



论文

物理场的团簇识别与分析技术

张广财*, 许爱国, 卢果, 莫则尧

北京应用物理与计算数学研究所国家级计算物理重点实验室, 北京 100088

* E-mail: zhang_guangcai@iapcm.ac.cn

收稿日期: 2009-11-15; 接受日期: 2010-03-01

国家自然科学基金(批准号:10102010,10775018)、中国工程物理研究院发展基金(编号: 2009A0102005, 2009B0101012)和实验室基金资助项目

摘要 复杂构型和物理场的理解与规律提取一直是个难点. 该文提供了一种团簇识别和分析技术, 用于各类实验或模拟研究中的数据处理和分析. 主要算法如下: 采用离散点构造面元或体元以分割粒子所在空间; 根据物理场的特征合并相同或相似性质的元, 从而获得反映物理场特征的团簇结构. 在算法中, 对空间物体(点、线、面、体、团簇等)统一采用分级树数据管理结构, 实现了对空间物体的两种快速搜索, 搜索算法复杂性约化到. 空间多级树的建立方法及快速搜索算法是通用的, 不依赖于空间维数, 可以方便地推广应用于其他领域. 作为应用实例, 我们将该方法应用到二维和三维随机数据处理, 验证了该方法的准确性和可靠性.

关键词 空间多级树, 快速搜索, 复杂构型, 动态物理场, 团簇识别

PACS: 05.90.+m, 07.05.Kf, 07.05.Rm

复杂构型和动态物理场在武器物理、天体物理、等离子体物理、材料物理等领域广泛存在. 这些结构和场的特性与演化模式决定着相应物理系统的各种性能. 例如, 界面不稳定性极大地限制着惯性约束聚变(Inertial Confinement Fusion, ICF)装置的设计^[1]; 冲击波和射流等现象^[2]在高能物理是常见的运动模式; 云团、星云的分布^[3-5]等是天体物理非常关注的问题; 在强激光与等离子体相互作用过程中存在着成团、成丝^[6]等现象; 金属材料塑性变形过程中位错聚集形成的位错带结构^[7]决定着材料的软化性. 这些结构特征也是理解多尺度物理过程的关键: 小尺度的物理规律决定较大尺度稳定结构的生长、变化及结构间的相互作用; 稳定结构演化规律的描述为更大尺度上的物理建模提供本构关系. 由于这些结构的特征缺乏周期性、对称性、空间均匀性和明显的关联

性等, 在模拟研究中对它们的识别和分析一直是个薄弱环节, 多年来进展缓慢.

现有的复杂构型与动态物理场特征分析方法主要有均匀物理场背景下小扰动场的线性分析、简单空间分布物理场(如一维不定常流)的特征分析(如特征线方法、冲击波结构和射流结构等)以及对粒子模拟结果的物理图像观察等. 由于这些方法对特征物理区的大小、形状、拓扑结构、流通量、物理量积分等特征量缺乏定量计算, 所以特征结构区域的演变过程得不到有效追踪, 难以把握特征区域之间, 特征区与其背景间的消长、转化关系.

特征分析方法的难点集中在两点: (1) 如何定义和描述特征区, (2) 如何界定特征区的空间结构. 前者涉及具体物理系统的演化方式(基本运动方程); 后者涉及离散点的几何结构(区域的体和边界). 近年来

数据挖掘技术中聚团分析技术^[8,9]在目标识别、规律总结方面有很好的应用,其主要关心的是数据集中的目标归类问题.相对来说,物理系统中更关心团簇的区域结构性.特征区域构造可以归结为空间几何结构的构造,其中的关键问题是离散点的连接方式. Delaunay 网格^[10,11]具有优良的空间近邻关系,本工作中,采用 Delaunay 三角形或四面体作为几何单元.

数据结构设计和快速搜索算法^[12,13]是当今计算机软件工程中的核心问题.用树结构管理空间分散数据在内存和快速搜索方面有明显优势.在天体演化及星系形成研究领域,空间多级树广泛用来管理空间的质量颗粒^[14,15],实现颗粒受力的快速计算和星系质量分布分析.

在本工作中,我们采用空间多级“树”管理 n 维空间中的物体,提出了适应给定搜索条件的两种快速搜索算法,在二维和三维空间上实现了 Delaunay 空间划分和团簇构造.

1 空间多级树管理结构

空间多级树已经成功应用于空间数据点的管理和索引^[16],但是,还没有应用于更加复杂的空间物体,例如线、面、体、团簇等.本文采用空间多级树管理空间位置、形状和尺寸各异的物体.基本思路如下:对于系统所在的 n 维空间,设计尺寸能够包含此系统的 n 维方体区域;将此方体区域每个维度方向一分为二,形成 2^n 个子方体,仅保留其内部含有物体的子方体;对每一个小方体区域继续分解直到所需要的分辨率;将相应的物体(点、线、面、体)依照其位置和尺寸,放入合适的方体中.将各个方体根据空间隶属关系连接成“空间多级树”.每个方体区域称为“枝”;其包含的子方体区域称为“子枝”;反之,相对于“子枝”,上述“枝”称为“干”.最大方体称为“根”.图 1 和 2 分别是二维、三维离散点的多级树管理结构示意图.由于无法预知每个“枝”中包含的“子枝”数目,同属于一个“枝”的“子枝”采用链表相连;同样,属于同一“枝”的所有“物体”间也通过链表相连.

由于应用过程中空间物体可能随时增减,空间多级树采用动态构造方法.由一个物体建立“树”的方法是:(1) 给定最小分辨率(即最小方体的边长) σ ;(2) 以物体的几何中心为方体的中心,检查边长为 σ 的方体能否将该物体包住.如果能,就以该方体为

“枝”,将此物体放入此枝中;若不能,将边长持续加倍,直到能够放下此物体为止,这时产生一个“枝”,将物体放入.到这里,已建立起一棵只包含一个“枝”的“树”,“树”上只包含了一个“物体”.图 3 以二维空间中的一个三角形物体为例,给出了原始树的构造过程.

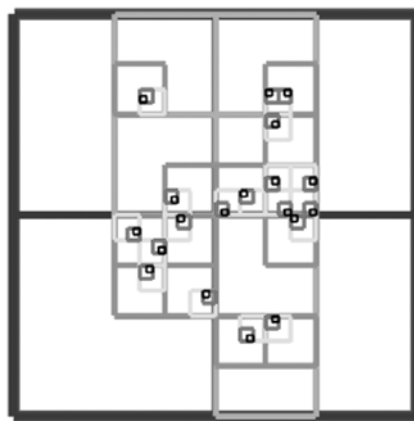


图 1 二维离散点的多级树结构的管理区域

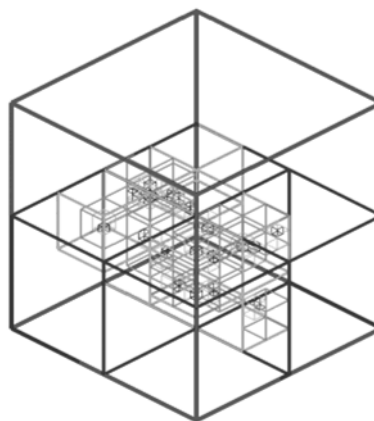


图 2 三维离散点的多级树结构的管理区域

假设空间多级树上已经放置了一些物体,用“A”代表.在树上增加一个物体 B 的算法为:(1) 建立新根.检查原树的根能否包含物体 B.若不能,则建立新根:计算物体 B 的中心(相对原根坐标系)所处的象限,并将位于该象限的原根(方体)顶点设为新根中心.这样,新根为原根的干,原根降级为枝,持续这个过程,直到新根能够包含物体 B 为止.(2) 放置物体 B.从新根出发,计算物体 B 的中心相对当前枝所处的象限,检查此象限对应的子枝是否能包含物体 B.若

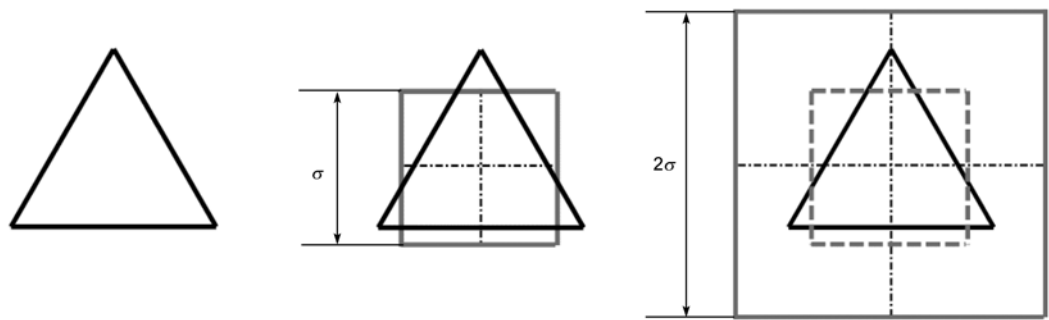


图 3 原始树的构造过程示意图

不能, 将物体 B 放在当前枝; 如果能, 把该子枝作为当前枝(如果该子枝不存在, 建立此枝), 继续这个过程, 直到形成一个能够存放物体 B 的子枝为止, 并将物体 B 放入子枝. 图 4 是增加一个三角形物体的过程示意图.

从树上删除一个物体的算法为: (1) 根据连接, 找到存放这个物体的枝. 从该枝所对应的物体链表中删除此物体. (2) 当此枝不再含有物体且没有子枝时, 删除此枝, 将它的干作为当前枝. 持续这个过程, 直到删除所有无用的枝为止.

空间多级树动态构造算法中, 除了增加一个枝的干或增加一个枝的子枝(需要计算新枝的大小、位置)外, 其余操作与具体空间维数没有关系. 可以用一个程序统一实现任何维数空间的树的构造. 空间多级树所需的内存量与空间维数没有关系, 只与物体数目 N 及分辨率有关, 约为 $kN \ln N$, 在维数比较高或者空间物体分布比较分散的情况下, 比空间背景网格的管理方式节约内存. 另外, 本文的空间多级树

管理模式中, 树的构造是动态的. 随着物体的增加或删除, 系统的尺寸可以动态地增大或减小, 这是背景网格管理模式难做到的.

2 基于空间多级树的快速搜索法

在处理空间物体的邻域关系及构造空间几何时, 需要对满足一定条件的物体进行快速搜索. 遍历性搜索计算量是 N , 在处理较多物体时代价太高, 因而需要发展快速搜索办法. 根据空间多级树的层次结构, 可以找到计算量为 $\log N$ 的快速搜索方法. 基本思想是: 不直接查找物体, 而是检查树枝, 及时排除不含待找物体的树枝, 快速地将搜索限制在很小的范围. 根据应用计算需求, 我们提出两种快速搜索: 条件搜索和极值搜索. 条件搜索是指搜索空间中满足一定条件的物体的集合(比如, 处在某一给定区域中的所有物体的集合). 极值搜索是指搜索空间中满足一定极值条件的一个物体(比如, 查找距离某固定

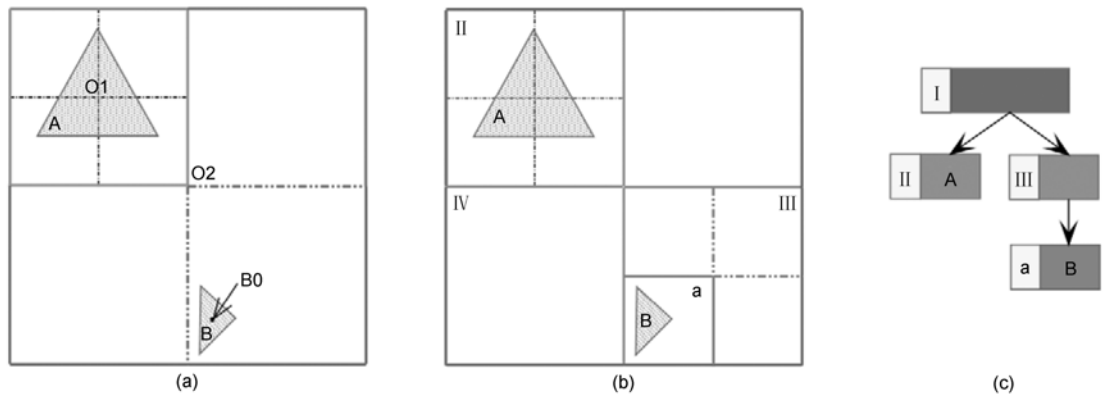


图 4 树上增加物体过程的示意图

(a) 生成新根; (b) 放置物体; (c) 与(b)对应的空间多级树

点最近的物体).

2.1 条件搜索

条件搜索的思路是: 判断一个枝的区域是否含有满足条件的物体, 如果不包含就不对枝内物体进行搜索. 比如, 要搜索出一个特定圆区域内所有的物体, 需要评估一个枝的区域是否与它相交. 这样, 搜索只限于与这个圆的区域有交叠的枝所包含的物体.

条件搜索的算法使用堆栈结构实现, 具体算法:

(1) 将根入栈; (2) 从栈中取出一枝 b , 检查它所挂的物体是否满足条件, 将符合条件的物体挑出来. (3) 检查 b 的每个分枝, 将满足条件的枝入栈; (4) 重复(2)和(3), 直到栈空为止.

图 5 给出了管理平面三角形物体的空间结构划分和圆形区域, 图 6 是与图 5 对应的空间多级树示意图(其中“/”代表根). 找到在圆内部的三角形的快速搜索过程如图 7 所示: 第一步, 将根放入栈中; 第二步, 从栈中将枝取出, 找出此树枝上在圆内的物体; 检查它的子枝, 在本例中, 发现它的 4 个枝(I)~(IV)都与这个圆有交叠, 于是将这些枝都堆栈. 接着重复第二步, 取出枝(IV), 找出此树枝上在圆内的物体; 检查它的子枝, 在本例中它的两个枝都在圆外. 继续重复第二步, 直到栈中再也没有枝为止, 最终找到物

体 A 和物体 B 在圆内. 这样, 在各级树枝层次上避免搜索很多树枝中的物体, 使得搜索范围迅速减小. 这

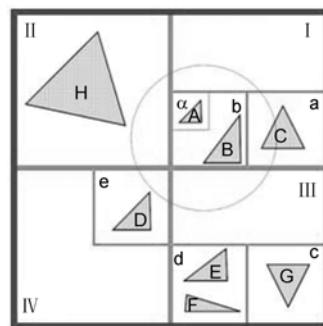


图 5 平面三角形分布和空间划分

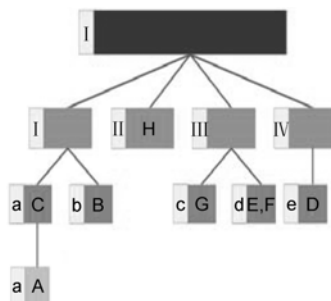


图 6 与图 5 对应的空间多级树

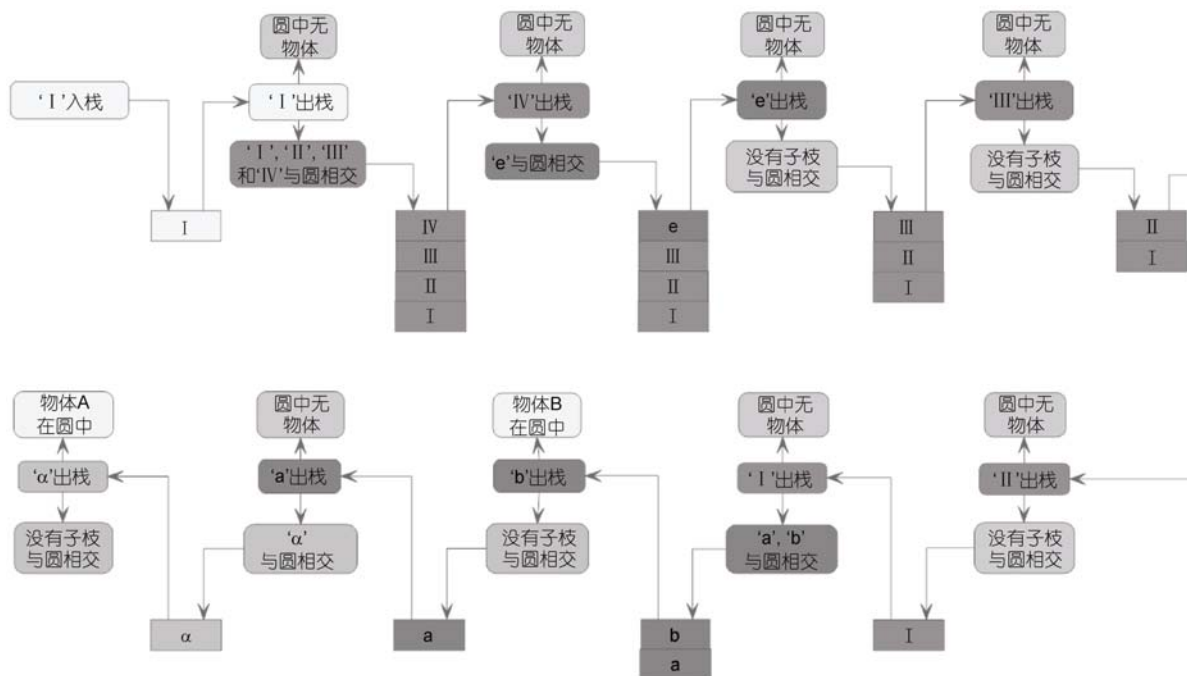


图 7 快速搜索圆形区域中物体的过程图

样搜索计算量仅与多级树的总级别成正比, 即正比与 $\ln N$.

条件搜索在抽象层次上通过提供条件函数和判别函数来实现. 条件函数用于界定一个物体是否满足条件, 判别函数用于识别一个枝的区域中有可能包含满足条件的物体. 搜索算法的有效性体现在判别函数中. 判别函数识别越准确, 需要搜索的枝就越少. 如果判别函数恒真, 那么该方法退回到遍历性搜索. 针对方体区域的判别函数有时很复杂, 可以将枝的方体区域扩展为外接球区域, 针对球形区域的构造判别函数可能很简单.

2.2 极值搜索

极小值搜索算法的思路是: 通过比较枝的值域, 避免对一些枝中物体的搜索. 为方便描述, 定义几个概念. (1) 枝的值域: 包含在该枝内所有物体所对应的函数值的范围, 对应下限和上限两个值. (2) 值域枝: 枝的值域和该枝合在一起构成的新数据结构. (3) 候选值域枝: 可能包含极小值物体的值域枝. 在极小值搜索算法中, 必须保存许多候选值域枝, 而且可能动态地、任意次序地增删候选值域枝, 所以采用候选值域枝链表来存放候选值域枝. 为了加快搜索速度, 将下限最小的候选值域枝放在链表头. 另外需要设置候选物体和候选值来保存在搜索过程中当前的最小值物体及其函数值.

极小值搜索算法: (1) 将树根及其值域合成值域枝, 加入候选值域枝链表 L 中. 设置候选值为无穷大, 候选物体为空. (2) 从候选值域枝链表 L 中取出值域枝 B , 检查它所挂的物体的函数值 V . 如果比候选值还小, 候选值设置为 V , 将候选物体设置为此物体;

删除 L 中下限大于 V 的值域枝. (3) 对 B 的每个子枝构造值域枝 Z , 如果 Z 的下限比候选值大, 放弃 Z ; 如果 Z 的上限比 L 中某个值域枝 C 的下限还小, 删除 C 后面的所有值域枝; 如果 L 中某个值域枝的上限比 Z 的下限还小, 放弃 Z , 否则将 Z 按照其下限值正次序插入到链表 L 中. (4) 重复(2)和(3)步骤, 直到候选值域枝链表变空, 最终的候选物体即为满足极值条件的物体.

例如, 找出平面上距离一定点最近的点. 图 8 是平面点分布和空间划分. 图 9 是对应的空间多级树. 图 10 给出了搜索到距离 A 点最近点的流程.

极小值搜索在抽象层次上可以通过提供一个求值函数和一个值域评估函数来实现不同的最小值搜索. 求值函数计算物体的函数值, 值域评估函数估算枝的方体区域中物体的函数值范围. 最小值搜索效率取决于值域评估函数. 值域评估函数给出的值域范围越小, 搜索越快. 最差的值域评估函数给出的值域是下限为 $-\infty$, 上限为 $+\infty$, 搜索退化为遍历性搜索.

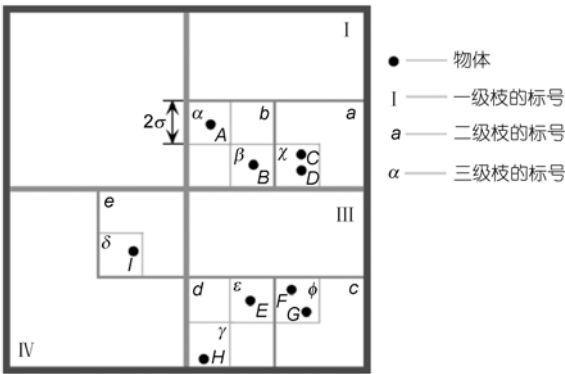


图 8 平面点分布和空间划分

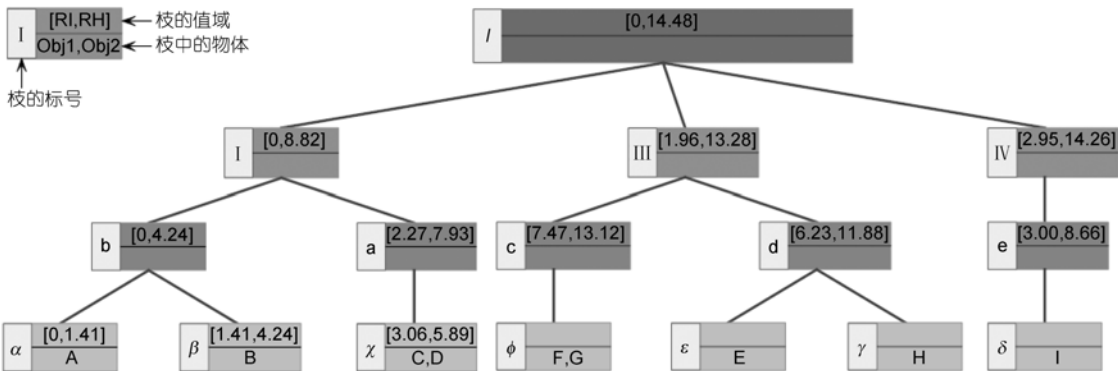


图 9 与图 8 对应的空间多级树

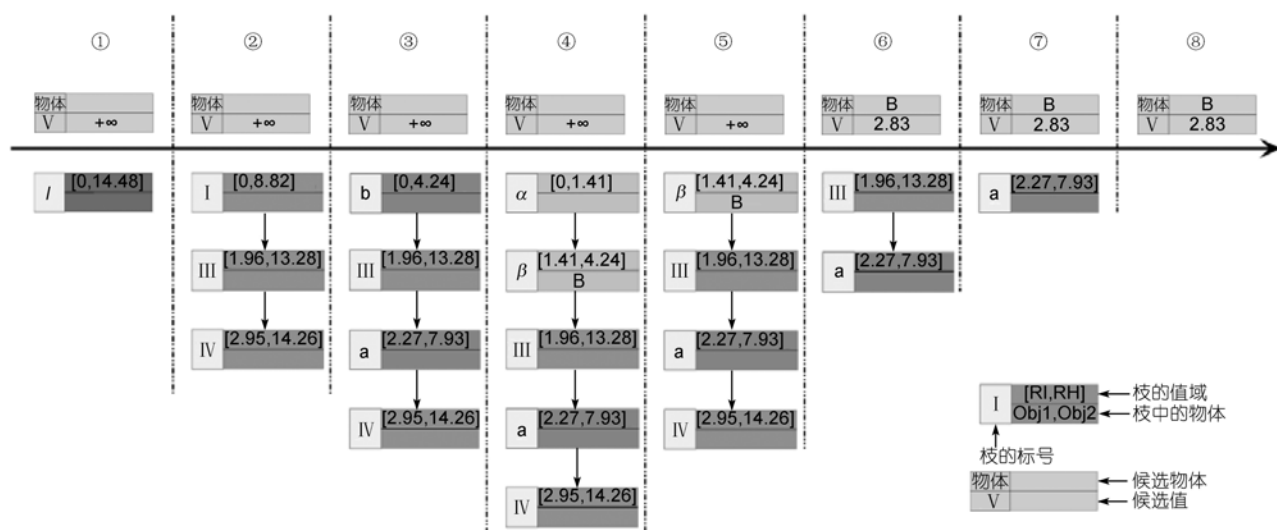


图 10 快速搜索距离 A 点最近点的过程图

对于较大数量的物体的搜索,应尽量让值域评估函数给出的值域范围更准确以减少搜索的物体数目,但计算量较大的值域评估函数也会影响搜索效率,因此实际的评估函数应尽可能简单以保证计算量小,同时给出尽可能准确范围.由于球区域比方体区域更容易计算,对于复杂的极值搜索,常用方体的外接球来评估枝的值域.

3 Delaunay 四面体和三角形的快速构造法

构造团簇之前,需要使用空间的离散点对空间进行有效剖分,我们选用 Delaunay 剖分构造离散点连接的几何结构.使用三维空间中的离散点构造 Delaunay 四面体及利用二维空间中的离散点构造 Delaunay 三角形都有很多计算方法,大多数算法的复杂性集中在搜索上.我们提出了一种算法,借助于多级树上的快速搜索,算法简单直观,而且特别适合推广到更高维空间中.

算法的思路是:向当前已经形成的 Delaunay 剖分结构中添加新点,同时重新划分新点附近网格.根据 Delaunay 划分的定义,如果新点在某个 Delaunay 单形的外接球之外,新点的加入对此 Delaunay 单形没有影响;反之,如果新点在哪个单形外接球之内,就会影响到这个单形,需要重新划分.具体地说,将其外接球包含新点的所有单形挑选出来,构造复形结构.新点与复形的每个单形面形成新的单形.

在三维空间中使用离散点构造 Delaunay 四面体的算法为:(1)生成足够大的四面体,保证所有离散点都处该四面体之内,并记录该四面体的球心和半径,将球接四面体加入空间多级管理树 t (球接四面体树)中;(2)取出一个离散点 P ,在球接四面体树 t 上搜索所有的其外接球包含点 P 的球接四面体,将其从 t 中摘除;并将这些摘除的四面体合在一起,形成外接四面体集合 Q ;(3)将 Q 中所有四面体的每个三角面加入到空间多级管理树 s (三角形外界面树)中,同时剔除出现两次的面(因为它们是内界面), s 树上的三角形就构成 Q 的外界面;(4)取出空间多级管理树 s 上的每个三角形,与 P 构造四面体,记算其外接球球心和半径,将球接四面体加入球接四面体树 t 上;(5)重复(2)~(4)步骤,直到用完所有点.球接四面体树 t 上的四面体就构成了 Delaunay 四面体划分.

算法实现过程中,涉及对包含某一固定点的球接四面体的搜索和外界面的搜索.这里使用了空间多级的快速搜索.对于二维情况,算法相同,只需将四面体换成三角形,三角形面换成边线.图 11 给出加入一个二维离散点和重新划分空间的过程.图 12 是采用这种办法计算的二维 20000 个随机分布的离散点构造出的 Delaunay 三角形划分.图 13 是采用这种办法计算球区域中 20000 个随机分布的离散点构造出的 Delaunay 四面体划分.

这个算法很容易推广到 n 维空间.只需要将四面体换为 n -单形,三角形换为 $(n-1)$ -单形.算法中需要

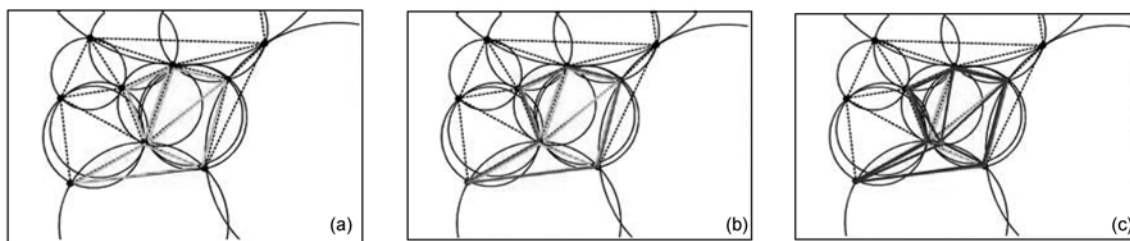
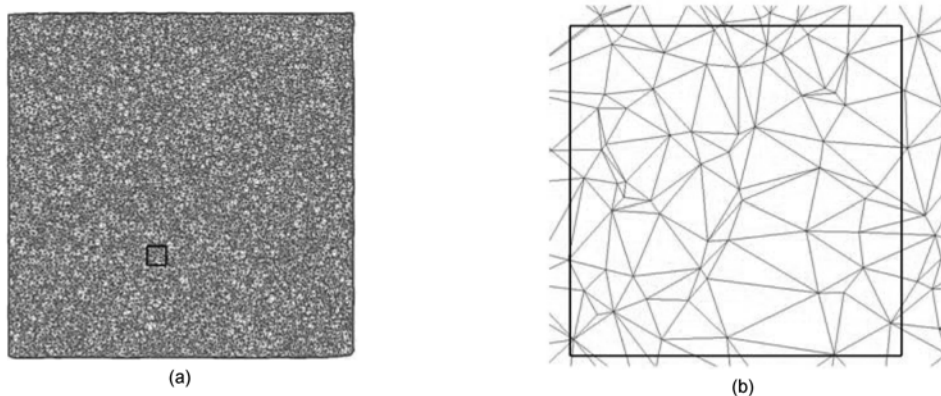


图 11 二维 Delaunay 划分中加入一个新点的三个步骤

(a) 查找外接圆包含新加入的点 p 的三角形; (b) 将这些三角形的内部线段去掉, 保留外界线段; (c) 将每一个外界线与 p 相连, 形成新的三角形

图 12 二维正方形 $[0,4] \times [0,4]$ 区域中随机分布的离散点构造的 Delaunay 划分

(b) 是 (a) 中黑色小矩形区域的局部放大图

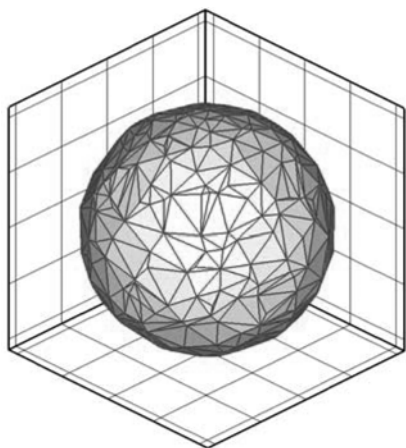


图 13 三维球型区域中随机分布的离散点构造的 Delaunay 划分

用 n 维空间上 $n+1$ 个点 $\{r_1, r_2, \dots, r_{n+1}\}$ 构造外接球. 球心坐标 c 的计算公式为 $c_i = A_i / B$, 球半径为 $\sqrt{(c - r_1)^2}$.

其中

$$A_i = \begin{vmatrix} 1 & -2r_1 & r_1^2 \\ 1 & -2r_2 & r_2^2 \\ \dots & \dots & \dots \\ 1 & -2r_{n+1} & r_{n+1}^2 \\ 0 & e_i & 0 \end{vmatrix},$$

$$B = \begin{vmatrix} 1 & -2r_1 & r_1^2 \\ 1 & -2r_2 & r_2^2 \\ \dots & \dots & \dots \\ 1 & -2r_{n+1} & r_{n+1}^2 \\ 0 & 0 & 1 \end{vmatrix},$$

e_i 为第 i 个方向的单位矢量.

4 团簇构造算法及其分析方法

对于空间离散点, 并不存在严格的团簇结构. 如果将离散点视为以其为中心的物体(比如分子球、晶格或网格等), 这些物体相连能够形成团簇. 这些假

定的物体的平均大小称为离散点构造团簇的分辨率. 团簇构造方法很简单: 将空间中距离小于分辨率距离的所有点连在一起, 构成一个团簇.

使用离散点构造出空间的 Delaunay 划分后, 除去距离大于分辨率的连线, 剩下的空间结构会具有各种维数. 根据连通性, 相互之间不连通的结构可以分解为不同的团簇. 每个团簇中又会有各种维数的结构, 比如在三维空间中有相互共边的三角形、三角形与四面体连接形成的结构, 等等. 在物理问题中, 对系统起主要作用的是测度较大的结构, 因此, 仅分析最高维数的结构团簇.

构造团簇的算法分三个步骤. 准备阶段: (1) 利用空间离散点构造 Delaunay 四面体, 相应的空间多级树为 t ; (2) 删除 t 上所有存在大于分辨率边长的四面体; (3) 设置团簇树 c 和三角面树 i 为空; 构造单个团簇: (3) 如果 t 上还有四面体 T , 从 t 上删除 T . 生成新团簇 C , 设置它的体树 $C \rightarrow t$ 和表面树 $C \rightarrow s$ 为空; 将 T 添加到 $C \rightarrow t$ 上, 将其每个三角形面添加到表面三角形树 i 上. (4) 从 i 上摘下一个三角形面 S , 在四面体树 t 上寻找含有这个面的四面体 Y . 如果能够找到, 将 Y 添加到树 $C \rightarrow t$ 上, 将 Y 的所有面添加到树 i 上, 添加时遵循相反面相消的原则; 否则, 将 S 添加到 $C \rightarrow s$ 上. (5) 继续过程(4), 直到 i 树为空. 构造所有团簇: (6) 将构造好的团簇 C 添加到团簇树 c 上, 继续构造单个团簇的过程, 将每个构造好的团簇添加到 c 上, 直到 t 上不再含有四面体.

向树 i 添加一个面 S 的算法是: 搜索树 i 上是否

含有 S 的反面, 若存在, 从树上删除它; 若不存在, 将 S 添加在树 i 上.

这样, 构造出的团簇全部放置在团簇树 c 上, 对于团簇树 c 上的每一个团簇 C , 所有四面体元放置在它的体树 $C \rightarrow t$ 上, 所有的表面三角形放置在表面树 $C \rightarrow s$ 上. 空间多级树除了方便查找外, 还提供了方便的数据管理结构. 图 14 和 15 分别是用二维和三维空间中随机点构造的团簇结构.

这个算法对 n -维空间的离散点也通用, 只需要将四面体换成 n -单形, 将表面三角形换成 $(n-1)$ -单形. 对于三维以上空间的情况, 由于近邻点数目和连通度增加, n -单形的数目随维数增长很快, 需要内存量也增长很快, 可以分区构造 Delaunay 单形. 算法技巧在于采用空间多级树主要枝干将空间中的点分区, 依次添加每个区域中的点. 在一个区中的点全部添加之后, 删除其外接球完全处于已经完成的区域且至少有一边长大于分辨距离的 n -单形.

5 结论与讨论

我们构造了一种新的适合任何维数空间物体的管理方法和快速搜索方法, 在此基础上构造了 Delaunay 单形结构, 然后形成了空间物体团簇的计算方法. 在二维、三维空间离散点的处理表明, 这种算法简单、直观.

空间多级管理树可以方便地管理不同空间位置、不同大小甚至是不同形状的物体. 可以对其进行快速搜索. 这种管理方法, 有非常多的应用, 比如目前

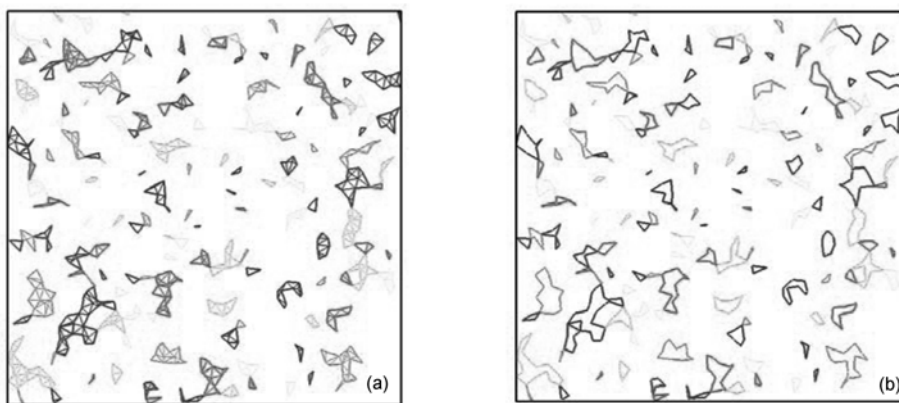


图 14 二维空间 $[0,1] \times [0,1]$ 区域中 1000 个随机离散点计算得到的团簇结构

(a) 团簇; (b) 相应的团簇边界

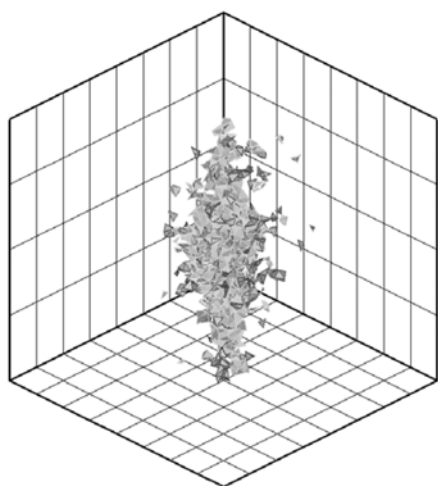


图 15 三维空间中 5000 个随机离散点计算得到的团簇结构

有限元方法研究中, 由于复杂形状的有限元搜索和相对位置计算复杂, 目前多限于简单形状, 单一尺寸的情况. 通过空间多级管理树, 可以方便地在不同形状、不同大小物体中搜索物体的相邻关系, 可以实现复杂形状、任意尺寸的有限元运动的模拟. 在复杂系统处理方面很有用.

空间多级管理树及以其为基础的两类快速搜索方法都建立在抽象的框架上. 对于不同维数空间物体的处理采用同样的办法, 因此可以扩展到其他方面. 通过物理量参数化, 可以将试验数据置于同一空间管理树中, 根据相似性进行快速搜索. 另外, 根据计算应用需要可以将物体同时置于几个管理树下, 相当于同时设置几个性质上的索引, 可以方便地实现不同性质上的快速搜索.

参考文献

- 1 Ament P. Effects of ionization gradients on Inertial-Confinement-Fusion capsule hydrodynamics stability. *Phys Rev Lett*, 2008, 101: 115004
- 2 de Vries P C, Hua M D, McDonald D C, et al. Scaling of rotation and momentum confinement in JET plasmas. *Nucl Fusion*, 2008, 48: 065006
- 3 Bowler B P, Waller W H, Megeath S T, et al. An infrared census of star formation in the horsehead nebula. *Astron J*, 2009, 137: 3685—3699
- 4 Hernquist L. Hierarchical N-body methods. *Comput Phys Commun*, 1988, 48: 107—115
- 5 Makino J. Vectorization of a treecode. *J Comput Phys*, 1990, 87: 148—160
- 6 Hidaka Y, Choi E M, Mastovsky I, et al. Observation of large arrays of plasma filaments in air breakdown by 1.5-MW 110-GHz gyrotron pulses. *Phys Rev Lett*, 2008, 100: 035003
- 7 Nogaret T, Rodney D, Fivel M, et al. Clear band formation simulated by dislocation dynamics: Role of helical turns and pile-ups. *J Nucl Mater*, 2008, 380: 22—29
- 8 Kotsiantis S B, Pintelas P E. Recent advances in clustering: A brief survey. *WSEAS Trans Inform Sci Appl*, 2004, 1: 73—81
- 9 Fan Y J, Iyigun C, Chaovalitwongse W A. Recent advances in mathematical programming for classification and cluster analysis. *CRM Proc Lecture Notes*, 2008, 45: 67—93
- 10 Chazelle B, Devillers O, Hurtado F, et al. Splitting a Delaunay triangulation in linear time. *Algorithmica*, 2002, 34: 39—46
- 11 Clarkson K L, Varadarajan K. Improved approximation algorithms for geometric set cover. *Discrete Comput Geom*, 2007, 37(1): 43—58
- 12 Black P E. Entry for data structure in *Dictionary of Algorithms and Data Structures*. U.S. National Institute of Standards and Technology. 15 December. 2004
- 13 Knuth D E. *The Art of Computer Programming*. Vol. 3: Sorting and Searching. Redwood City: Addison Wesley Longman Publishing Co, 1998
- 14 Barnes J, Hut P. A hierarchical $O(N \ln N)$ force-calculation algorithm. *Nature*, 1986, 324: 446—449
- 15 Pfalzner S, Gibbon P. *Many Body Tree Methods in Physics*. New York: Cambridge University Press, 1996
- 16 Nam B, Sussman A. A comparative study of spatial indexing techniques for multidimensional scientific dataset. In: *Proceedings of the 16th International Conference on Scientific and Statistical Database Management (SSDBM'04)*, June 21-23, 2004. 171