

基于海量地震数据的多分辨率扩展八叉树模型

张磊, 唐杰, 武港山

(南京大学计算机科学与技术系, 南京 210093)

摘要: 针对现有 PC 硬件条件的限制和地震数据的海量特征, 提出一种基于扩展八叉树的分块多层多分辨率模型。该模型保存了叶节点属性和中间节点属性, 实现多分辨率建模, 采用基于 Morton 码的数据块索引方式来提高查询效率。实验结果表明, 该模型是一种高效的海量地震数据组织方式, 在地质建模领域具有较高的应用价值。

关键词: 海量地震数据; 多分辨率; 八叉树

Multi-resolution Ratio Extended Octree Model Based on Mass Seismic Data

ZHANG Lei, TANG Jie, WU Gang-shan

(Department of Computer Science and Technology, Nanjing University, Nanjing 210093)

【Abstract】 Aiming at limitations of existing PC hardware and the feature of mass seismic data, this paper presents a partitioning multilevel multi-resolution model based on extended octree. The model stores attribute of leaf nodes and intermediate nodes, realizes multi-resolution modeling, and uses data block index method based on Morton code to enhance query efficiency. Experimental results show that the model is an efficient organization style of mass seismic data, and it has well application value of the geological modeling.

【Key words】 mass seismic data; multi-resolution ratio; octree

1 概述

在地质建模中, 通常需要对大规模地震数据进行分析以发现地下潜在可能的油气资源。地震数据是体数据的一种, 通过野外地震勘探施工得到。其采集包括测量、钻浅井孔埋炸药、埋检波器、布置电缆线至仪器车、爆破产生地震波、仪器车接收地震波、得到地震数据等多道工序。地震数据是地质解释中经常遇到的数据, 该数据按照一定分辨率规则采样得到, 数据往往是海量的。由于整个场景不能全部载入主存, 因此要把数据放在容量更大的辅存上。而辅存的读写速度非常慢, 导致程序将大量的时间耗费在频繁的 I/O 上, 成为实时绘制的瓶颈。基于外存的算法尽可能减少不必要的 I/O 操作, 避免由频繁的 I/O 操作引起的大规模数据场体绘制性能的急剧降低。

许多学者对该问题做了研究。文献[1]提出一种利用空间跳跃技术与三维纹理硬件结合的体绘制方法, 有效降低了纹理硬件的数据交换量。文献[2]提出一种基于数据重要性的八叉树可视化方法, 这 2 种方法较适合处理分类较好的数据。文献[3]利用小波变换对体数据进行无损压缩, 在绘制时对编码数据实时解码, 每个纹理需要在 CPU 上完成解码后载入图形硬件进行绘制, 耗时、且压缩/解码过程复杂。文献[4]提出采用矢量量化技术对体数据进行压缩, 在一定误差内可以自动得到压缩后的数据来近似表示原始数据。但仅采用最近邻插值进行绘制, 影响了绘制图像的质量。由于地质人员更加关注地震数据绘制的实时性, 对外存消耗的关注通常是第二位的, 因此数据压缩不是必要的, 它甚至有可能影响绘制的速度。文献[5]提出一种多方向的数据分块方法。

针对目前海量地震数据实时可视化算法存在的问题, 本

文提出的模型对海量地震数据进行分块并建立八叉树层次结构, 实现了在 PC 机上对大规模体数据的交互体绘制。

该模型具有以下优点:

(1) 没有使用复杂的编码方式对体数据进行压缩, 避免了在绘制中耗时的解码操作对交互性的影响以及编解码带来的误差。

(2) 保存了中间节点和叶子节点, 可以实现多分辨率建模。

(3) 对每层数据按 Z-Order 顺序存储在磁盘, 在快速查询方面具有较高的效率。

2 数据分块

在任何一个处理海量地震数据的算法中, 优化数据的组织, 减少 I/O 操作是很重要的。一个有效的解决办法是将整块地震数据划分为较小的块。数据块的大小将会对系统的效率产生重要影响, 分块过小会导致 I/O 中断次数增加, 影响系统效率, 而过大的分块会使单次读取时间增加, 导致一些无用数据的读入。因此, 必须根据操作系统类型、磁盘缓存大小、磁盘组织方式(单块磁盘还是 RAID)等因素, 选择合适的分块大小。

一般来说, 数据块的大小应该保持在 0.5 MB~4 MB 之间^[6], 同时为了使数据块能够载入显存, 要确保数据块的边

基金项目: 江苏省自然科学基金资助项目(BK2008262); 江苏省高新技术研究(工业部分)基金资助项目(BG2007037)

作者简介: 张磊(1985-), 男, 硕士研究生, 主研方向: 计算机辅助几何设计; 唐杰, 副教授; 武港山, 教授

收稿日期: 2009-05-21 **E-mail:** zhang20020901@163.com

长为 2 的整数次幂, 如, $32 \times 32 \times 32$ 个体素或 $64 \times 64 \times 64$ 个体素等。一旦选定了合适的分块大小, 磁盘的单次读取操作将以数据块为单位, 即一次读取整个数据块。

3 分块多层多分辨率扩展八叉树模型

八叉树结构的概念是由 Hunter 博士于 1978 年在其博士论文中首次提出的一种数据结构。它是由四叉树结构推广到三维空间而形成的一种三维数据结构, 其树形结构在空间分解上具有很强的优势, 因此, 得到广泛应用。在八叉树的树形结构中, 根节点表示整个三维空间区域。将该区域分成 8 个大小相同的小区域, 用其 8 个子女表示。对于每一个这样的区域, 继续将其分成 8 个更小的区域。按此规则一直分割到不再需要分割或达到规定的层次为止。常用的八叉树主要有: 指针八叉树, 线性八叉树等。指针八叉树是最普通的八叉树结构, 它和常见的二叉树类似, 通过内存指针指向其子女。由于指针八叉树只在内存中实现, 因此其查询效率较高, 但大量的内存指针消耗了宝贵的内存空间, 同时也不适合于海量数据。线性八叉树由于只保存了叶节点属性数据和空间位置, 不保存中间节点信息, 因此, 不易实现多分辨率建模。

本文的扩展八叉树模型以线形八叉树模型为基本数据结构, 在保存叶节点属性的同时, 也保存中间节点属性, 可以实现多分辨率建模。

3.1 层次结构的构建

由于目前的地震数据通常以长方体的研究区域居多, 因此立方体和近似立方体的研究区域较少, 传统的八叉树 x, y, z 方向上的分解度通常是相同的。这对不同方向上尺度相差较大的地震数据来说会造成有的方向上分辨率不够, 而另外的方向上分辨率过高的情况。鉴于此, 本文采用自底向上的方法构建八叉树, 该方法可以有效控制研究区域内不同方向的分辨率, 统一不同方向上的分辨率, 在一定程度上减少存储空间的占用, 避免在某个方向上的浪费。

在构建层次模型时, 首先将原始数据分块, 作为最高级分辨率数据。在最高级数据的基础上, 将相邻 8 个数据块合并为一个数据块, 形成次高级分辨率数据。依次循环直至达到某一条件, 完成多分辨率层次数据的构建。所有分块大小相同, 分块所在的层次越低, 每个分块代表的空间大小越大。

为了防止绘制时块与块之间产生裂缝, 对每块的边界做特殊处理。本文采用的方法是对每个块的“高”边界重复一个像素。图 1 表示其一维示意图。

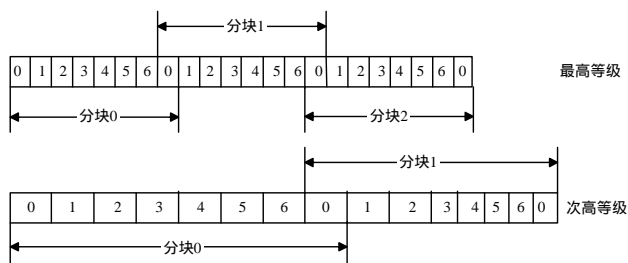


图 1 重复边界体素

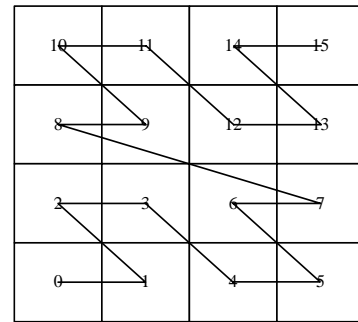
在图 1 中, 每个分块有 8 个体素。最高等级的分块 0 和分块 1 的有效体素是 0~6, 它们的最右边的体素是为了避免裂缝而重复其下一个分块“低”边界的体素。最高等级的分块 2 是边界分块, 分块 2 的“高”边界无需再重复一个体素, 但为了保证每个分块的大小相同, 对分块 2 的“高”边界补

一空白体素。同样, 次高等级的分块 0 的“高”边界也重复了一个体素, 分块 1 的 4 体素、5 体素、6 体素、0 体素是为了使每个分块大小相等而补齐的空白体素。

3.2 数据块索引与查询

空间填充曲线是一条过且仅过一次 N 维空间中每个点的曲线, 它可以将 N 维空间中的点映射到一维空间。空间填充曲线有很多种, 如: Sweep, Scan, Z-Order, Hilbert 等。但 Z-Order 空间填充曲线有更好的局部性, 按 Z-Order 顺序 (Morton 码由大到小) 保存的节点的空间相邻关系也能得到较好的保持, 有助于提高磁盘缓存命中率, 并且三维坐标与 Morton 码之间的转换效率高, 使用简单的移位操作即可快速计算。采用 Z-Order 顺序保存节点后, 每个节点不必保存父子兄弟关系, 此类信息可以通过节点 Morton 码的快速计算得到, 节省了存储空间。

对于每一层数据块, 按照 Z-order 顺序将三维的地震数据块映射到一维的磁盘空间中, 每层数据对应一个一维线性存储空间, 映射结果如图 2(a) 所示。由于目前地震数据的研究区域多为长方体, 使用 Morton 码作为地址码会产生大量的空白 Morton 码, 即某个 Morton 码没有节点与之对应, 结果如图 2(b) 所示。



(a) 结果 1

$j=10$	8	9	12
$j=01$	2	3	6
$j=00$	0	1	4
	$i=00$	$i=01$	$i=10$

(b) 结果 2

图 2 Morton 编码

其中, Morton 码 5, 7, 10, 11 没有对应的数据。数据块按 Morton 码的顺序存储在磁盘中, 因此 Morton 码为 5, 7, 10, 11 的位置没有数据, 但也会在磁盘上占用一个数据块大小, 造成磁盘空间浪费。

为了解决此问题, 本文提出 2 种 Morton 码表方法:

(1) 三维 Morton 码表。使用数据块的三维索引作为码表的下标, 数组的内容为有效 Morton 码。如图 2(b) 所示, Morton 码为 6 的数据块, 在码表中的 Morton 码为 5, 同样, Morton 码为 12 的数据块, 在码表中的 Morton 码为 8。该方法的优点是不浪费 Morton 码表的存储空间, 但在该情况下, Morton 码只能索引数据块, 数据块的父子兄弟关系不能通过 Morton

码快速计算得到。三维 Morton 码表如图 3 所示。

0	1	2	3	4	5	6	7	8
A[0][0]	A[1][0]	A[0][1]	A[1][1]	A[2][0]	A[2][1]	A[0][2]	A[1][2]	A[2][2]

图 3 三维 Morton 码表

0	1	2	3	4	-	5	-	6	7	-	-	8
A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]	A[9]	A[10]	A[11]	A[12]

图 4 一维 Morton 码表

(2)一维 Morton 码表。使用数据块的原始 Morton 码作为码表下标，码表的内容为有效 Morton 码。对于图 2(b)而言，产生的一维码表形式如图 4 所示。该方法的优点是数据块之间的父子兄弟关系可以通过 Morton 码快速计算得到，但会造成 Morton 码表的空间浪费。

考虑到 Morton 码表的数组元素都是 int 型变量，在保证父子兄弟关系快速计算的前提下，这样的空间浪费是可以容忍的。因此，本文采用一维 Morton 码表的方式。

Morton 快速计算父子兄弟关系表现在以下 2 个方面：

(1)父子关系：第 i 层父曲线上 Morton 码为 M 的节点，将 M 乘以 8，得到它的第 1 个子女在第 $i+1$ 层曲线中的位置。反之，将 M 除以 8，就得到它的父节点在第 $i-1$ 层曲线中的位置。这 2 个过程可以通过左/右移 3 位来实现，实现效率较高。

(2)兄弟关系(兄弟指同一个父节点下的兄弟)：

1)已知父的 Morton 码，计算父的任意子女的 Morton 码。首先通过移位得到父的第 1 个子女的位置 M_0 ，然后根据子女的坐标再次计算，具体公式如下：

$$M = M_0 + Z \times 4 + Y \times 2 + X \quad (1)$$

2)已知任意节点 Morton 码，计算其任意兄弟的 Morton 码。已知节点的位置 M_0 ，然后根据其领域的坐标计算得到兄弟的位置，具体公式如下：

$$M = M_0 + Z \times 4 + Y \times 2 + X \quad (2)$$

其中，子女、领域坐标均以坐标正方向为 1；负方向为-1。

4 数据调度

本文采用的多分辨率调度策略是自顶向下的。当要进行细化操作时，从当前数据块开始，对每个数据块测试判断是否在视锥内，若不在视锥内，则不再对该分支进行测试；若在视锥内，则继续测试其 8 个孩子的数据，直到内存中数据块数目达到要求。反之，当视点平移或要进行粗化操作时，从八叉树顶层开始搜索。使用本文提出的扩展八叉树模型，无须进行拓扑关系运算，仅使用移位操作和数据块索引即可快速定位孩子数据块。

为了进一步提高实时性，使用数据缓存机制。当程序需要的数据块在缓存中时，直接从缓存中得到数据，无须访问辅存，从而节省了 I/O 时间。

5 实验结果分析

本文实验采用的地震数据为石油勘探工业界中最广泛使用的 SEG-Y 格式数据。SEG-Y 地震数据格式是美国 SEG 学会(勘探地球物理学家协会)制定并推荐的一种交换格式标准，其数据属于标量数据场。

在 CPU 为 Intel Core T6300，内存为 1 GB，显卡为 ATI Radeon X1300Pro 256 MB，Windows XP 的 PC 上对算法进行测试。单个分块的时间读取结果如表 1 所示。

表 1 读取单个分块的时间 ms

输入文件体素	分块大小(32^3)	分块大小(64^3)
601×601×6 001(8.15 GB)	5.40	35.20
1 202×1 202×6 001(32.6 GB)	-	38.60

测试实验效果如图 5 所示。

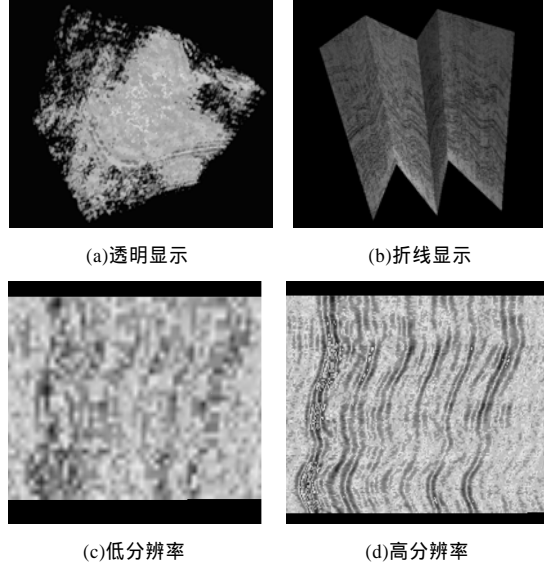


图 5 实验效果

6 结束语

本文提出的扩展八叉树模型能够对海量地震数据进行建模，具有较高的节点查询效率，在实时绘制中能够通过索引快速检索所需的节点，提高了绘制速度。本模型已应用于自行开发的地质建模软件中，结果证明了其可行性。

参考文献

- [1] 董欣, 唐圣泽. 基于空间跳跃的三维纹理硬件体绘制算法[J]. 计算机学报, 1998, 21(9): 807-812.
- [2] Boada I, Navazo I, Scopigno R. Multi-resolution Volume Visualization with a Texture-based Octree[J]. The Visual Computer, 2001, 17(3): 185-197.
- [3] Guthe S, Wand M, Gonsler J, et al. Interactive Rendering of Large Volume Data Sets[C]//Proc. of IEEE Visualization 2002. Boston, Massachusetts, USA: IEEE Computer Society, 2002.
- [4] Schneider J, Westerm A R. Compression Domain Volume Rendering[C]//Proceedings of the 14th IEEE VIS'03. Seattle, WA, USA: IEEE Computer Society, 2003: 293-300.
- [5] Daniel W, Manfred W, Thomas E. Maintaining Constant Frame Rates in 3D Texture-based Volume Rendering[C]//Proceedings of Computer Graphics International. Washington D. C., USA: IEEE Computer Society, 2004: 604-607.
- [6] William R. Gigabyte Volume Viewing Using Split Software/Hardware Interpolation[C]//Proceedings of the 2000 IEEE Symposium on Volume Visualization. New York, USA: ACM Press, 2000: 15-22.

编辑 陆燕菲

