

粒度与结构统一的多层次三维 城市模型数据组织方法

周东波¹ 朱 庆² 杜志强² 张叶廷²

(1 华中师范大学国家数字化学习工程研究中心,武汉市珞喻路 152 号,430079)
(2 武汉大学测绘遥感信息工程国家重点实验室,武汉市珞喻路 129 号,430079)

摘 要:提出了一种高效的磁盘、内存、显存三级数据存储粒度与结构一致的数据组织方法。首先,根据城市空间对象数据内容,建立层次嵌套、多类型混合的空间索引结构,以空间索引节点统一磁盘存储、内存场景管理与绘制缓存的基本操作单元;其次,以绘制缓存对象的紧凑存储结构为基础,统一内存场景管理对象与磁盘存储对象的数据块结构,并将数据布局方法扩展到对象粒度进行磁盘存储组织。实验结果表明,所提方法大幅度减少了 I/O 次数,提高了数据调度效率,为大规模三维城市模型的实时绘制奠定了基础。

关键词:实时绘制;三维城市模型;数据组织

中图法分类号:P208

随着三维城市模型数据量呈现出几何级数增长的趋势,大规模数据的存储与管理方法成为近期研究焦点。已有研究多集中于解决大数据量从慢速的硬盘到高速的内存、显存间的传输瓶颈^[1],所提出的方法可以分为两类:减少磁盘访问数量的外核技术^[2]和考虑特定磁盘访问模式的数据布局技术^[3-6]。现有的数据布局技术只是针对几何图元一级数据的存储与组织,并没有考虑在此粒度下纹理存储与组织方法的巨大差异。因此,在对象数量众多、个体差异大、几何复杂、纹理多样的三维城市模型进行组织与存储时,由于不同存储层次中数据表达粒度不一致,内存管理中的对象将被分裂到多个磁盘存储对象中,一个绘制对象也分属于多个内存管理对象。磁盘存储对象、内存管理对象及绘制缓存对象的数据组织粒度不一致,导致单一数据请求需要多个磁盘读取 I/O 操作才能完成。这将导致较多的磁盘 I/O,花费较长 I/O 的寻址时间,读取和传输并不需要的数据。

针对上述问题,本文根据现有计算机系统的存储结构模型,建立了磁盘存储对象、内存数据管理对象与绘制缓存对象数据粒度与存储结构一致

的几何与纹理数据组织方法,并结合磁盘存储中基于对象粒度的布局技术,减少数据获取过程中的磁盘 I/O 次数以及 I/O 的寻址时间。

1 粒度统一的多级存储数据组织方法

现有的三维 GIS 软件及应用中,系统注重表达城市中的每一个实体对象,在内存管理中用面向对象的模型来表达,在绘制缓存中使用基于面的粒度,而在磁盘存储中将整个工程数据存储为一个或少数文件。针对三维城市模型的构成特点,本文建立了一个多层次、混合空间索引结构,以空间索引的节点单元来构建磁盘存储对象、内存管理对象与绘制缓存对象的统一粒度大小,并使用基于对象粒度的磁盘数据布局方法存储数据。

1.1 多层次混合空间索引

三维城市模型中数据量大、关系复杂,其数据特点表现为:空间数据分布不均衡,局部稀疏程度不一,单位空间体块中模型属性不唯一;模型的几何与纹理表达中,几何拓扑关系复杂,纹理数量、类型过多等。本文将城市空间中的数据内容分为

7 大类:地形数据、数字正射影像图数据、建筑模型数据、道路网络数据、地质数据、管线数据以及其他设施模型数据。

数以 10 万计的城市模型对象在整个城市范围内的数据分布,从宏观上看,可以认为对象散列在城市的平面上;而在不同局部范围内,对象密集分布且空间位置关系复杂,不同类型的数据分布差异大。地形、正射影像图等数据呈现平面分布特点,而地质、建筑模型等以单个实体表达的数据则呈现局部聚集特点。因此,对整个大范围城市空间中的多种地物,难以用单一的空间索引结构来检索所有类型的数据。本文首先对不同数据内容的空间索引进行层次组合,然后使用空间索引叶节点表达内存管理单元,进而统一绘制缓存对象与磁盘存储对象。

索引分为两个层次:第一个层次为格网索引,针对城市空间范围建立大的分区格网。格网的大小根据城市的范围而定。第二个层次根据三维城市模型对象的数据内容而定:对于地形、正射影像图采用四叉树索引组织,并依此建立层次多分辨率表达;管线模型以及道路网络模型使用格网索引;建筑模型以及其他设施模型等个体对象,计算其对象的包围盒范围,构建 LOD-R 树空间索引结构^[5];而地质体模型则采用八叉树进行组织。

1.2 基于多层次空间索引的数据粒度统一

本文采用内存映射文件存储几何与纹理数据。存储组织依据空间索引组织进行,索引单元内地物模型数据对应 3 个内存映射文件,即空间索引映射文件,存储格网单元中不同地类所建立的空间索引及其对象关联信息;几何数据映射文件,存储索引单元内所有几何对象,包括 LOD 数据;纹理数据映射文件,存储单元中所用到的纹理贴图。对于几何数据文件与纹理数据文件,内存映射文件的存储组织中使用类型分段、节点分块、对象分项建立 3 个基本存储粒度单元。

类型分段:几何数据文件与纹理数据文件组织依其所对应的空间索引结构进行分段,即同一个空间索引结构的所有地物数据存储在同一段中。

节点分块:存储映射文件中,一个段对应着一个空间索引结构,段内数据组织则根据此空间索引的节点进行分块,一个空间索引节点对应一个存储数据块结构。在数据块内,依据数据类型进行聚合,将同一地物类的数据顺序存储。

对象分项:存储数据块中,最小组织单元为数据项,对应内存管理中一个地物对象,以及空间索

引结构中的一个叶节点。内存管理对象中的每一个地物对象,通过计算其包围盒结构,并根据地物类插入到相应的空间索引结构,表达为空间索引树结构的一个叶节点。从而建立起磁盘存储数据对象、内存管理对象与绘制缓存对象的对象一级粒度对应关系。

2 结构统一的存储方法

统一磁盘存储对象、内存管理对象以及绘制缓存对象的粒度,可以建立起三者之间操作单元的对应关系,从应用端出发,可以得到最小的不同级存储设备间的数据交换次数。然而,不论是在磁盘、内存还是显卡缓存中,数据都是以一维的线性数据序列进行表达。若不考虑结构的统一,数据从磁盘取出后,进入内存管理时,需要进行线性序列的变化,以达到内存管理所需要的模式;从内存传输到绘制缓存时,需要将内存的对象模型转换为绘制的顶点数组或显示列表,以达到最终为绘制所接收的内容。

磁盘的存储中,首先需要建立与内存及绘制缓存存在不同粒度模式下相匹配的文件组织与存储方式,达到以最少的 I/O 次数,读取恰好需要的数据。另外,针对现有磁盘访问的特点,采用基于数据项粒度的磁盘数据布局,以实现连续访问需求数据的连续读取,从而减少磁盘的寻址时间。

2.1 磁盘文件组织

本文将地物对象的空间索引信息、几何数据与纹理数据分别存储于索引映射文件、几何数据映射文件与纹理数据映射文件中。索引映射文件存储了每一个地物对象的基本属性信息及其在几何与纹理映射文件中的位置。几何映射文件依据空间索引节点的顺序,将每一节点关联数据内容以块的形式存储到文件中。而纹理映射文件中,根据空间索引层次结构建立纹理树,并将纹理数据打包写入映射文件。对城市范围建立多个不同分区,每一个分区对应混合索引结构中格网索引中的一个格网,每一个分区内的地物对象及其索引数据存储在上述 3 个映射文件中。

随机数据访问中,无法预先判定所需要的数据。因此,数据块写入映射文件的顺序影响其读取效率。通过使用空间索引结构,快速判断所需要对象过程中场景空间索引结构的访问模式成为应用中的数据访问模式。因此,数据的磁盘存储中,利用空间索引节点访问顺序表达磁盘数据存储顺序,可以实现连续读取操作,减少寻址时间。

本文的磁盘存储中,针对不同类型的索引建立与访问顺序相应的存储顺序。四叉树索引与 R 树索引的层次结构中采用先序遍历的顺序,格网索引中采用基于行的顺序存储。

2.2 内存数据组织结构

内存存储数据块采用基于绘制缓存的高效数据结构进行组织,而目前公认高效的组织表达为索引三角形条带组织。因此,原有的顶点列表与多边形列表的组织结构需要进行有效的条带化。本文使用 nVidia 公司的 NvTriStrip 库^[6],根据不同的粒度单元,将其数据构建条带,通过条带的序列化将其表达成一维线性连续存储空间。根据数据所对应的粒度将内存块加入内存管理对象集,内存对象管理集表达为不同粒度内存块列表。各

个内存管理对象与混合空间索引结构中的组织结构一一对应。

3 实验与分析

3.1 实验环境

测试中选择了几个典型的场景,第一个是校园模型;第二个是校园地下模型;第三个是大范围城市地上模型。计算配置情况为:CPU 为 Intel i7 处理器;内存为主频为 1 066 MHz 6 GB;硬盘为 2 个 7 200 转 500 GB 硬盘;显卡为 nVidia GeForce GTX260 带 1 GB 显存;操作系统为 Windows XP。测试场景全景渲染效果如图 1 所示,实验数据情况如表 1 所示。

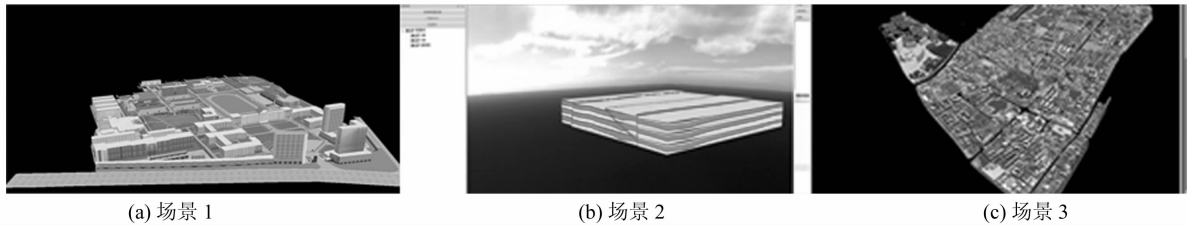


图 1 测试数据场景全景图
Fig. 1 Scenes of Test

表 1 测试场景数据表
Tab. 1 Data of Test Scenes

场 景	几何文 件数量	纹理文 件数量	几何数据 量/MB	纹理数 据量	地物对象 数量/个	纹理数 量/个
1	2	575	9.1	61.1 MB	7 156	575
2	3	572	12.78	61 MB	6 086	572
3	5	1 112	63.7	2.48 GB	10 708	1 112

3.2 实验结果与分析

实验 1 对比现有 GIS 软件非粒度一致的组织方法与本文的组织方法,结果如表 2 所示。测试中,对可见数据进行加载。本文设置了两个视场大

小:第一个视场为数据局部范围,场景 1 和场景 2 较小数据量的情况;第二个视域范围表达全部的场
景,而对场景 3,选择不同大小的范围。第一个查
询范围大小为第二个查询范围的 2 倍。从结果来
看,本文方法在几何数据的组织上,与原方法相比,
I/O 数量明显减少,且内存缓存数据量小,主要由于
对几何数据进行分块存储后,只需要将空间范围
包围的数据取出,并不需将整个几何文件取出然后
才能得到对象。在数据全部读取时,内存缓存数据
量差异不大,而加载的磁盘 I/O 数大大减少。

表 2 不同存储组织中磁盘 I/O 次数与缓存大小对比表

Tab. 2 Comparison of I/O and Cache Size Between Different Methods

		非粒度一致的组织方法			本文方法		
		场景 1	场景 2	场景 3	场景 1	场景 2	场景 3
查询一	磁盘 I/O 次数	291	245	123	68	62	41
	缓存大小	166 MB	165.8 MB	412 MB	132 MB	131.5 MB	380 MB
查询二	磁盘 I/O 次数	582	579	423	158	160	111
	缓存大小	281 MB	295.2 MB	1.21 GB	281 MB	295 MB	1.04 GB

实验 2 对比了场景 1 在不同存储组织下的加载与解析时间,结果如表 3 所示。从表 3 中可看出,几何数据的时间加载与解析提高达 1 倍,而纹理数据的加载提高达 4 倍,解析时间提高约 1 倍。
对场景 3 进行实时漫游,场景范围 6 km²,地物对象达 1 万个以上,几何与纹理总数据量达

表 3 格式转换时间开销对比表/s

Tab. 3 Time Consuming of Dataconversion/s

	几何加载 时间	格式转 换时间	纹理加载 时间	格式转 换时间
非结构一致格式	0.57	2.47	3.8	1.07
本文结构一致格式	0.24	1.20	0.72	0.56

2.5 GB。采用本文的数据组织方法,结合多线程动态调度,可以稳定地达到 20 帧/s 以上的效果(见图 2)。

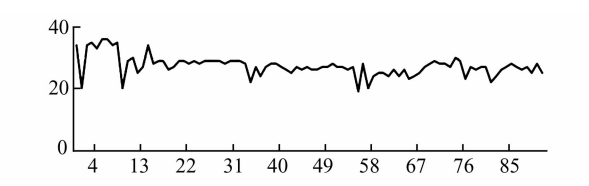


图 2 场景 3 实时绘制帧率

Fig. 2 Real-Time Frame Per Second of Scene 3

4 结 语

磁盘存储对象、内存场景管理对象与绘制缓存对象数据组织粒度与结构的不一致,导致产生过多磁盘 I/O 而引起磁盘访问延迟,成为数据调度延迟的一个主要原因。本文通过基于城市模型数据内容的层次混合空间索引结构建立三者的统一粒度,并以应用端数据结构为基础,统一内存与磁盘的存储结构,减少格式转换代价。下一步工作考虑在磁盘建立基于空间访问特征的数据存储顺序来减少磁盘寻址开销,提高实时绘制效率。

参 考 文 献

[1] Sajadi B, Huang Y, Diaz-Gutierrez P, et al. A Novel Page-Based Data Structure for Interactive Walkthroughs[C]. Association for Computing Machinery, Boston, Massachusetts, 2009

[2] 吴玲达,高宇,魏迎梅. 大规模复杂场景交互绘制技术综述[J]. 计算机研究与发展, 2007, 44(9): 1 579-1 587

[3] Yoon S L P P, Manocha D. Cache-Oblivious Mesh Layouts[J]. ACM Transactions on Graphics, 2005, 24(3): 886-893

[4] Sander P V, Nehab D, Barczak J. Fast Triangle Reordering for Vertex Locality and Reduced Over Draw[J]. ACM Transactions on Graphics (TOG), 2007, 26(3): 89-97

[5] 朱庆,龚俊,杜志强,等. 三维城市模型的多细节层次描述方法[J]. 武汉大学学报·信息科学版, 2005, 30(11):965-969

[6] 朱庆,龚俊. 一种改进的真三维 R 树空间索引方法[J]. 武汉大学学报·信息科学版, 2006, 31(4): 340-343

第一作者简介:周东波,博士,讲师,研究方向为虚拟地理环境。
E-mail:zhoudongbo@263.net

A Data Granularity and Structure Coherence Organization
Method for Large-Scale 3D City Models

ZHOU Dongbo¹ ZHU Qing² DU Zhiqiang² ZHANG Yeting²

(1 National Engineering Research Center for E-Learning, Central China Normal University,
152 Luoyu Road, Wuhan 430079, China)

(2 State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing,
Wuhan University, 129 Luoyu Road, Wuhan 430079, China)

Abstract: The data granularity and structure difference among the disk storage objects, memory management objects and rendering cache objects results to heavy counts of disk I/Os. This becomes one of the reasons causing data access delay. We propose a data organization method to uniform the granularity and structure in disk, memory and display cache. Firstly, we use the spatial index node from a hierarchical-nested and hybrid spatial index to uniform the disk storage, memory management and rendering cache basic operation unit. Secondly, the data structure is formatted in disk and memory based on the structure of rendering cache objects. And the data layout method is extended to the object level using in disk storage to reduce the data format convention and disk I/O seeking time. Tests prove the obvious reduction of I/Os in the real-time rendering of large-scale 3D city models.

Key words: real-time rendering; 3D city model; data organization