | |
| July 2003 |  | **SIGGRAPH '03:**SIGGRAPH 2003 Papers |
| 【7】H. W. Jensen and P. H. Christensen. Efficient Simulation  of Light Transport in Scenes with Participating Media using  Photon Maps. In *SIGGRAPH 98 Conference Proceedings,*  *Annual Conference Series*, pages 311–320, July 1998. |  |  |