基于分布式集群的高分机载激光雷达对地观测数据存储和计算方法

叶小川[1]

1. 武汉大学测绘学院，湖北 武汉 430079；

Data Storage and Calculation Method of Airborne Lidar Earth Observation Based on Distributed Cluster

YE Xiaochuan[1]

1. School of Geodesy and Geomatics, Wuhan University, Wuhan 430079, China

Abstract: With the improvement of lidar accuracy and the development of UAV technology, high-resolution airborne lidar earth observation can generate large-scale point cloud data. Aiming at the problem that the point cloud data is too large and difficult to store and calculate, this paper discusses the spatial index technology suitable for high-resolution large-scale point cloud data such as octree, K-D tree and distributed computing technology such as Hadoop. B-EagleV architecture is discussed at the whole process and specific application of point cloud data processing.

Key words: point cloud; high resolution; earth observation; distributed system; Hadoop; spatial index

摘　要：随着激光雷达精度的提高和无人机技术的发展，高分辨率的机载激光雷达对地观测能够产生的大规模点云数据。针对于点云数据过大难以存储、计算的问题，本文探讨了适合高分辨率大规模点云数据如八叉树、K-D 树的空间索引技术和 Hadoop 等分布式计算技术，举例探讨了以 B-EagleV 架构针对于点云数据处理的全流程和具体应用。

关键词：点云；高分辨率；对地观测；分布式系统；Hadoop；空间索引

0 引　　言

当今社会已经进入信息时代。三维空间数据作为信息的一种,可以表示空间物体的 位置信息和其它附加信息，人们可以根据这些信息，更进一步理解空间物体。

同时，科学技术的发展也丰富了人们获得空间数据的手段。特别是激光雷达技术的迅猛发展，使 得获取物体表面几何信息和灰度信息变得越来越容易，促进了激光扫描技术的发展。

同时，随着无人机 UAV（unmanned aerial vehicle)技术的日渐成熟和应用领域的不断扩展，激光雷达也逐渐开始被安装在无人机上实行勘测任务。机载激光扫描系统 LiDAR(Light Laser Detection and Ranging）作为一种新型的遥感系统，具有全天候、主动、实时、快 速、高效率、高密度等优点[1]。

目前，机载激光扫描系统已经在海岸带测绘、自然灾害监测、油气管道线路测绘和道路施工勘察等多个领域有了长足的应用，同时还可以精确、 快速地生成地形、植被和建筑物的三维表面信息，应用于地形测绘、植被图绘制、城市三维数字建模、土地沙化监测、国防边境安全等领域[2-4]。目前机载激光扫描系统的扫 描频率已经可以达到几十万点/秒以上，理论上一次飞行可采集数亿个点[5]。

虽然机载激光扫描系统得到了长足的发展，但是针对于大规模对地观测的点云数据进行存储和处理问题亟需解决，如何在分布式集群中对对地观测的点云数据进行存储和高效查询，是需要解决的问题。

1 机载激光点云对地观测

机载激光点云是经机载激光扫描系统采集，通过坐标转换到同一坐标系下的空间数 据点集,其属性主要包含三维坐标信息和激光反射强度等。机载激光点云高精度、高密 度、以及低成本的特点使其在地形测绘、深海探测、自然灾害评估等领域得到广泛的应用。

1.1 机载激光扫描系统

机载激光扫描系统主要由机载系统和地面系统构成。其中机载系统主要包括激光雷达、惯性测量单元 IMU（Inertial Measurement Unit）、全球定位系统 GNSS 移动站、嵌入式系统及无线数据电台，其主要功能是实现各传感器数据的采集，并传送给地面服务器。

图 1 机载激光扫描系统
Fig.1 UAV Lidar System

地面系统主要由无线数据电台、GNSS 基站和服务器构成。其主要功能是完成数据接收以及后续的数据管理、处理及显示工作。机载激光雷达是机载激光扫描系统的核心传感器，可以获取目标的距离信息，它的性能直接影响最终的点云成图效果。惯性测量元件可以获取物体三轴的姿态角。GNSS 移动站和 GNSS 基站组成的差分 GNSS 可为机载激光扫描系统提供厘米级的定位服务。

1.2 机载激光点云数据生成

机载激光点云由激光雷达、惯性测量单元 IMU 和 GNSS 的数据融合得到。激光雷达通过扫描目标可得到与目标的距离，惯性测量单元 IMU 可得到姿态信息，GNSS 移动站和地面 GNSS 基站可确定位置信息。通过这三类数据的融合，并进行坐标转换后，可计算得到激光点云的坐标。其中，涉及的坐标系主要有激光雷达坐标系，惯性平台坐标系和当地水平坐标系。

2 点云数据索引技术

网格划分、K-D 树和八叉树是三种常用的空间数据索引技术，它们均是自顶向下，逐级对空间进行划分的索引技术。

2.1 基于网格划分的索引技术

基于网格划分空间，实际上在是把三维空间等分为多个子空间，每个子空间都是一个三维网格，记录下每个空间内的数据。要查找某个数据的时候，只需要按照一定规则先查找到它所在的网格，然后再在这个网格内逐一查找即可[6]。

如果把网格划分为 $I\times J\times K$ 个子空间，那么数据查找的范围就缩小到了原查询范围的 $1/(I\times J\times K)$。此外，基于网格划分的方法也可以很容易地进行二次划分。如果觉得一级划分的粒度还不够细，一个网格内的数据还是太多，可以使用同样的方法对每一个一级网格进行二级划分，对每个二级网格进行三级划分，以此类推。类似于查找地址的场景，如果只在国家层次进行了划分，需要直接在一个国家内精确查找某个地址，查找开销巨大，显然划分粒度太粗。此时，可以再进行细粒度的省、市、区的划分。查找时通过所在国家、省、市、找到所在区，再在区内精确查找地址，此时，查找效率得到明显的提升。

网格划分实际上是一个哈希过程。空间数据 (x,y,z) 的外接包围盒为 $\{x_{\min},x_{\max},y_{\min},y_{\max},z_{\min},z_{\max}\}$，网格划分间距为 $\Delta x,\Delta y,\Delta z$，可以利用以下的哈希函数进行网格划分，得到三维网格编号 (i,j,k)：

$$i=INT(\frac{x-x_{\min}}{\Delta x}) \tag{1}$$

$$j=INT(\frac{y-y_{\min}}{\Delta y}) \tag{2}$$

$$k=INT(\frac{z-z_{\min}}{\Delta z}) \tag{3}$$

这样就完成了一个网格对应多条数据，一条数据对应一个网格的映射。如果要进行点查询，可以计算出该点所在的网格编号，先查找网格，再在网格内查找数据；如果要进行区域查询，就是找出区域所包含的网格内的所有数据；如果要进行 K 近邻查询，一种近似的方法是在目标点所在的网格内找 K 近邻。若点的数目小于 K，则在该网格近邻的网格内继续找，直到找到 K 个近邻。

2.2 基于八叉树的索引技术

八叉树是一种树型结构，它递归地把空间平均划分为 8 个子空间[7]。8 分三维空间是 4 分二维空间和 2 分一维空间的拓展，即八叉树是四叉树、二叉树的拓展。八叉树的每一次划分实际上就是在每个维度上都用一个面二分空间，三个维度同时进行划分之后，得到 8 个子空间再在每个子空间中继续进行同样的划分，直到划分深度达到最大树深。设定八叉树最大深度的原因是要定义最小子空间的外包围盒大小，即最小的划分粒度。设划分的空间总个数为 N，插入新数据或者查找数据的时间复杂度为 $O(\log_8 N)$。八叉树还可以设计其它的划分终止条件，例如子空间的数据少于一个设定值就不继续划分等。如果每个子空间都进行划分，直到最大深度，即可得到满八叉树。满八叉树可以看作特殊的网格划分，相当于把空间划分 $2^n\times 2^n\times 2^n$ 个三维网格，n 是树的深度。

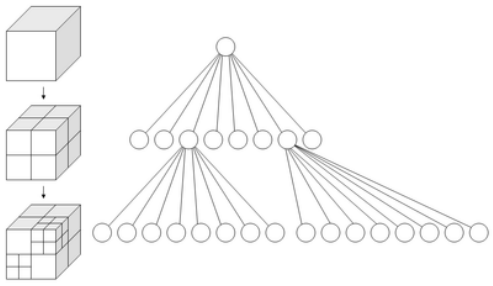

图 2 八叉树进行空间划分
Fig.2 Spatial Divided By Octree

八叉树的根结点表示整个待索引空间，根结点的八个指针分别指向划分出来的八个子空间。非叶结点存放空间外包围盒的位置信息，一般有八个子结点，如果定义子空间中无数据就停止树的划分，非叶结点就退化成叶结点，如图 2 所示。满八叉树的叶结 点代表最小的空间，数据的引用一般存放在叶结点上。如果数据是空间形状或者物体，并且同时与多个空间相交的时候，数据的引用还可以设计放在非叶结点上。

2.3 基于 K-D 树的索引技术

K-D 树（K-Dimension Tree）是一种空间划分树，也是二叉树的拓展，主要应用于多维数据的搜索[8]。对于三维空间，K-D 树的 K 就是 3。K-D 树每一个结点都有一个划分 超平面与之对应。K-D 树也是通过递归的方式在 K 个维度上轮流对空间进行划分。与八叉树不同，K-D 树的划分不是等分。这 K 个维度需要设定一个顺序，每次按设定顺序选定一个维度后，把数据集上这个维度上的中位数作为划分值，小于该值的数据划分到左子树，大于等于该值的数据划分到右子树，一直重复进行，直到结点里的数据量少于预设值或者树深达到最大树深。

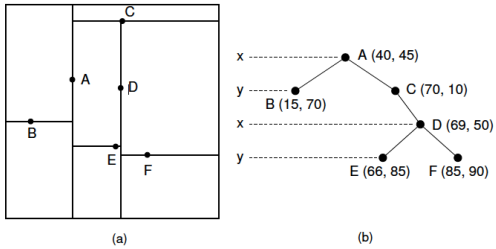

图 3 K-D 树
Fig.3 K-D Tree

K-D 树有如下特点，当一个结点的左子树不为空时，左子树的所有结点的第 d 个维度的值一定小于它第 d 维的值。当一个结点的右子树不为空时，右子树的所有结点的值一定大于或者等于它的第 d 维的值。并且该节点的左右子树都是 K-D 树。对于有 N 个 K 维数据的数据集来说，K-D 树的建树时间复杂度是 $O(N\log N)$，单点查询和插入复杂度在 K-D 树较为平衡的时候是 $O(\log N)$，邻域查询时需要回溯。此外，还需要设计一些算法来保证 K-D 树的平衡，会有额外的开销。

3 点云存储技术

3.1 Mongodb 数据库

MongoDB 是一种适用于海量数据读写的非关系型数据库。它是面向文档的数据库， 由数据库、集合、文档三个层次组成[9]。

文档的键（key）和值（value）可支持多种数据类型和类型嵌套。通过文档、数组和 各种数据类型的灵活组合，可以用一条记录即一个文档表达复杂的层次关系。此外，由于没有固定的模式，添加和删除字段变得非常容易。

MongoDB 有两种存储引擎，一种是 MMAPv1，另一种是 WiredTiger。如果使用 MMAPv1 引擎，MongoDB 的数据实际上是存在硬盘中的，使用内存映射技术实现硬盘 到内存的映射，使得 MongoDB 可以直接对这块区域进行读写，减少硬盘读写开销。整个过程的内存管理由操作系统完成。如果使用 WiredTiger 引擎，写入数据时先写入内存和持久化写入 Journal 文件，每隔一段时间，可对 Journal 文件提交一个 checkpoint，将 内存中的数据变更刷新到硬盘中，利用 Journa 文件可以保证 checkpoint 之间的数据持 久化写入；WiredTiger 通过 MVCC（Multi-Version Concurrency Control）实现细粒度的文档级别并发控制，提升 MongoDB 的读写并发能力；此外，WiredTiger 还进行了数据压缩，能极大地加快数据的写入。

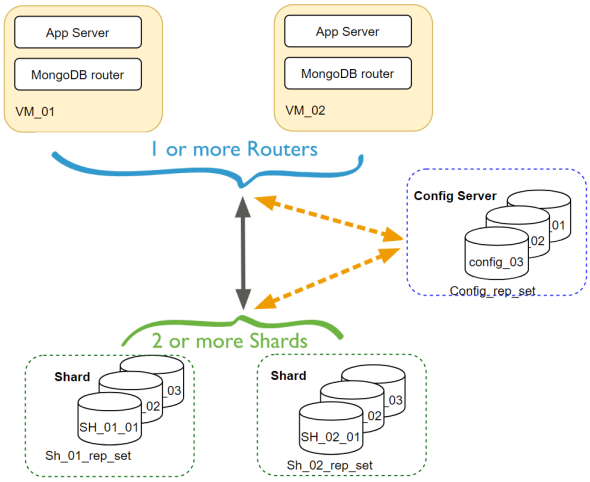

图 4 MongoDB 数据库集群
Fig.4 MongoDB Cluster

MongoDB 是一个充分利用 CPU 和内存加速的数据库， 适用于数据频繁读写的场景，可以极大地提升数据读写的效率。此外，MongoDB 支持 B 树和哈希索引，对数据建立索引能有效提升数据查询速度。同时 MongoDB 支持横向扩展，面向文档的数据类型有利于在多台服务器之间进行分割。海量机载激光点云是非结构化数据，适合使用 MongoDB 进行存储，以实现高效存取[10,11]。

3.2 HBase 分布式数据库

HBase 是 Google Bigtable[12]的开源实现，利用 HDFS 作为文件存储系统，利用 MapReduce 处理海量数据，利用 ZooKeeper 作为对应的功能。不同于一般的关系数据库，HBase 是一个适合于非结构化数据的基于列存储的数据库。

HBase 具有如下特点：

① 模块化及线性可扩展性；严格一致的读写；
② 读取速度快；
③ 表的自动分片和配置；
④ 支持 Region Server 之间的自动故障转移

⑤ 易于使用的客户端访问；
⑥ 方便基类支持 MapReduce 作业与 HBase 的表；
⑦ 布鲁姆过滤器实时查询和块缓存；
⑧ 可扩展的基于 JRuby 的脚本；
⑨ 支持监控信息通过 Hadoop 子系统导到文件或 Ganglia

HBase 数据库的结构与传统数据库的存储方式类似的地方是它都是由行和列组成的。

行与传统数据库类似，称之为行键（Row Key），用来检索记录的主键；而列则有所不同，列最大的索引方式是列族（Column Family），每一个列族下面又可以有很多列（Column），所有 Column 均以二进制格式存储，而且不同行可以有不同的列，这是与传统数据库最不一样的地方。因此 HBase 比较灵活，数据可以根据用户自己的需要自由存放在 HBase 之中。

行和列对应一个单元格（Cell），用来存储数据，称之为值（Value），它可以由行键和列一起检索出来。另外每一个单元格还有时间戳（Timestamp）作为区分，即相同单元格 不同时间戳的值还可能是不一样的。这样可以动态的显示表格中数据的更新。

灵活的 Hbase 数据库特别适合多维度的离散点云数据。此外 HBase 访问接口有多种方式，不同方式的适用场景不同，HBase 的访问接口有 Native Java API、HBase Shell、Thrift Gateway、REST Gateway、Pig 和 Hive 等几种。针对于点云数据，可以方便的利用 Thrift 和 pdal 或者 pcl 这种 C++库进行点云数据的读取、索引[13]。

4 分布式架构

以 Hadoop 、Spark 为代表的的分布式系统的出现，为处理海量数据提供了新的解决思路。分布式系统的存储能力与计算能力会以线性增长方式进行水平扩展[14]，由一台扩展到上千台，并且它的生态圈具有非常多的组件，从而用来对大数据进行计算、存储和分析，主要介绍基于 Hadoop 分布式架构的大规模点云计算方法。

4.1 Hadoop 架构

自从 2003 年 GFS[15]和 2004 年 MapReduce[16]论文的发表到 2008 年 Hadoop 成为了 Apache 的顶级开源项目，它经历了迅速的成长，多种的应用场景已经证明了它的成功、多样性与生命力。

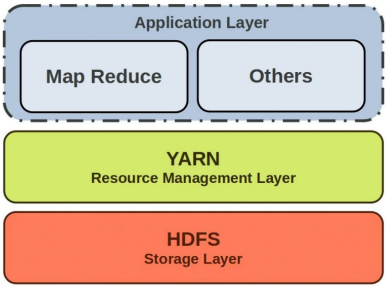

图 5 Hadoop 架构生态
Fig.5 Hadoop Structure

Hadoop 项目由众多模块组成，其内部生态圈如图 5 所示。HDFS 是 Hadoop 中数据存储管理的基础，负责管理 Hadoop 集群中每个存储节点上的文件。HDFS 具有高容错性，能够自动保存多个副本，出现故障后系统会自动检测并通过副本及时处理。HDFS 也能够实现高吞吐量的流式数据访问。其中 YARN 是任务调度和集群资源管理框架。

4.2 MapReduce 计算框架

MapReduce 是一个并行计算框架，主要由 Map 和 Reduce 两大模块组成，数据根据键值对的方式进行处理，Map 在接收到传入的一堆错杂无序的 Key-Value 数据之后，经过用户自定义解析处理完成后仍然以 Key-Value 的形式输出到 Reduce，Reduce 最终将处理完成之后的最终结果写到 HDFS 中的某个目录中。

MapReduce 的出现，把一个复杂任务成功地拆分成为了多个简单的小任务，这些小任务分别在多个计算节点上进行计算，大大提高了效率。

4.3 基于 Hadoop 的大规模点云分布式计算存储案例

基于 Hadoop 架构，Nguyen[17]等提出了 B-EagleV 一种分布式大规模点云存储渲染方案。

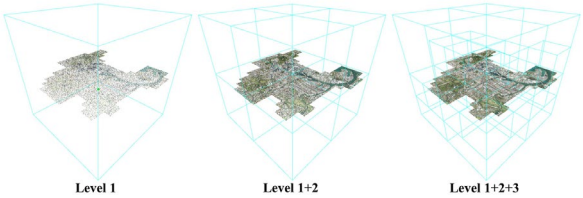

图 6 动态嵌套八叉树结构
Fig.6 Modifiable Nested Octree Structure

其空间索引采用了动态嵌套八叉树结构对离散的点云进行层次化和索引，如图 9 所示。这种改进八叉树结构能够保证点云数据从树根到节点都能均匀分布。

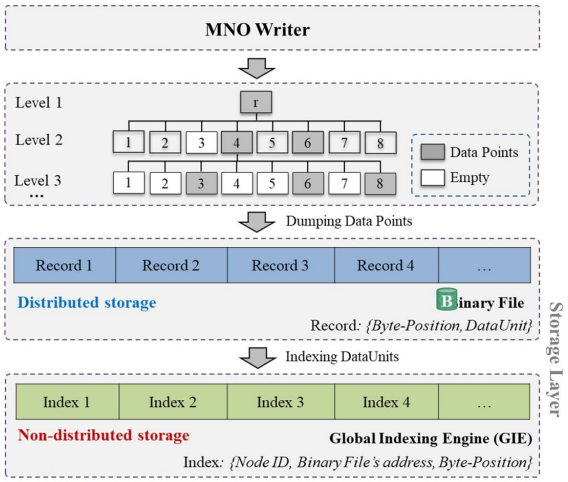

图 7 存储结构
Fig.7 Storage Structure

如图 7 所示，通过多层的索引后，将点云数据作为二进制文件存储到分布式集群中，其中根据 NodeID，二进制文件的地址和字节位置建立点云数据存储的索引。

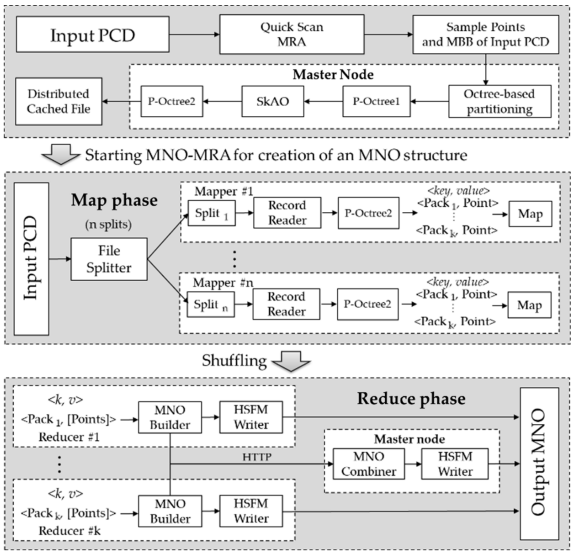

图 8 MapReduce 过程
Fig.8 MapReduce

经过存储后的点云数据片先经历 split 分块处理，其中每个 split 操作都通过一个 RecordReader 来将数据细分到 key-value 键值对，然后传入 Map 过程。其中针对于 csv、pcd 和 las 等不同类型的数据，只要是以键值对形式存在的点云数据都能够对其进行处理。

首先通过 Hadoop 框架提供的 Map 过程，把离散的数据先存进动态嵌套八叉树的节点中，接着 Reduce 过程每个 mapper 获取一个输入数据块，并根据 P-Octree2 值实现分区数据点。这样，可以将相同数量的数据点分配给 reducer，每个 reducer 使用动态嵌套八叉树构建器来生成多解数据。在此过程中，将需要合并的八叉树节点发送给 Hadoop 主节点进行合并。

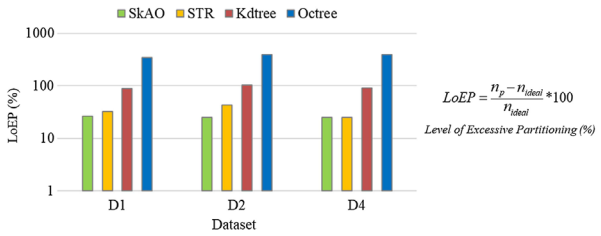

图 9 不同空间划分方式对比
Fig. 9 Spatial partitioning Methods Comparison

文中对比了不同划分方式的多余划分率，文中采用 SkAO 方法取得了较小的多于划分率，能够很好减少冗余数据。其中八叉树的多余划分率最大，因为它采用了线性的划分方式。

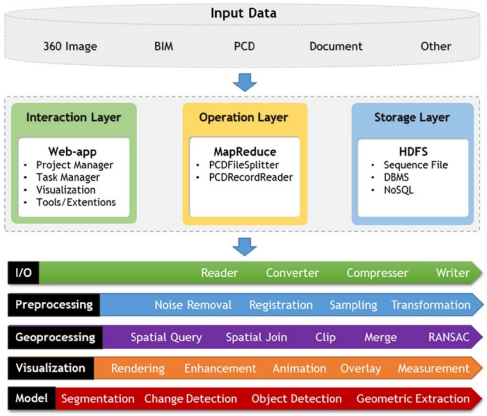

图 9 B-EagleV 系统的应用流程
Fig. 9 Full design of B-EagleV for applications in the construction industry.

B-EagleV 应用如图 9 所示，可以看到在分布式集群下，解决了点云数据的读写 I/O 瓶颈，可以对大规模点云进行批量预处理，以及空间查询、合并等操作。针对于具体的在工程建设过程中的应用，通过 Web3D 可以实现坐标测量，改造面积估算等，如图 10 所示。

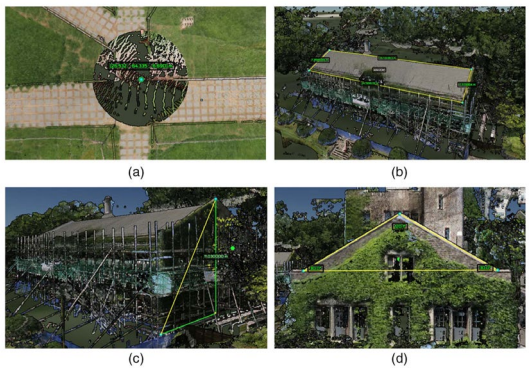

图 10 B-EagleV 系统的 Web3D 应用:（a）坐标测量;（b）改造面积估算;（c）在建房屋高度估计;（d）在建屋顶坡度计算
Fig. 10 Using B-EagleV's Web3D application for survey activities in construction: (a) measurement of coordinates; (b) estimation of area for renovation;(c) height estimation of building in construction;(d) roof slope estimation of building in construction

6 结　论

随着机载激光雷达对地观测技术的发展，机载激光雷达能够产生大量的点云数据，基于 Hadoop 等分布式系统的存储和计算架构被提出，能够更好的对离散、复杂、大量的点云数据进行高效处理和存储，避免了单个节点的运算瓶颈，很好支持了更高精度对地观测的机载激光雷达点云数据，即点云密集、范围大时也能通过增添分布式节点经济的解决处理、计算难题。

此外通过 web 服务对点云数据进行发通过用户端能够更好的利用点云数据在测量、建造、自动驾驶高精度地图等方面，也推进了大规模点云数据的使用。

参考文献：

[1] 曲瑞超. 海量机载 LiDAR 点云数据管理及可视化应用初探[D]. 中国测绘科学研究院, 2014[2022-12-02].

[2] LERONES P M, FERNÁNDEZ J L, GIL Á M, 等. A practical approach to making accurate 3D layouts of interesting cultural heritage sites through digital models[J]. Journal of Cultural Heritage, 2010, 11(1): 1-9.

[3] GIGLI G, CASAGLI N. Semi-automatic extraction of rock mass structural data from high resolution LIDAR point clouds[J]. International Journal of Rock Mechanics and Mining Sciences, 2011, 48(2): 187-198.

[4] BLAIR J B, RABINE D L, HOFTON M A. The Laser Vegetation Imaging Sensor: a medium-altitude, digitisation-only, airborne laser altimeter for mapping vegetation and topography[J]. ISPRS Journal of Photogrammetry and Remote Sensing, 1999, 54(2-3): 115-122.

[5] 方芳. 机载 LIDAR 技术现状及发展方向[J]. 2009 全国测绘科技信息交流会暨首届测绘博客征文颁奖论文集, 2009: 103-108.

[6] DUAN J, ZHAI W, CHENG C. A spatial grid index based on inverted index and its query method[C]//2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS). IEEE, 2017: 6189-6192.

[7] MEAGHER D. Geometric modeling using octree encoding[J]. Computer graphics and image processing, 1982, 19(2): 129-147.

[8] GANG Z, MAOMEI W, YI X, 等. Research on spatial index structure of massive point clouds based on hybrid tree[C]//2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA). IEEE, 2017: 134-137.

[9] BANKER K, GARRETT D, BAKKUM P, 等. MongoDB in action: covers MongoDB version 3.0[M]. Simon and Schuster, 2016.

[10] WANG W, HU Q. The Method of Cloudizing Storing Unstructured LiDAR Point Cloud Data by MongoDB[C/OL]//2014 22nd International Conference on Geoinformatics. 2014: 1-5. https://doi.org/10.1109/GEOINFORMATICS.2014.6950820.

[11] XU X, GUO R. Research on Storage and Processing of MongoDB for Laser Point Cloud under Distribution[C/OL]//2016 3rd International Conference on Materials Engineering, Manufacturing Technology and Control. Atlantis Press, 2016: 1559-1564[2022-12-02]. https://www.atlantis-press.com/proceedings/icmemtc-16/25852401.

[12] CHANG F, DEAN J, GHEMAWAT S, 等. Bigtable: A Distributed Storage System for Structured Data[J/OL]. ACM Transactions on Computer Systems, 2008, 26(2): 4:1-4:26. https://doi.org/10.1145/1365815.1365816.

[13] VO A V, KONDA N, CHAUHAN N, 等. Lessons learned with laser scanning point cloud management in Hadoop HBase[C]//Workshop of the European Group for Intelligent Computing in Engineering. Springer, 2018: 231-253.

[14] WU J. Distributed system design[M]. CRC press, 2017.

[15] GHEMAWAT S, GOBIOFF H, LEUNG S T. The Google file system[C]//Proceedings of the nineteenth ACM symposium on Operating systems principles. 2003: 29-43.

[16] DEAN J, GHEMAWAT S. MapReduce: simplified data processing on large clusters[J]. Communications of the ACM, 2008, 51(1): 107-113.

[17] NGUYEN M H, YOON S, JU S, 等. B-EagleV: Visualization of Big Point Cloud Datasets in Civil Engineering Using a Distributed Computing Solution[J/OL]. Journal of Computing in Civil Engineering, 2022, 36(3): 04022005. https://doi.org/10.1061/(ASCE)CP.1943-5487.0001021.