1. 绪论
   1. 课题研究目的和意义

随着科学技术的迅猛发展，科学计算在各行各业中起到的作用越来越重要，其所涉及到的对象越来越多，问题越来越复杂，计算规模也越来越大；而这时从海量的计算数据中挖掘有效的、具有指导意义的信息就变得越来越困难，于是也变得更加重要，其中可视化后处理【1】是一个非常关键的领域。

近年来，无网格法【2】发展迅速，但是由于其相对于广泛使用的有限元法等一系列科学计算方法而言，其发展时间相对较短，在很多方面的发展还不够完善，尤其是专门针对无网格法可视化后处理方面的研究工作。

有限元法主要处理小变形的固体力学问题，且由于基于网格，于是在有限元法的可视化后处理中【3】，主要显示模型的网格和表面信息、位移信息、应力等值线（面）等。有限体积法等计算流体力学（CFD）方法，主要基于欧拉网格计算，其可视化后处理【4】主要针对标量场的等值线（面）显示，矢量场的点图标、矢量线（面）显示等功能实现。针对这一类的仿真计算方法，市面上已经有不少商业后处理软件，如Tecplot；很多商业计算软件如Abaqus，Ansys，FLUENT等也都带有后处理模块。

然而，无网格法的可视化和以上的几类方法的情况有所不同。首先，无网格法多数是基于粒子表示的数据，比较典型的代表是光滑质点流体动力学法（SPH）【5】和物质点法（MPM）【6】，于是传统后处理中常用的等值线（面）显示等在网格和表面上处理的方式便不方便被无网格法使用。虽然有如Delaunay三角化【7】等由粒子生成非结构网格的算法可以将传统方法使用在无网格法数据上，但是鉴于无网格法本身的特点，仍有大量的情况不适合使用该方法。其一，无网格法经常被用来模拟冲击、爆炸等大变形物理过程，经常伴随有大量高度飞溅的粒子，如爆炸烟雾、侵彻后的碎片云等，这些物质并不适合构造网格或表面。其二，无网格法经常用来模拟液固耦合、相变、化学反应等多种不同状态物质的混合物理过程，使用传统的等值线（面）等表示方法并对不同状态物质以颜色区分这种形式，很多场合下也并不能够使人对仿真结果有一个直观的认识。于是，现在大多数情况下人们还是采取了直接在原始粒子数据上表示计算结果的方式，如使用Paraview等商业软件。如图【1-1】所示，

左为导弹侵彻混凝土墙的照片效果，右为使用MPM仿真的此类问题并用Paraview直接显示粒子信息。右图的效果很大程度上让人无法接受，如果实现左图的真实效果，会好很多。

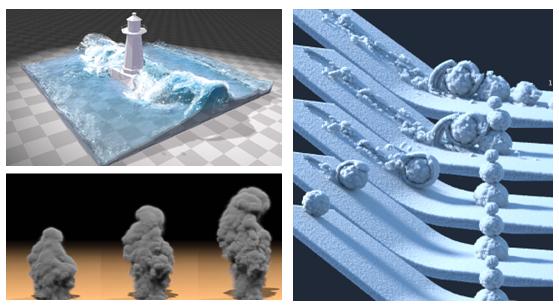
由此可见，对该类问题的可视化，采用真实感渲染的后处理方式是最能够直观、有效地表现物理过程的方法之一，而且其另一个优点是方便和实现效果对比。

清华大学航天航空学院计算动力学研究室开发的MPM仿真计算软件PeneBlast就是使用Paraview进行后处理的，其效果不能令人满意，于是这促成了本课题，我通过本课题的研究，开发了一套基于真实感渲染的无网格法可视化后处理系统，并将其加入到了PeneBlast中。

* 1. 基于物理仿真的真实感图形渲染工作介绍

虽然专门针对无网格法的通用真实感渲染系统非常少见，但是在图形学界，针对某一类具体自然现象而使用合适的仿真计算方法实现的真实感图形渲染工作还是比较多的，其中包括近些年非常火爆的基于SPH的流体仿真，基于各类CFD算法的烟雾、火焰和云的仿真，以及其它一些使用仿真计算生成的真实感效果，比较典型的是迪斯尼于2013年提出的使用物质点法模拟的雪的效果【8】

图【1-1】



其中由于液体和火、烟雾等介质体的渲染效果和无网格法主要适用物体最为相关，所以这里着重介绍这两方面的工作。

* + 1. 基于粒子的液体效果仿真

由于自然界中的液体效果大多是自由表面的，所以相比于有限体积法等一般CDF算法，SPH法在处理这类问题时有其独特的优势。也正因为如此，使用SPH法仿真和渲染时也需要关注如下几个方面的问题：邻近粒子搜索、粒子表示的表面追踪、特殊细节（如泡沫、水花以及空气泡等）的渲染。

1.邻近粒子搜索

SPH在计算过程中以及随后的渲染过程中都需要用到邻近粒子搜索，而为了防止蛮力搜索方式极度耗费时间的缺点，一般都会引入各种方便构造和查找的空间结构来加速搜索。比较普遍采用的结构有k-d树、八叉树等层次结构【9-11】和空间规则网格这两类。K-d树等层次结构的优点是其不受空间范围的限制，而其数据的组织时间复杂度为，查找复杂度为，相比于空间规则网格的和复杂度，在粒子数非常庞大时，其效率上的不足就会比较明显，且由于大多数仿真的流体都不会过于分散，空间范围的因素可以不必过多考虑，所以空间规则网格的方法被更多地采用。

规则网格结构通常是将空间划分为规则的立方体单元，然后使每个粒子都和一个单元对应，搜索邻近粒子时只需要直接查找目标位置及其附近的若干单元中的粒子。但是，该方法在并行化过程中，虽然查找过程比较方便实现，但是在构造网格的过程中可能会出现写冲突的情况，于是为了克服此问题，人们提出了索引排序【12】及Z索引排序【13】等算法。另一种高效且普遍采用的算法是空间哈希算法【14】，该算法通过将空间点坐标通过特定的散列函数映射到一个预先设定好长度的哈希表中以实现线性的插入及查找工作。一个比较普遍的散列函数如下：



其中m为哈希表长度，，，为三个大质数，【14】中取73856093, 19349663和83492791

2.粒子表示的表面重构

由于粒子数据无法直接用于渲染得到液体表面，于是我们需要用这些粒子来重构液体表面。这部分工作的主要挑战是如何高效地构造足够光滑的表面，而又能够捕捉到如液滴、液体薄膜及细丝等细节。重构的主要算法有基于3D标量场、显式法及屏幕空间渲染这几大类。

对于3D标量场类的算法，不同于欧拉网格的level-set方法，基于粒子的方法通常为每个粒子定义一个核函数，通过所有粒子核函数的叠加来定义一个密度标量场，然后通过MarchingCubes【】等算法提取该标量场的等值面来构造表面。然而这类方法的精度以及效率很大程度上取决于MarchingCubes算法中网格的尺寸，一般来说，为了捕捉到足够的细节，比较合理的网格尺寸为粒子半径大小【15】。这方面最早的工作为【16】中提到的元球算法，该算法是在每个粒子的中心位置上放置一个高斯分布的各项同性密度场核，于是其缺点就是容易造成构造出的表面凹凸不平。为了平滑考虑，一种广泛采用的算法是通过搜索一个粒子周围较大半径（通常为4倍粒子半径）的粒子，然后求它们的加权平均位置的算法来平滑表面粒子的位置【17】。以上方法都属于各向同性核方法，为了获得更加平滑的效果以及更好地捕捉液体薄膜等细节，人们又提出了基于各向异性核的方法，这部分工作详见第二章的分析。

对于显式算法，根据标量场构造表面多边形面片的过程可以免去，而可以通过直接计算光线和等值面的交点方法显式的渲染表面，交点计算可以使用元球或者距离场。另外还可以通过点泼洒算法【18】来处理各向同性核或各向异性核实现渲染。

以上算法实现起来效率一般较低，无法达到实时效果，而屏幕空间的重构算法由于在二维平面上操作，故而可以实现更高的效率。这类算法的一般做法是在屏幕空间上由3D点云生成一张深度图，并对其进行平滑操作。在渲染时，一类方法是使用MarchingSquares算法构造表面网格【19】，另一类是直接对深度图进行渲染【20】。

* + 1. 火、烟雾等介质体的仿真和渲染

这类自然现象属于典型的流体问题，其仿真多数采用基于欧拉网格的CFD算法，其中使用比较多的是半拉格朗日法【Fedkiw】。然而此类流体通常带有比较复杂的流动效果，如涡和湍流效果，而使用半拉格朗日法容易导致比较高的数值耗散，不易保持这些流动细节，于是很多基于粒子的拉格朗日方法也被用在了仿真这类流体上，其中比较主要的工作是针对涡保持的涡量法，包括涡量粒子法【21】，以及针对其涡扩散问题的改进方法涡量片法【22】及涡量线法【23】。

这类介质的渲染主要有两类算法：基于纹理的算法和基于体渲染的算法。基于纹理的算法是使用艺术制作或预先计算好的纹理图片，将其映射到粒子系统【3-1】或各类简单几何结构上【24-26】，然后使用透射率融合及颜色梯度技术等方法来处理边界以生成最终图形。该方法的最大优点是其效率非常高，被广泛采用于传统渲染管线。但是其缺点也非常显著，由于此方法并非基于物理真实的光照效果，所以其渲染图形的真实感通常较差。

体渲染一般使用光线跟踪或光线投射算法进行渲染，并且能通过光子映射等算法实现全局照明，虽然由于该算法需要将场变量或粒子数据离散到空间网格上以及计算全局光照导致的巨大耗时，但是其渲染质量也是非体渲染类算法不能相比的，所以目前在火、烟雾、云等真实感渲染中，体渲染还是主流算法。

* 1. 真实感渲染算法介绍

真实感渲染（Photorealistic Rendering）是指基于光照的物理规律，通过计算机模拟来生成与现实世界中所见的景物光影效果基本一致的图形。真实感渲染得到的图形一般做不到和真实世界完全相同，总会有误差存在，而根据误差的类型，我们将渲染算法分为有偏渲染和无偏渲染。有偏渲染指的是算法所采用的物理模型和近似给渲染结果带来了系统误差，虽然可以通过增加采样点等方式来使结果逐步收敛到正确结果，但并不能做到一致收敛，即同一套算法在针对不同算例时渲染结果可能会存在偏差。而无偏算法却可以保证计算稳定地收敛到正确的结果上。

典型的无偏算法包括传统光线跟踪算法、双向光线跟踪算法【27】以及Metropolis光线传播算法【28】等，这些方法都是基于Kajiya等在1986年提出的渲染积分方程【29】基础上发展而来的光线跟踪类算法。这类算法虽然能精确收敛到正确结果，但是其缺点是其计算量庞大，非常费时。其中传统光线跟踪算法的思路是根据光路可逆原理，从视点发出光线并计算光线与场景的交点，然后根据交点处的局部光照及该处反射光和折射光和场景其它位置的交点处对该点的光照贡献的总和来计算每个像素的光照。而传统光线跟踪处理间接光照非常困难，于是出现了双向光线跟踪，该算法从视点和光源均发射光线和场景中的物体求交点，并据此快速寻找视点到光源的光线通路，可以提高间接光照的计算速度，然而其速度仍不够令人满意。于是Veach等人提出了Metropolis采样算法用于光线跟踪，该算法使用基于马尔科夫链的采样序列启发式地找到贡献较高的光线进行渲染，大大提高了渲染速度和效果。

典型的有偏算法主要是光子映射算法【3-Jensen】。该算法将传统的光线跟踪算法分为两步：第一步为从光源出发向场景发射大量带有能量的光子，并在场景中追踪，将对物体照明有贡献的光子存储起来；第二步为从视点发出追踪光线，当光线和场景相交时，不是使用传统局部光照模型计算该点光照，而是搜索交点附近的光子，并根据光子的能量来计算该点的光照。该算法可以以较快的速度实现全局照明，但由于它是有偏渲染，结果存在偏差，需要增加光子数量来使渲染逐渐收敛到正确结果，为了克服这个问题，后来又相继出现了渐进光子映射法（PPM）【30】及随机渐进光子映射法（SPPM）【31】等改进算法。

* 1. 本文工作

如前所述，虽然对于某种特定类型的物理现象有很多利用了包括无网格法在内的仿真算法去进行真实感渲染，但是鲜有工作专门针对无网格法的计算数据研究统一的真实感渲染系统。这其中最主要的问题是，无网格法的计算结果涵盖各种类型的仿真对象，而很多时候一个算例中的粒子数据同时包含若干种对象类型，如液固耦合、侵彻、爆炸等情形。而无网格法处理的最多的对象也就是这些类型。于是本文的目标便为实现这样一个统一的框架，可以在用户读入仿真计算数据结果并设置简单的用户参数后，自动识别数据中不同类型的对象，并渲染为真实感图形。由于无网格法计算类型涵盖范围太广，本文不可能将所有类型都涉及到，于是只选取其中最重要的两个效果来实现该框架，分别为粒子表示的连续固体或液体的渲染以及冲击爆炸等过程形成的碎片云或烟雾效果。

本文第二章针对粒子表示的连续固体或液体的渲染中涉及到的最重要的算法——表面重构算法作了简要的介绍，并提出了适合于本系统的一些算法上的改进，使系统具有自适应性和鲁棒性，并提高了渲染效率。

第三章针对碎片云和烟雾效果选定了合适的渲染算法，并提出了针对全局光照渲染精度的算法改进以及运动模糊效果的实现算法。

第四章提出了一种能够实现自动区分渲染对象类型的算法，并介绍了实现整个渲染系统的程序框架。

第五章介绍了将该后处理系统集成到计算软件PeneBlast中的方法。

第六章总结了全文工作。

1. Nielson G M, Shriver B. Visualization in scientific computing[M]. IEEE Computer Society Press, 1990.
2. 张雄,宋康祖,陆明万. 无网格法研究进展及其应用[J]. 计算力学学报,2003,06:730-742.

[3]范彦斌,杨彭基. 有限元分析计算结果的计算机图形可视化显示[J]. 计算机辅助设计与图形学学报,1995,01:11-16.

[4]雷勇,魏涛,柳共青. CFD可视化研究与进展[J]. 航空计算技术,1999,01:8-11+19.

【5】Gingold R A, Monaghan J J. Smoothed particle hydrodynamics: theory and application to non-spherical stars[J]. Monthly notices of the royal astronomical society, 1977, 181(3): 375-389.

【6】Sulsky D, Chen Z, Schreyer H L. A particle method for history-dependent materials[J]. Computer Methods in Applied Mechanics and Engineering, 1994, 118(1): 179-196.

[7] Watson DF. Computing the n-dimensional Delaunay tessellations with application to Voronoi polytopes[J].

The Computer J., 1981, 24(2):167-172.

【8】Stomakhin A, Schroeder C, Chai L, et al. A material point method for snow simulation[J]. ACM Transactions on Graphics (TOG), 2013, 32(4): 102.

【9】KEISER R., ADAMS B., DUTRÉ P., GUIBAS L.,

PAULY M.: Multiresolution Particle-Based Fluids. Tech. rep.,

ETH Zurich, 2006.

【10】KEISER R.: Meshless Lagrangian Methods for Physics-

Based Animation of Solids and Fluids. PhD thesis, ETH Zürich,

2006.

【11】ADAMS B., PAULY M., KEISER R., GUIBAS L.:

Adaptively sampled particle fluids. In ACM Transactions on

Graphics (Proceedings SIGGRAPH) (2007), vol. 26, pp. 48:1–

48:7.

【12】PURCELL T. J., DONNER C., CAMMARANO M.,

JENSEN H. W., HANRAHAN P.: Photon mapping on programmable

graphics hardware. In Proceedings of the ACM SIGGRAPH/

EUROGRAPHICS Conference on Graphics Hardware

(2003), pp. 41–50.

【13】GOSWAMI P., SCHLEGEL P., SOLENTHALER B., PAJAROLA

R.: Interactive SPH simulation and rendering on the

GPU. In Proceedings of the ACM SIGGRAPH/Eurographics

Symposium on Computer Animation (2010), pp. 55–64.

【14】TESCHNER M., HEIDELBERGER B., MÜLLER M.,

POMERANETS D., GROSS M.: Optimized spatial hashing for

collision detection of deformable objects. In Proceedings of Vision,

Modeling, Visualization (VMV) (2003), pp. 47–54.

【15】AKINCI G., IHMSEN M., AKINCI N., TESCHNER M.:

Parallel surface reconstruction for particle-based fluids. Computer

Graphics Forum 32, 1 (2012), 99–112.

【16】BLINN J.: A generalization of algebraic surface drawing.

ACM Transactions on Graphics 1, 3 (1982), 235–256.

【17】ZHU Y., BRIDSON R.: Animating sand as a fluid. ACM

Transactions on Graphics (Proceedings SIGGRAPH) 24 (2005),

965–972.

【18】ZWICKER M., PFISTER H., VAN BAAR J., GROSS

M.: Surface splatting. In SIGGRAPH ’01: Proceedings of the

28th annual conference on Computer graphics and interactive

techniques (New York, NY, USA, 2001), ACM Press, pp. 371–

378.

【19】MÜLLER M., HEIDELBERGER B., HENNIX M.,

RATCLIFF J.: Position based dynamics. J. Vis. Commun. 18,

1. (2007), 109–118

【20】GREEN S.: Screen space fluid rendering for games.

"http://developer.download.nvidia.com/

presentations/2010/gdc/Direct3D\_Effects.

pdf", 2010.

【21】Selle A, Rasmussen N, Fedkiw R. A vortex particle method for smoke, water and explosions[C]//ACM Transactions on Graphics (TOG). ACM, 2005, 24(3): 910-914.

【22】Pfaff T, Thuerey N, Gross M. Lagrangian vortex sheets for animating fluids[J]. ACM Transactions on Graphics (TOG), 2012, 31(4): 112.

【23】Angelidis A, Neyret F. Simulation of smoke based on vortex filament primitives[C]//Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation. ACM, 2005: 87-96.

【24】Lee H, Kim L, Meyer M, Desbrun M (2000) Meshes on fire. In: EuroGraphics 2000 workshop

on animation

【25】Roberts I (2001) Realistic modeling of flame. Bachelor thesis, Univ. of Bristol

【26】Zhao Y, Wei X, Fan Z, Kaufman A, Qin H (2003) Voxels on fire. In: Proceedings of IEEE

visualization, pp 271–278

【27】Veach E, Guibas L. Bidirectional estimators for light transport[M]//Photorealistic Rendering Techniques. Springer Berlin Heidelberg, 1995: 145-167.

【28】Veach E, Guibas L J. Metropolis light transport[C]//Proceedings of the 24th annual conference on Computer graphics and interactive techniques. ACM Press/Addison-Wesley Publishing Co., 1997: 65-76.

【29】Kajiya J T. The rendering equation[C]//ACM Siggraph Computer Graphics. ACM, 1986, 20(4): 143-150.

【30】Hachisuka T, Ogaki S, Jensen H W. Progressive photon mapping[C]//ACM Transactions on Graphics (TOG). ACM, 2008, 27(5): 130.

【31】Hachisuka T, Jensen H W. Stochastic progressive photon mapping[C]//ACM Transactions on Graphics (TOG). ACM, 2009, 28(5): 141.