2.1概述及相关工作

在以SPH，MPM等为代表的各类无网格法仿真计算的结果表示中，多以粒子的形式存在。各粒子携带有物质的质量、速度、温度等各种物理量，而其最重要的一个特点就是，粒子的位置分布是非规则的，所以这也造成了其所代表的模型很难从这样的非规则分布粒子中提取表面信息。

针对这类问题，隐式表面方法是一种被广泛采用的重构表面信息的方法。该方法主要有两种形式，其一为Hoppe等人【1】提出的符号距离函数法，该方法分两个步骤：首先通过一点附近邻域内的点集进行线性拟合，得到一个局部的正切平面；然后对于空间中任意一点，计算到的距离函数，若位于表面外则记为正，反之为负。则该重构的表面便可以用隐式地表达。其二为M¨uller等人【2】使用的用标量场的等值面定义表面的算法。该算法在每一个粒子的位置定义一个“核”，空间一点处的值为该点到核心的半径的函数，对空间中任意一点，该处的标量场值为所有“核”在该点处贡献的叠加。

第一类方法适用于粒子数量较大、密度分布较为均匀的情况，而当模型局部的粒子比较少的时候，通过局部拟合来构造表面的方法明显不合适，而第二类方法却不受这个限制，即使只有一个粒子，也可以完成表面构造。而本文需要处理碎片等情况，会有局部粒子较少的情况，所以考虑使用第二类方法。

本文中此部分的功能主要基于Yu和Turk在2013年的工作【3】，这篇文章主要基于【2】的方法，但是通过引入各向异性核，并加入表面粒子位置调整功能从而实现了更加光滑的表面构造，并且能够保持边角棱等模型的局部细节。本章针对【3】中的一些不足之处对其中的算法提出一定改进，以及结合本文后处理模块的特点和需求，对其算法进行了适用性扩展，主要工作如下：

1.【3】中的主要效率瓶颈为标量场的计算，其算法要为模型中的每个粒子计算标量场值，这消耗了大量时间，同时存储所有网格结点也会耗费较大内存。本文通过引入一种增长型八叉树结构，可以实现只计算模型表面附近的标量场从而达到大大降低计算时间和减小内存消耗的目的；同时实现了空间划分，加速光线跟踪算法中光线和重构出的三角形表面求交运算的速度。

2.由于【3】中算法仅仅是针对算例，需要根据模型数据的不同手工调试参数，这对于后处理系统来说明显不现实，于是本文设计了一套对数据文件的自动分析模块，具体实现参考第四章。该模块的分析结果在表面重构中可以动态自动设置标量场值范围、粒子半径及邻近粒子搜索半径等参数。

2.2各向异性核重构算法描述

2.2.1表面的隐式表示

对各向同性核，标量场可以表示为



其中j为所有影响到位置 的粒子的编号，W即为每个粒子处的各向同性核，形式如下：



其中是一个缩放值，d为仿真的维数，h为核心的影响半径，一般由仿真计算阶段提供，为某点到核心的位移矢量 ，P为一个对称的衰减函数，用来描述场变量值在核内的分布。分母中的保证了场值在一个核内的积分为固定值，这样可以保证不同粒子对整体表面的贡献度只和该粒子所实际代表的物理量相关，不会受粒子间距的影响。将所有影响到任意空间点的核在处的场值累加后便得到了该点最终的场值，在设定阈值t后，该标量场的t等值面便为该表面的隐式表示。

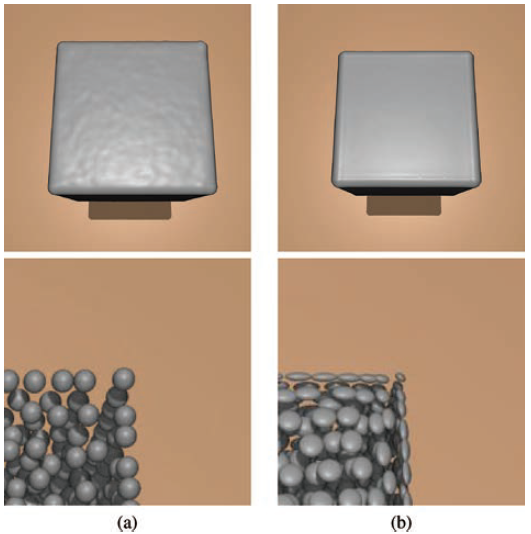
对各向异性核，【3】中使用了一个的正实数矩阵来代替，这使得核心从球体变成了椭球体（椭圆），的各阶特征值的倒数代表该椭球体的各主轴长度。此时核心表示为



于是场变量表示为



其中是影响域内第j个粒子为了平滑表面而被调整后的位置。其中矩阵的构造需要用到邻近粒子的信息，即要进行邻近点搜索，并有构造过程中还需要计算矩阵特征值和特征向量的操作，于是这是该算法最耗时的地方。不同核心的对比效果如图【】所示



其中（a）是各向同性核的表示及其效果，可见容易产生表面的凹凸不平，(b)是各向异性核的效果，可见表面明显比各向同性核要平滑很多。

2.2.2 隐式表面的显式表示方法

由于使用以上算法重构出的表面时隐式表示的，不能够直接用来渲染生成最终图形，所以需要对该隐式表面进行采样表示。

对于光线跟踪渲染框架，一种比较直观的办法是在追踪光线的运行路径上进行二分查找，即以一个初始步长向前移动光线前端，计算每一步光线前端处的场变量值，当该值大于等值面阈值后，将步长二分，并将光线前端退回新步长距离，如此二分下去，直到步长值小于给定精度要求。过程如图【】所示。在找到交点后，根据标量场的梯度计算该点法向量即可实现光照计算完成渲染。

该方法的优点是精度非常高，因为它可以精确计算光线和物体表面的交点，但其缺点是每次需要用到物体表面的某一点信息时，都需要重新寻找该点，造成巨大的浪费，在进行光子追踪，阴影计算和表面光照计算时很可能若干次用到该点信息，这样将造成巨大的浪费，导致效率极端低下，于是现在比较流行的算法是Lorensen 和 Cline 在1987年提出的Marching Cubes算法【4】。

该算法通过在数据区内构建空间规则网格，并在网格点处对标量场进行采样，然后针对每个立方体单元，根据角点和重构表面的相对位置来在单元内构造三角形面片，找到全部表面的面片后再进行传统渲染。该算法的优点是对表面进行离散采样，速度较快，且生成的三角面片方便各种渲染体系进行渲染，适用性比较广；但是其缺点是需要构建空间规则网格，并计算所有网格点的场变量值，当模型比较巨大的时候，其运算量和内存消耗量都是非常可观的。

2.3基于扩展型八叉树的算法改进

2.4重构算法中各参数的自适应调整

2.5算法的实现及结果

2.6总结与展望

[1]Hugues Hoppe, Tony DeRose, Tom Duchamp, John McDonald, Werner Stuetzle. Surface reconstruction from unorganized points. Computer Graphics (Proceeding of ACM SIGGRAPH 92),26(2):71-78,1992

【2】M¨ULLER, M., CHARYPAR, D., AND GROSS, M. 2003. Particle-based fluid simulation for interactive applications.

In Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation. Eurographics Association, 154–159.

【3】Yu, J. and Turk, G. 2013. Reconstructing surfaces of particle-based fluids using anisotropic kernels. ACM Trans. Graph. 32, 1, Article 5 (January 2013)

【4】LORENSEN, W. E. AND CLINE, H. E. 1987.

Marching cubes: A high resolution 3d surface construction algorithm.

In Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques. ACM, New York, 163–169.