OTB – Onderzoek voor de gebouwde omgeving
Faculteit Bouwkunde

TOWARDS 10^{15} POINTS MANAGEMENT - AN nD POINTCLOUD APPROACH

Haicheng Liu
August, 2018
GISt Report No. 75

TU Delft Delft University of Technology

Challenge the future

TOWARDS 10^{15} POINTS MANAGEMENT – AN ND POINTCLOUD APPROACH

PhD Research Proposal

Haicheng Liu

Supervisors:
Prof.dr. Peter van Oosterom
Dr. Martijn Meijers
Ir. Edward Verbree

Co-reader:
Dr. Roderik Lindenbergh

August, 2018

OTB – Research for the built environment
Faculty of Architecture and the Built Environment
Julianalaan 134, 2628 BL Delft
Tel. (015) 278 30 05
E-mail: OTB-bk@tudelft.nl
http://www.otb.bk.tudelft.nl
ISSN: 1569-0245
ISBN: 978-90-7702-944-2

ABSTRACT

The proliferation of high accurate laser scanners and the advent of new sensors such as multi-beam echo sounders and 3D cameras prompt the production of point clouds. In parallel, arising application fields pose high requirements of computation on massive point clouds which can be more accurate than conventional raster and vector representations, taking robotics and virtual reality for example. These developments raise essential demand for smart and highly efficient data management. However, we lack effective tools. Database vendors provide state-of-the-art solutions, but still critical problems such as inefficient loading, limited operators and lack of support of continuous Level of Detail (cLoD) exist. Previous research has demonstrated the advantage of converting time or LoD in addition to spatial attributes to an organizing dimension which could then be utilized for clustering and indexing. However, a thorough validation and theory are still missing. Besides, how new computational platforms such as cloud and quantum computing could bring benefit also needs further exploration.

To address these problems, first I will devise an adapted smartPC model which guides the implementation of the data management considering new contextual environment. Then, I will implement an nD PointCloud structure which particularly focuses on the flexible yet efficient organization of various dimensions. As key components, cLoD and space filling curves in a high dimensional space will be investigated. 3 use cases will be established to test the efficiency and functionality of the new data structure. They are 4D visualization and change detection based on X/Y/Z/cLoD, 5D GPS feature extraction focusing on the additional temporal dimension and 5D indoor navigation with information of classification involved apart from X/Y/Z/cLoD. In a second phase, I will deploy the nD PointCloud structure on a cloud computing system with specified streaming, transmission and caching strategies. The overall performance will be assessed by performing a comprehensive benchmark. I will determine sensitive factors such as data size and query type that could influence the performance significantly by an initial benchmark test. Then I will conduct a full benchmark with respect to sensitive factors to evaluate as well as tune the data storage. It is expected in the end that not only should a better solution be developed, also new insights into dimensions and the LoD concept of nD point cloud could be obtained.

CONTENTS

1 PROBLEM STATEMENT 1
- 1.1 Visualization . . . 1
- 1.2 Storage of segmented results . . . 3
- 1.3 File systems and DBMS . . . 4
- 1.4 Advanced computing platforms . . . 5
- 1.5 Benchmark . . . 5

2 RESEARCH QUESTIONS 7
- 2.1 Sub-questions . . . 7
- 2.2 Scope . . . 8

3 RELATED WORK 9
- 3.1 LoD . . . 9
- 3.2 Visualization . . . 11
- 3.3 Data accessing method . . . 11
- 3.4 File based systems . . . 14
- 3.5 DBMS . . . 17
- 3.6 Hardware platforms . . . 21
- 3.7 OGC protocols . . . 23
- 3.8 Benchmark . . . 25
- 3.9 Outlook . . . 25

4 METHODOLOGY 27
- 4.1 Initial investigation and practice . . . 27
- 4.2 Construction of the conceptual model . . . 28
- 4.3 Design and implementation of an nD PointCloud structure . 31
- 4.4 Functionality development . . . 33
- 4.5 Implementation on new platforms . . . 37
- 4.6 Benchmark design . . . 40
- 4.7 Benchmark test . . . 42

5 PRACTICAL ASPECTS 43
- 5.1 Time plan . . . 43
- 5.2 Data, software and hardware . . . 44
- 5.3 Education . . . 45
- 5.4 Research visits . . . 47
- 5.5 Organization . . . 48
- 5.6 Contributions and benefits . . . 49
- 5.7 Publication goals . . . 49
- 5.8 Journals and conferences . . . 51

Bibliography 53

1 PROBLEM STATEMENT

Last decades have seen the increasingly growing production and application of point clouds. For one thing, the increase is due to the advent of all kinds of sensors for data acquisition, including airborne/terrestrial laser scanner, dense image matching, multi-beam echo sounder and 3D cameras [Otepka et al., 2013]. For another, it is a result of constantly arising application fields such as terrain modeling, forest estimation, coastal monitoring, robotic and virtual reality. Considering the repeated scans of the same area periodically, i.e. the temporal dimension, point clouds can achieve billions, even multiple trillions (10^{12}) of points [Van Oosterom et al., 2015]. For instance, AHN is a detailed elevation model of the whole Netherlands measured with Airborne Laser Scanner (ALS). Fugro among other surveying firms has been working on the second version, i.e. AHN2 from 2008 to 2013 and acquired 640 billion points [Wijga-Hoefsloot, 2012; Van Oosterom et al., 2015]. Besides, it is pre-calculated that the Dutch national mobile LIDAR dataset (streetview) collected by CycloMedia can achieve 35 trillion points in the end. Moreover, in January 2018, the National Institute of Standards and Technology (NIST) of US initialized a public safety research program of which the topic was collection of indoor point clouds [NIST, 2018]. They want to build a standard prototype for indoor point cloud models through the initiative, as point clouds may become the basis for indoor applications for the next generation.

However, faced with these application challenges, as well as the big data issue, we lack effective tools to manage point clouds to further perform efficient computation and processing. Current state-of-the-art solutions such as Oracle, PostgreSQL, PDAL, Lastools and HDF present problems including cumbersome loading process, inefficient querying processes, much development work for subsequent applications, etc. When more attributes like intensity, classification and color information are incorporated, the burden of efficient point cloud management would become even heavier. The innovations of data storage approaches and architectures are no doubt imperative.

This chapter summarizes existing problems concerned with massive point clouds management collected from practical experiments as well as literature. In the following, issues of visualization, storage of segmented point clouds, management solutions, platforms and benchmarks will be reviewed.

1.1 VISUALIZATION

In fact, both industry and academia have already experienced technical bottlenecks of visualizing large point clouds. Specifically, Fugro has established an inspiration center to introduce Virtual Reality (VR) services to users and the scene inside is built directly on massive points. However, due to limited memory capacity, the VR equipment can only store small amount of points, which confines the size of the scene. If more points were inserted,

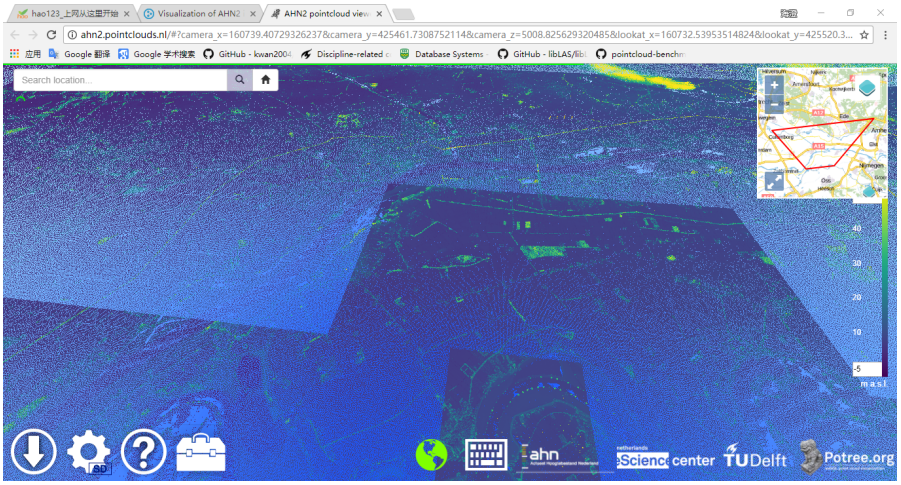

Figure 1.1: "Block pattern" of the AHN2-viewer

the VR headset would crash. In addition, when loading billions of points into prevalent point cloud viewers (Table 3.1 on page 12), these viewers all crash due to overload of memory. A LoD structure where the upper level samples a portion of data from the lower level in a data pyramid organization can be an effective solution. By adopting prevalent LoD structure such as Octree, caching can also be achieved. When zooming in, more points at lower levels can be rendered while points at original LoD outside the range will be purged out from the cache.

However, LoD has a side effect. The AHN2-viewer presents the "block pattern" [Van Oosterom et al., 2016] where sharp boundaries can be seen (Figure 1.1). This is caused by the Octree data structure used to organize the data storage. Octree is basically a *discrete* LoD (dLoD) structure, and during the rendering, due to different distance, blocks with different densities, i.e. at different layers are shown in the same scene.

To solve the problem caused by dLoD, a *continuous* LoD (cLoD) concept [Van Oosterom, 2016] may be developed to gradually visualize points at different scales, which can make the rendering process more smooth and natural. In the cLoD structure, every point is assigned an importance value indicating its ranking among the whole dataset. So there is no discrete data layers as before. Such cLoD structure plays a more critical role for the VR environment which poses high requirements of vivid simulation, i.e. real-time, extremely high accuracy and semantic enrichment. Previous researchers [Ohori et al., 2015; Van Oosterom and Meijers, 2014] proposed methods to realize LoD as a *dimension* to form the cLoD structure for 3D city modeling and map generalization respectively. Perceptually, there is no difference between dimension and attribute. They are equivalent terms. Every type of information such as sound and temperature could be perceived as one dimension for us to comprehend the world. However, in terms of storage, two types of dimensions are identified. One type is called organizing dimension which is utilized to cluster and index the data, e.g. X/Y/Z. The other is the property dimension such as color, intensity and classification which is not frequently queried. Depending on applications, these two types of dimension are interchangeable. Specifying and organizing dimensions is a crucial task for efficient data management. From this perspective, the implementation of cLoD dimension can efficiently fulfill application needs. However,

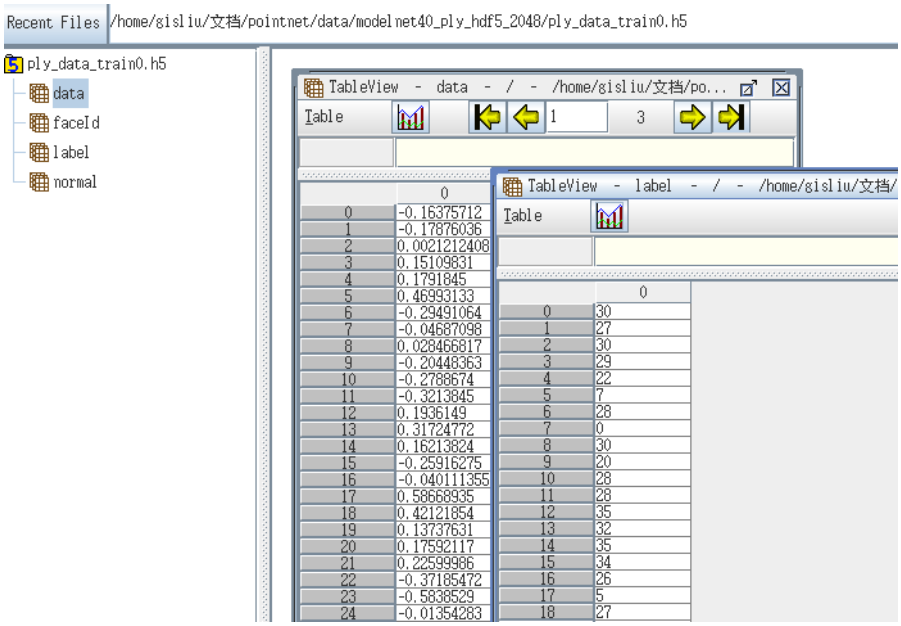

Figure 1.2: Screen shot of the HDF5 file layout, 'label' is the class identifier

calculating importance values for points to construct such a cLoD dimension remains an open question.

Besides the "block pattern", data rendering of AHN2 web service is intermittent and slow, with blocks popping up discontinuously, but the CPU and I/O of the server remains a easy mode. So the performance issue must lie in the data streaming and transmission processes. Consequently, either current protocols should be structured and combined in a more efficient way or new protocols based on TCP/IP for massive point data retrieving in a distributed environment should be developed.

1.2 STORAGE OF SEGMENTED RESULTS

Plenty research has been conducted on segmentation, classification and object identification algorithms, while seldom considers the storage of the compartmental results. The straightforward method is to add an additional property dimension for each point indicating the class identifier, which has been incorporated into the LAS file standard. In the PointNet project [Qi et al., 2017], researchers manage classification values of points into individual dataset inside HDF5 files (Figure 1.2). And one point corresponds to one value in the class array. Another commonly used method is to store all points belonging to an object into a single file with a name indicating the type together with an instance number of the specific object (e.g. chair1, chair2, classroom1, corridor1, etc.). So the whole dataset can then be decomposed into a set of individual files, which facilitates searching objects/-classes, but semantics all resides inside the file name.

Classification provides very significant semantic information of point clouds and applications can thus frequently request it. Yet it is also special as in most cases it is only constituted by a limited number of integers. Such features which could be utilized to organize the data storage is not realized in current solutions. It is suitable to be realized as an organizing dimension.

Besides, each class can contain large numbers of different instances, the house for instance. The identifier should be stored and spatial index such as R tree or Octree may then be utilized to search specific instances. Moreover, regarding such a hybrid data structure, dedicated interfaces should be developed for different query types. As an illustration, the spatial query should directly reach the spatial index, while classification query should only work with limited class members.

1.3 FILE SYSTEMS AND DBMS

Most current software for visualization and segmentation are based on files. Some are directly constructed on LAS/LAZ files, HDF files, while other vendors adopt their own formats. Additional sorting and indexing to create specific data structures have to be manually performed. With available APIs, developers may program using script languages to resolve specific tasks in a short time. However, when a new function is required, then redevelopment or additional development should be performed. Yet the scalability cannot be guaranteed. Besides, when other data types are involved, multiple formats, libraries and systems need enormous amount of effort to be integrated.

A natural solution is to adopt a database management system (DBMS) which has advantages to support multi-user access, out-of-box parallel processing, fine scalability and different data types. As a matter of fact, main vendors, Oracle for example released state-of-the-art solution for point cloud management. Two approaches are available, one is based on SDO_PC blocks and the other is utilizing Index-Organized Tables (IOT). However, a key issue for implementation is the complex data loading process as was learnt from experiments. Basically for both approaches, the data has to be first loaded into a heap table in Oracle, and then the heap table will be converted into either IOT or blocks. As is indicated by Van Oosterom et al. [2016], importing full AHN2 dataset containing 638,860,225,350 points into Oracle takes 34 hours to finish with 16 cores utilized ('Pakhuis' server, Section 5.2 on Page 44). During test and development, data loading may be repeated several times, which might take weeks before real computation is performed. Inefficient data loading is also a major reason that prevents industrial engineers from adopting a DBMS solution. As regard to the storage, table approaches which store each point as a record present significant drawbacks such as storage overhead and excessive full table scans. While block approach is mainly implemented for efficient storage but fairly time consuming to construct due to sorting. Also the operation is not as intuitive as tables, maximum selection for example.

Point Data Abstraction Library (PDAL) is a C/C++ open source library and applications for translating and processing point cloud data. Using PDAL, if point clouds are presorted, they can be directly loaded into Oracle/PostgreSQL database as blocks without the heap table process. But it also utilizes its own APIs to manipulate blocks in the databases and the query is encapsulated in a XML file together with native SQL statement, which is more complex.

1.4 ADVANCED COMPUTING PLATFORMS

To further accelerate large point clouds visualization and computation, a generic approach is to utilize advanced hardware platforms. Point clouds management in parallel is expected to benefit the whole management and query chain due to nature of the data, i.e. spatial subdivision could be easily realized on individual points. Local problems such as coordinate transformation and data loading [Van Oosterom et al., 2015] can be resolved straightforwardly. To the best of our knowledge, parallel index which is researched intensively for other spatial data types is rarely investigated for point cloud management.

Big data are commonly spread over different locations. Cloud infrastructure such as Microsoft Azure and Amazon Web Services (AWS) can be an effective approach to solve intensive computation at large scales. Firstly it can distribute computing workload and therefore owns fine scalability. Secondly, redundant versions of data can be distributed on the cloud to make application more robust. Thirdly, due to the replication, data can reside close to users, which avoids long network transmission process. However, experts from academia report the poor performance when processing large point clouds on the cloud platform. This is mainly caused by large volume of communication between clusters and limited network bandwidth. Besides, industrial engineers are unsatisfied with chaotic management of large quantities of files on the cloud. Also the mismatch between I/O and computing is gradually widening by further growth of computing capability, especially in cluster scenario with thousands of cores [Ma et al., 2015]. Multi-level hierarchical storage architecture comprised of cache, virtual shared memory, the flash memory and disk arrays of modern clusters become increasingly prevalent, but the data structure supporting such hardware environment is still missing.

As a breakthrough, quantum computing technology is expected to start a new paradigm of computation. And it has been rigorously proved that certain applications, e.g. factoring large numbers can run much faster using a quantum solution. But in the geoinformation domain, quantum related research is still vacant on both hardware and software.

1.5 BENCHMARK

In the end, to systematically assess performance of different tools as well as advancement brought by new solutions and platforms, it is essential to perform a comprehensive benchmark test. Expert consultancy and industrial experience should be referenced for designing a scientific benchmark. Aspects including query types, data size, hardware parameters and setting of data structures (e.g. block size) should be specifically addressed. Different sets of tests need to be established and repeated several times to determine the influence of each of the factors, which is a very labor-intensive work [Van Oosterom et al., 2015; Liu et al., 2016]. Also, a benchmark may not be directly used to derive an optimal solution in reality. In benchmark tests, execution of queries is repeated in a certain pattern but users have different query habits. For example, a rectangle spatial selection may be then followed by a time series extraction. As a matter of fact, cache usage of discontinuously query execution varies from the repeated case [Liu et al., 2016]. As a result, the best solution tested may actually not work efficiently as ex-

Figure 1.3: Clearance computation of railway system is a spatial operation between 3D point clouds of the environment and a simulated rail corridor box (picture from MNG)

pected. Martinez-Rubi [2015] performed initial experiments for a realistic benchmark by simulating simultaneous querying processes from multiple users. The queries were based on logs of real querying processes. But the simulation in a broader scope elaborating users' query habits is still lacking.

In addition, more complex query processes are missing in benchmarks previously, apart from spatio-temporal selections [Van Oosterom et al., 2015; Psomadaki, 2016]. Spatio-temporal join operation for example is a fundamental spatial computation and frequently takes place in spatial analysis based on vectors and rasters. Point clouds are also involved for this operation in many cases, for instance, the cross section of railway clearances (Figure 1.3) and k-Nearest-Neighbor (kNN) search involved in change detection. Perspective-view query for visualization is another essential type during the rendering process of 3D terrain models.

Besides benchmark sets, the architecture for benchmark testing is a sophisticated issue. The massive point cloud management environment could be a distributed sever-protocol-clients system where a query execution can pass through a multi-level hierarchy in a single machine and also communications between servers constitute a critical part. Additionally, in a cloud environment, other jobs run in parallel may occupy essential computational power, which hampers a convincing testing result. These processes should be monitored and minimized to some extent.

As regard to the rest of the research plan, research questions are proposed in the next chapter. They are specifically designed to address issues mentioned above. After it, I summarized the related work to provide more background knowledge and information in Chapter 3. Methodology which forms the research body is discussed in Chapter 4, elaborating possible solutions to all research questions. In the end, practical aspects of the PhD research such as time plan, data, organization and contribution are described.

2 RESEARCH QUESTIONS

To solve the problems mentioned including visualization, storage, data loading, querying, platforms and benchmarking, I will answer the following research question:

What is an optimal point cloud data structure supporting different types of applications processing 10^{15} *points from the perspective of efficiency and functionality?*

The main research question is consist of several sub-questions. Besides, complementary questions are also needed to address problems that are not covered by the main research question. They are listed in Section 2.1. The scope of the research is indicated in Section 2.2 to guarantee a success of the PhD research.

2.1 SUB-QUESTIONS

1. What use cases and datasets will be utilized for testing the newly designed solution for managing massive point clouds?

2. What elements are required in a conceptual model to incorporate various applications? What is the architecture of the conceptual model?

3. What is the role of cLoD in the nD space consisting of more dimensions other than XYZ? How to compute the cLoD value for each point?

   The cLoD value refers to the importance value. Its computation is closely related to the application. For example, temporal dimension is critical for GPS data mining, while elevation counts a lot when visualizing airborne laser data. On the whole, all cLoD values should follow a certain distribution, and considering cognition, i.e. beauty [Jiang, 2013], small cLoD values should be much less than large values. Guan et al. [2018] implemented a uniform sampling method for a dLoD structure that resulted in the same type of distribution (Figure 3.2 on Page 10).

4. To what extent can the transformation between the organizing dimension and the property dimension facilitate management and query of point clouds? How to manage dimensions of the data at the storage level?

   Space Filling Curve (SFC) is a method that can transform data in a multidimensional space into one dimensional space. However, with respect to the performance, employing the appropriate unit (e.g. meter, second) which controls the resolution and range of the data representation in the SFC is a critical issue in which a proper balance has to be determined. Besides, encoding and decoding the SFC needs time, it is not rationale to incorporate every dimension into the SFC.

5. What are the possible data loading options and their performance? How to solve the cumbersome loading problem of DBMSs, more specifically, the multi-phase loading process and intensive sorting operations inside the database?

6. What specific platforms will be utilized for accelerating point clouds management, query and smooth visualization? How to employ the data structure and an efficient caching strategy on the platform? In what perspective can quantum computing prompt the point cloud processing chain?

7. What protocols are required to link the point cloud database to viewers/software, e.g. VR, desktop client and mobile phones and what are the shortcomings of the protocols on a distributed platform? What elements should be included in a new and efficient protocol?

8. How to evaluate the performance of the data structure? What queries and processes should be elaborated in the benchmark to learn the efficiency of the data structure? How to devise the benchmark to be more comprehensive yet efficient?

   Sample queries will elaborate for instance, spatio-temporal selection, aggregation, spatial computation, perspective view selection for visualization and processes can be data loading, transmission and change detection.

2.2 SCOPE

The following aspects will be limited or excluded in this research:

- Development of advanced rendering algorithms for point clouds
- Registration and dense image matching
- Algorithms of segmentation, classification and object identification
- Benchmark on various cloud platforms

On the other hand, the research will deal with:

- Compression of point cloud data both in databases and communications through the network
- Development of interfaces to interact with the new data structure, e.g. the encoding and decoding programs

3 RELATED WORK

This chapter provides an overview of data management methods and applications concerned with point clouds. First, the concept of LoD is introduced together with its application in point cloud management. This is followed by Section 3.2 which presents solutions of visualizing point clouds through the web and a brief comparison among current point cloud viewers. Afterwards, data management concepts and solutions are reviewed, namely, data accessing method, file based systems and DBMS. Advanced platforms including cloud and quantum computing are discussed within the context of the research in Section 3.6. The protocols which are very critical for point cloud transmission in a network environment are briefly introduced in Section 3.7. An illustration of a state-of-the-art benchmark for point clouds is presented in Section 3.8. The chapter ends with a general remark on the essence of the research.

3.1 LOD

LoD is a critical factor which is always concerned with modeling. Even though use cases may vary significantly, LoD is a general way to solve the big data issue. Computation can be controlled at a certain LoD and if more details or higher accuracy are required, data at lower levels may then get involved, which normally forms a pyramid structure. LoD is also related to importance, and a small LoD value corresponds to high importance (Figure 3.1).

Quadtree is a common LoD structure for organizing 2D data [Samet, 1984]. Its 3D version, Octree structure is the state-of-the-art at present for managing point clouds. Based on it, Elseberg et al. [2013] developed a suite of efficient encoding, compressing and rendering data structures and algorithms. Besides, Xie et al. [2013] implemented a hierarchical LoD structure using random sampling to manage massive point clouds. The random ap-

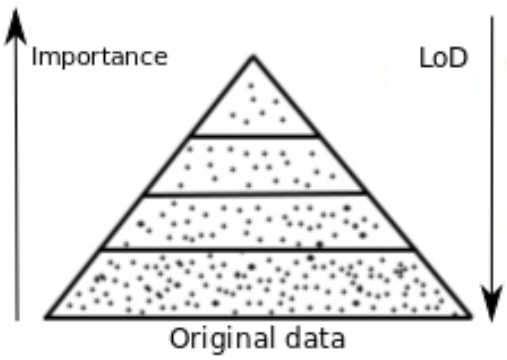

Figure 3.1: Structure of LoD pyramid

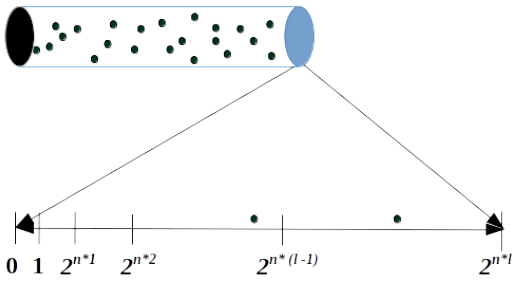

Figure 3.2: LoD value computation using random sampling, l indicates the level

Input: Z-order index	1	2	3	4	5	6	7	x
binary representation	000	001	010	011	100	101	110	111
reverse bits	000	100	010	110	001	101	011	111
after random mask $M = 101$	101	001	111	011	100	000	110	010
new binary ordering index	6	2	8	4	5	1	7	3
priority ordering index	5	2	7	3	4	1	6	x

Figure 3.3: An illustration of the priority ordering process. The first line describes the input Zordering index. There are 7 points and one dummy point designated as x [Zheng et al., 2017]

proach is based on two principles: 1. Points at the parent layer should be removed from the child layer for the sake of storage efficiency. 2. The sampling rate of the child layer depends on the number of parent nodes. Based on this structure, fine performance of interaction, neighborhood searching and dynamic updating is achieved. However, parent nodes may contain critical points, e.g. in the corner or on the boundary. Therefore, all layers might be needed for certain applications based on random sampling. Analogously, Guan et al. [2018] developed an algorithm for random LoD value generation (Figure 3.2). In the algorithm, the original datasets are treated as the bottom layer, and the points at level i+1 are sampled by a uniform random method to the upper level i. The factor between two adjacent levels can be 2^n with n the number of dimensions, i.e. one in every $2^n + 1$ points of level i+1 will be randomly selected and inserted into the level i. Totally 32 levels are defined, so it is still a dLoD approach.

Instead of direct random sampling, Cura [2016] indicated that a suitable LoD should be homogeneous in space, insensitive to density variations, regular, efficient and deterministic. So the Octree structure was utilized and the points closest to centers of cells constituted different layers of the LoD. In fact, the study expressed the cLoD idea, but the implementation was still a dLoD approach. The advantages of the LoD were demonstrated through visualization and density correction applications later. Zheng et al. [2017] proposed a complex cLoD computing method. They first sort all points in the Z-order, and then by reversing bits, taking a random mask and removing dummy points, a new priority ordering index which is actually the cLoD value can be created (Figure 3.3). This approach can be more efficient at determining representative points than normal random sampling approach according to them.

3.2 VISUALIZATION

After analyzing the shortcomings of popular solutions for point cloud visualization, e.g. insufficient support for large datasets and scalability of desktop solutions, Martinez-Rubi et al. [2015] chose to develop the AHN2-viewer[1] using Potree renderer [Schütz and Wimmer, 2015]. The Potree renderer is an implementation realized by standard web technologies including WebGL, three.js and Javascript. The input data comes from the PotreeConverter tool which is used to convert point cloud files in LAS, LAZ, PTX or PLY format to a multi-resolution Octree data structure, where each node of the Octree is stored in a LAZ file. Through the AHN2-viewer, a public web service for the AHN2 3D web visualization and downloading is provided.

Another web technology for rendering massive point clouds is the Plasio which supports the de-facto standard LAS/LAZ file format. In fact, it is a compound solution integrating Entwine [Manning, 2017], Greyhound and Plasio [Verma and Butler, 2014], which also provides a multi-resolution indexing capability. Entwine is an indexing library that uses a multi-resolution (dLoD) data structure that responds to cubic Octree queries, but packs in a similar way to a Quadtree. It is powered by PDAL. Greyhound is a RESTful HTTP server that links with the Entwine library to stream the indexed data to a client. Metadata and real point cloud data can be transferred to the clients, while the data remains in the original schema and LAZ-perf compression is applicable. This is also a promising approach which has involved many developers.

As a exercise to learn basic functions provided by current point cloud viewers, a brief comparison of viewers is conducted (Table 3.1). The hardware is a ThinkPad T550 laptop with Dual-Core Intel i5-5200U at 2.2 GHz, 12 GB of main memory, and a WIN10 operating system. The disk storage is a 1 TB SATA 7200 rpm in RAID5 configuration.The sample data used are

1. an AHN2 sample with 13,346,502 points with only X, Y, Z.
2. an AHN3 sample with 508,564,458 points with X, Y, Z, intensity, return number, GPS time and classification

These desktop viewers (except AHN2-viewer, a web viewer) present different problems of data management, mainly lack of the LoD support which can be an effective method to solve the crash issue during data loading. Besides, smart caching strategy plays a crucial role in smooth rendering.

3.3 DATA ACCESSING METHOD

To boost data access, two major strategies can be adopted. One is utilizing an indexing structure, while the other is to cluster data according to data access patterns.

3.3.1 Indexing

Yang and Huang [2014] performed a comprehensive 3D point cloud data management research. It focuses on novel indexing strategies for point clouds from two different sources, namely, airborne and terrestrial laser

1 http://ahn2.pointclouds.nl

Table 3.1: Comparison of prevalent point cloud viewers

Viewer	Dimension support	LoD support	Data loading	Smoothness of response	Memory usage
Bentley pointools view[a]	Color, intensity	No	Conversion to plt format should be performed before loading. AHN3 data loading fails as it cannot fit into memory	Smooth as all data is cached	Normal, it depends on data loaded. When there is operation, e.g. rotation, zoom, memory usage increases
AMC bridge viewer[b]	Color, intensity	Octree	During loading, the memory usage is increasing steadily, and AHN3 causes computer frozen. There is no memory crashing mechanism.	Smooth, as all data and Octree data generated are in memory	Several times larger than data itself, mainly due to additional Octree data
Fugro viewer[c]	Color, classification, intensity, sourceID, return number	No LoD, data is stored in both 2D and 3D	Data is loaded into memory gradually and slowly. It pops up a warning for shortage of memory for AHN3 sample, and loading process is frozen after a while.	2D points are rendered according to scanning order every time there is an operation, while 3D visualization keeps smooth.	2D and 3D scenes both occupy memory space. Memory only changes when 2D and 3D scenes exist simultaneously, and there is a zoom operation in the 2D layer.
Cloud compare[d]	Color, intensity	No LoD, but support Octree computation	When data is very large, say, the AHN3 sample, very slow loading, but no frozen, no crashing	Smooth as data is all cached	New operations do not change memory occupation (or very little change)
AHN2-viewer	Height, intensity, sourceID	Octree	It depends on network speed	Not smooth, point blocks are shown gradually	New operations raises memory usage which after a while decreases due to smart caching strategy
Plasio[e]	Color, intensity, classification	No	Loading is very fast. But the AHN3 sample fails immediately, and it seems that the web application can judge whether memory is capable for handling the data, i.e. pre-calculation	The software is a native application based on a web browser, i.e. no network communication. All data is loaded into the memory, so rendering is smooth.	Data is totally buffered into memory without additional data generated
SceneMark[f]	Color	No	Loading process is square tile based, i.e. loading one block after the other sequentially	Smooth and all data is cached	Hundreds of times smaller than the data itself. But the zoom operation can increase memory usage.

a https://www.bentley.com/en/products/product-line/reality-modeling-software/bentley-pointools-view
b https://www.amcbridge.com/labs/desktop-apps/point-cloud-viewer
c https://www.fugro.com/about-fugro/our-expertise/technology/fugroviewer
d http://www.danielgm.net/cc/
e http://plas.io/
f http://scenemark.com/

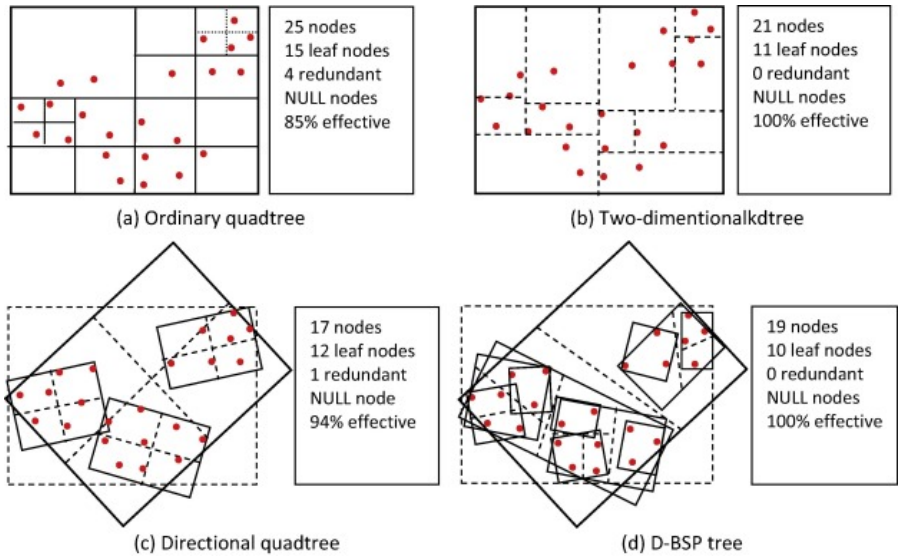

Figure 3.4: Comparison of spatial partitioning on a 2D point set [Zhang, 2016]

scanner (TLS). It is a pyramid model combining a global Quadtree index and local 3D R-trees residing in leaf nodes of the Quadtree. Airborne point clouds tend to be a 2.5 dimensional dataset where a Quardtree is capable for indexing the grid Digital Elevation Model (DEM). However, data from a TLS system are truly 3D points. Thus, the 3D R-tree is introduced. Compared with the Oracle SDO_PC approach, their solution could achieve faster data loading and querying. The disadvantage is that the solution adopts Windows file system, creating some problems in concurrent queries posed by multiple users. Wang and Guo [2012] proposed a management approach based on a combined 2D-Minimum Bounding Rectangle (MBR) and 3D-Minimum Bounding Box (MBB) indexing strategy for point clouds acquired in a single station, which is convenient for both 2D and 3D queries. A 3D R-tree is utilized for indexing root MBBs of multiple stations. As a case study, the storage and visualization of Forbidden City is then implemented. Results show that the approach is capable for efficiently managing and querying more than one billion points. Zhao et al. [2015] put forward a composed index of multi-grid and KD-tree used to efficiently perform neighborhood searching task. Within each grid/voxel, a KD-tree is constructed. The structure has the advantage of simple computation where the grid index can be determined by directly comparing the coordinates. It has lower depth compared with solo KD-tree. Thus, the local search using KD-tree could be more efficient. Final benchmark tests indicated that the speed for nearest neighbor search was significantly improved. Zhang [2016] presented a directional fuzzy c-means method for irregular spatial partitioning and finally generated a directional Binary Space Partitioning (D-BSP) tree structure (Figure 3.4) to manage point clouds. The D-BSP tree achieved lower memory resource consumption and higher speed in the frequent data accessing and rendering with testing data from millions to a billion points.

3.3.2 Clustering

Normally data stored closely on disk can be retrieved faster than data stored separately. So, in addition to the scanline order, smart SFCs are also utilized

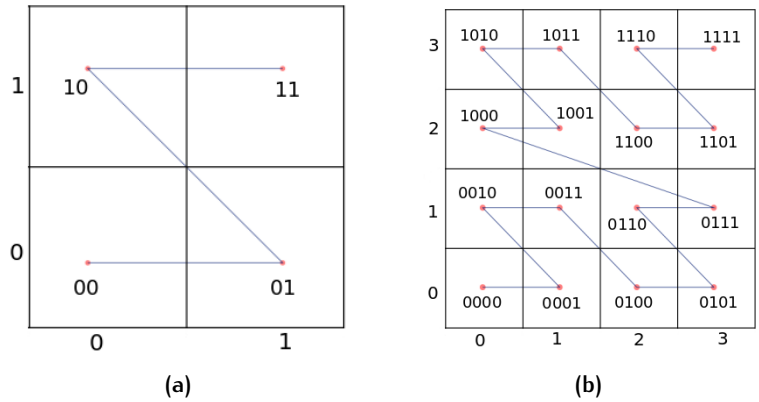

Figure 3.5: The first (a) and second order (b) of the Morton curve

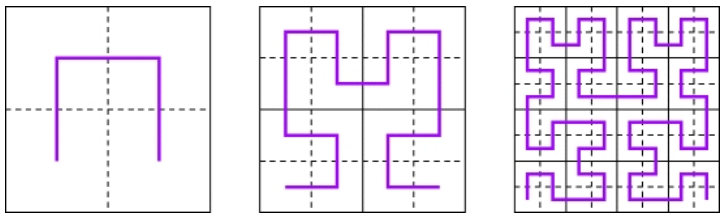

Figure 3.6: The first, second and third order of the Hilbert curve

widely for clustering spatial data in the storage [Wang et al., 2015, 2018a]. Among them, the most common orderings are the Morton (Figure 3.5) and the Hilbert curve (Figure 3.6). The Morton curve (also called Z-order or N-order curve) is based on interleaving the bits from the coordinates. For example, assuming a point with coordinates (3,2), its binary representation is (11, 10). By interleaving these bits, the Morton key 1101 can be derived, which is the 14th node on the curve (Figure 3.5). Points are then sorted according to the SFC codes to be grouped together, while their spatial relationship also retains.

Psomadaki [2016] collected requirements from coastal monitoring domain, and developed a SFC approach based on Oracle IOT for managing large dynamic point cloud data. Two treatments of Z and temporal dimension, i.e. as a property dimension or as an organizing dimension encoded in the Morton key were implemented and benchmarked. The best approach in her research, concerns an equal treatment of the spatial and temporal dimensions in the SFC. The approach also presents fine scalability. With SFC keys, a query translator should be implemented to transform normal spatial expressions into ranges of SFC code. Such a program has been developed in the SFClib [Guan et al., 2018] which encapsulated encoding, decoding and querying functions into abstract classes. Guan et al. [2018] then encoded XYZ and LoD value into a 4D Hilbert key and insert into Oracle IOT. Through benchmarking, the approach presented high query efficiency and fine scalability over massive point clouds.

3.4 FILE BASED SYSTEMS

As was indicated by Van Oosterom et al. [2015], the majority of point cloud data management is currently based on files. Among them, some formats

are vendor specific. For example, the Bentley pointool view only supports the native *plt* format so that a converter tool which transforms LAS/LAZ/TXT into the plt format is also provided. The OPALS data manager [Otepka et al., 2012] utilizes a single file to manage ALS data because it is an efficient light-weight tool tuned for high-speed processing and flexibility compared with databases. Regarding indices, the point data is first sorted into regular tiles whereas KD trees are built on the fly as a second level for each tile. Other formats are vendor neutral, including the ISPRS LAS file format, HDF and PDAL described below.

3.4.1 LAS/LAZ

The schema of LAS format [ISPRS, 2011] is composed of four parts:

1. Public header block: file source ID, project ID, LAS version, generating software, project extents, etc.
2. Variable length records: GeoKeyDirectory Tag (mainly coordinate reference system). For discrete full-waveform LIDAR data, the wave form descriptors are also included.
3. Point data records: X, Y, Z, intensity, return number, classification, channel, scan angle, GPS time, etc. X(t), Y(t) and Z(t), i.e. the shift to the first pulse received are recorded for full-waveform data.
4. Extended variable length record: waveform amplitude values for each packet.

A sound file based system also comprises the interface to interact with data. Van Oosterom et al. [2015] utilized LASsort[2] to reorder the points of each LAS file in a 2D Morton curve. Then LASindex[3] was adopted to create an index file (LAX), based on an adaptive Quadtree over the X and Y coordinates of all points, to speed up querying. The metadata (bounding box of each tile) was stored in a PostgreSQL database. LASmerge[4] and LASclip[5] were employed for querying.

LAZ is a modified LAS format with LASzip compression [Isenburg, 2013]. It employs the header and variable length records directly from LAS files. However, the real point cloud part is stored as chunks with scaling and offsetting techniques utilized for compression. LASzip decomposes different attributes into separate atomic items including POINT10, GPSTIME10, RGB12 and WAVEPACKET13 (Figure 3.7) and compresses them independently utilizing their specific features (mainly data range and interval). Depending on the input, the compression ratio can be as large as 1 000 percent (e.g. AHN2). Besides, spatial index (LAX) can be directly built, which makes LAZ a rather flexible and efficient format as the original LAS format.

3.4.2 HDF

HDF is a set of file formats (HDF4, HDF5) designed to store and organize large amounts of scientific data. In the geoscience domain, it is commonly used for storing multidimensional datasets concerned with regular spatial

2 https://rapidlasso.com/lastools/lassort/
3 https://rapidlasso.com/2012/12/03/lasindex-spatial-indexing-of-lidar-data/
4 https://rapidlasso.com/lastools/lasmerge/
5 https://rapidlasso.com/lastools/lasclip/

name of atomic item		size	point type and size					
			0	1	2	3	4	5
point attributes	format	size	20	28	26	34	57	63
POINT10		**20 bytes**	x	x	x	x	x	x
X	int	4 bytes	x	x	x	x	x	x
Y	int	4 bytes	x	x	x	x	x	x
Z	int	4 bytes	x	x	x	x	x	x
Intensity	u_short	2 bytes	x	x	x	x	x	x
Return Number	3 bits	3 bits	x	x	x	x	x	x
Number of Returns of Pulse	3 bits	3 bits	x	x	x	x	x	x
Scan Direction Flag	1 bit	1 bit	x	x	x	x	x	x
Edge of Flight Line	1 bit	1 bit	x	x	x	x	x	x
Classification	u_char	1 byte	x	x	x	x	x	x
Scan Angle Rank	u_char	1 byte	x	x	x	x	x	x
User Data	u_char	1 byte	x	x	x	x	x	x
Point Source ID	u_short	2 bytes	x	x	x	x	x	x
GPSTIME10		**8 bytes**		x		x	x	x
GPS Time	double	8 bytes		x		x	x	x
RGB12		**6 bytes**			x	x		x
Red	u_short	2 bytes			x	x		x
Green	u_short	2 bytes			x	x		x
Blue	u_short	2 bytes			x	x		x
WAVEPACKET13		**29 bytes**					x	x
Wave Packet Descriptor Index	u_char	1 byte					x	x
Bytes Offset to Waveform Data	u_int64	8 bytes					x	x
Waveform Packet Size in Bytes	u_int	4 bytes					x	x
Return Point Waveform Location	float	4 bytes					x	x
X(t)	float	4 bytes					x	x
Y(t)	float	4 bytes					x	x
Z(t)	float	4 bytes					x	x

Figure 3.7: Groups of attributes in LAZ [Isenburg, 2013]

grids, e.g. global precipitation rasters. Nowadays, in some geo-modeling processes, LIDAR data starts to replace rasters for the sake of high accuracy and as a consequence, and HDF format is being studied for such an adaptation. The National Geospatial-Intelligence Agency (NGA) of USA published a Sensor Independent Point Cloud (SIPC) standard [NGA, 2015] which identified the data and metadata generated by LIDAR systems and provided a framework for handling the resultant point cloud data processed from a single LIDAR sensor and ready for analyst exploitation. The implementation was based on HDF5 format. According to this standard, the positions of points, i.e. XYZ will be treated as a dataset in the HDF5 file in parallel with other dimensions like intensity and time. Such an implementation undermines the advantage of efficient data access through regular spatial grids of HDF5. For each spatial query, XYZ values in the HDF5 file have to be checked completely. Yet HDF5 has no support for a spatial indexing structure.

3.4.3 PDAL

PDAL is a software for managing point clouds and it functions in general as an abstraction layer on management solutions including LAS files, Oracle and PostgreSQL DBMS. Thus the same operations are available independently on which system actually contains the data. It has a set of commands that can be used to create a stand-alone file-based solution, thus offering an alternative to LAStools, as is investigated by Van Oosterom et al. [2016].

If the schema is not specifically defined, PDAL will utilize its default DBWriter to create blocks inside a DBMS, e.g. Oracle. Inside the block, scaling and offset are also utilized to transform the actual positions into

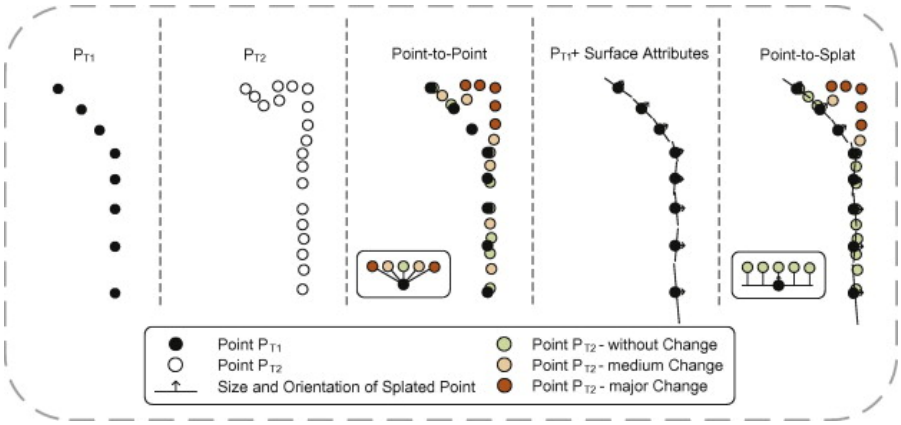

Figure 3.8: A Point-to-Point and Point-to-Splat change detection approach to update a database [Richter and Döllner, 2014]

32-bit integers for the storage. LAZ-perf compression is available for block storage and the compression ratio is similar to rapidlasso LAZ, around 15x smaller than a LAS file.

3.5 DBMS

DBMS is the preferable way for enterprise data management due to its properties of Atomicity, Consistency, Isolation and Durability (ACID), while in the GIS domain, its scalability, parallelization, efficient caching, interoperability of different data types and multi-user support are highly valued. However, a debate on whether DBMS should be specific or generic has been existing for a long time. On the one hand, specific customization like data structure and functionality can improve efficiency. On the other hand, DBMS should be generic to facilitate the management of heterogeneous data which could otherwise be fragmented. With respect to both aspects, plenty research and development have been done towards point cloud management.

For 3D city modeling, Richter and Döllner [2014] introduced a change detection approach (Figure 3.8) to reduce storage in case of incremental updates in the same area. It was implemented in a database. Basically, points that are identified as new surfaces will be inserted into the database, while new points locating on existing surfaces will not be added. In order to analyze surface changes in high mountain environments, Rieg et al. [2014] proposed a database approach with tiling strategies for efficiently managing multi-temporal point clouds. They also presented an overview of current point clouds management approaches incorporating file-based systems, databases and hybrid systems.

Besides, general studies on models of point cloud management are also published. Dobos et al. [2014] illustrated the shortcomings of relational database model for managing large point clouds. Then based on use cases of astronomy, they proposed a model of point cloud database including basic requirements and algorithms that should be implemented. Such a database is expected to work with large amounts of disk-resident data and scale out to multiple servers. Cura et al. [2017] presented a state-of-the-art point cloud management based on a PostgreSQL server to provide services.

The solution utilized patches to group point clouds and incorporated metadata management. Topologic framework was proposed to be constructed on patches to take advantage of graph analysis. Besides, several in-base point cloud processing functions such as clustering points using minimum spanning tree, extracting primitives (plans and cylinder), etc were prototyped and implemented. Vo [2017] also developed an Oracle solution, UMG_PC with a customized 2D-3D hybrid indexing structure. The top layer was a 2D Hilbert-coded, rectangular grid, and the bottom layer was consist of separate, in-memory, 3D Octree for each point block. Basic operators were realized in Java classes which then were registered with an Oracle DBMS as a package. 3 novel functionalities including the point clipping, the nearest neighbour search, and the planar segment selection were further implemented based on this scheme. The later two functions were concerned with inner block computation which was not available using Oracle SDO_PC, while the point clipping exploiting the new structure can be 4. 4 to 34 times faster than the SDO_PC solution. It also consumed less disc space and was created more rapidly.

3.5.1 Oracle

Oracle is an object-relational DBMS with the tables model. A spatial module is also available in Oracle for management of geometric types, using BLOB for large objects. There are flat table and block approaches.

Flat table approach

Several possibilities exist:

1. Normal table. Every point is stored as a record with X, Y, Z and other property dimensions stored using the NUMBER type. B tree index can be created on one or multiple columns. However, as each column has its favorable order of records, the flat table approach is not sufficient for different types of queries. Besides, it suffers excessive full table scans. Nevertheless, the operations are straightforward and intuitive.

2. Spatial indexed table. Every point is still stored as a record but with spatial attributes X, Y, Z stored as a geometry type. R tree index can be built on the geometry column. The approach could be more efficient for spatial queries. However, it needs additional decoding process to retrieve raw coordinates, which could take much time when dataset is very large. Also, the R tree index can occupy large storage space on the disk.

3. IOT (Figure 3.9). The selected organizing dimensions are encoded into a SFC key (e.g. Morton key in Figure 3.5) which is the primary key. Property dimensions remain normal storage. In this approach, no extra indexing structure is needed, primary key data and non-key column data stored within the same B* Tree structure as the leaves. Changes to the table data, for example, adding new rows, or updating or deleting existing rows, result only in updating the index. The downside is that this approach also needs decoding to get original value. In addition, the query process is based on the range of the SFC key instead of original dimensions, which implies a translator which converts multidimensional ranges into one dimensional range is required inside the database.

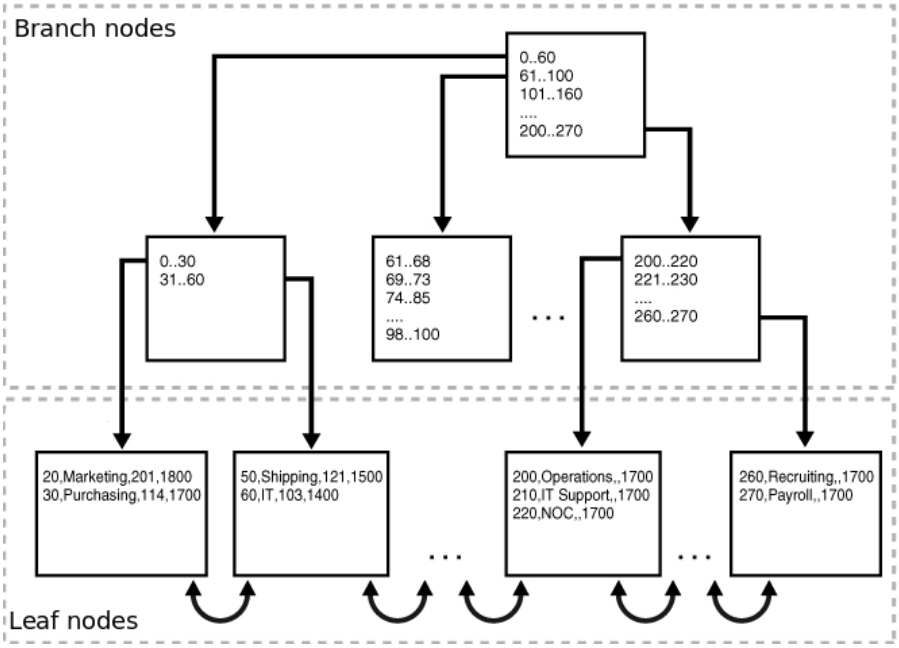

Figure 3.9: The structure of an index-organized departments table. The leaf nodes contain the rows of the table, ordered sequentially by the primary key which constitutes the branch node. For example, the first value in the first leaf node shows a department ID of 20. The upper layers contain the index and leaf nodes are linked as well

Block approach

The block approach also utilizes tables to organize the whole data model, but the core difference lies in the employment of blocks as the basic unit for operations. Blocks also allow compressing a group of points. Several options are available:

1. Native SDO_PC block (Figure 3.10). Two tables will be created in this approach. One is the BASE table recording the SDO_PC object and an identifier. Point cloud data is grouped as a set of blocks in another BLK_TABLE. Several fields are elaborated:
   - OBJ_ID, the identifier corresponding to the SDO_PC object in the BASE table.
   - BLK_ID, ID number of the block.
   - BLK_EXTENT, spatial extent of each block.
   - PCBLK_MIN_RES, the minimum resolution level at which the block is visible in a query, i.e. a query can define the resolution range with PCBLK_MIN_RES and the following PCBLK_MAX_RES.
   - PCBLK_MAX_RES, the maximum resolution level at which the block is visible in a query.
   - NUM_POINTS, total number of points in a block.
   - NUM_UNSORTED_POINTS, the number of unsorted points in the POINTS BLOB.
   - PT_SORT_DIM, Number of the dimension on which the points are sorted.

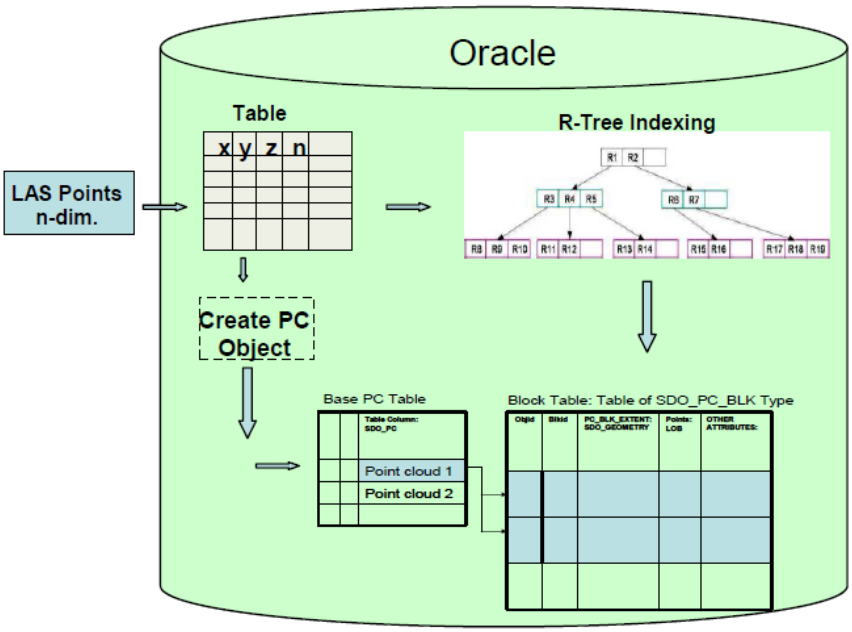

Figure 3.10: Storage Model for SDO_PC type [Ravada et al., 2010]

- POINTS, the BLOB containing a block of points. It is in fact a pointer linked with the BLOB object. In the BLOB, points are sorted and stored. Each point records additional metadata including PC_TOT_DIMENSIONS, BLK_ID and PT_ID.

2. PDAL block. This approach creates the same table structures as the SDO_PC method through the API of PDAL. However, the BLOB object created adopts PDAL's default schema which is not yet compatible with Oracle operators (Section 3.4.3). Laz-perf compression is supported.

3. SFC block. Analogous to the IOT approach, a consecutive set of SFC keys containing selected dimensions can be grouped into a block. Then, either a flat table or IOT is utilized for holding the blocks.

3.5.2 PostgreSQL

The storage strategies of PostgreSQL are analogous to Oracle, except the following:

1. IOT approach does not exist, but it is possible to cluster tables based on an index. However, this leads to two separate structures: one is the table, and the other is the index.

2. The block approach is from a point cloud extension developed by Ramsey [2014]. It stores each block in a row of a table using the PC_PATCH binary data type. A block records an ID and a list of points containing all dimensions (Figure 3.11). Blocks themselves will be stored in a TOAST table, a feature of PostgreSQL that separates and stores large objects. Besides, vast set of functions to manipulate the blocks are provided in the extension.

```
{"pcid":1,"pts":[
 [-126.99,45.01,1,0],[-126.98,45.02,2,0],[-126.97,45.03,3,0],
 [-126.96,45.04,4,0],[-126.95,45.05,5,0],[-126.94,45.06,6,0],
 [-126.93,45.07,7,0],[-126.92,45.08,8,0],[-126.91,45.09,9,0]
]}
```


Figure 3.11: A sample of the PC_PATCH data with 4 dimensions, X, Y, Z and intensity expressed in plain text

3. PDAL utilizes the point cloud extension of PostgreSQL for reading and writing blocks.

3.6 HARDWARE PLATFORMS

A general way to solve big data issue is to apply advanced hardware platforms with more computing power and storage. In this research, I will also explore advancement brought by new platforms, specifically the cloud service which integrates High-Performance Computing (HPC), parallelization and distributed computing technology. But only using application oriented protocols can a client exchange data with on the cloud platform. Additionally, current quantum computing technology is regarded as a revolution of the computation, which is nearly untouched in the realm of geomatics.

3.6.1 Cloud

As was indicated by Wang et al. [2018a], based on virtualization, scientists can make use of cloud computing which drastically improves the computing efficiency with powerful computing resources. These cloud systems are currently being constructed on HPC clusters in a wide range. Through cloud platforms, scientists can readily customize their HPC environment and access huge computing infrastructures in the cloud which becomes an ideal platform for managing massive distributed data. For example, Washington DC Office of the Chief Technology Officer (OCTO) and AWS have made LiDAR data publicly available as an AWS Public Dataset[6] as part of the Smarter DC initiative.

Several leading companies such as Amazon and Microsoft launched commercial cloud services of which the performance has been specifically investigated. Agarwal and Prasad [2012] assessed the MS Azure storage performance by establishing and performing the azurebench on three storage services, namely Table, Blob, and Queues. They then reported the basic performance data from a series of experiments. They tested insert, query, update and delete using the Table approach with each instance processed 500 entities. It turned out that update was the most time consuming operation and query took the least time. Besides, the time taken for all of the four operations increased drastically with increasing number of worker role instances. Also, through comparison, they indicated that the Queue storage scaled better than the Table storage as the number of workers growed. Akioka and Muraoka [2010] executed standard benchmark tests including the HPL [Petitet et al., 2004] and NAS Parallel Benchmarks [Bailey et al.,

6 https://aws.amazon.com/cn/blogs/publicsector/lidar%2Ddata%2Dfor%2Dwashington%2Ddc%2Dis%2Davailable%2Das%2Dan%2Daws%2Dpublic%2Ddataset/

1991] on Amazon Elastic Computing Cloud (EC2) to provide advice for normal users. HPL is a benchmark program which resolves linear equations by Gaussian elimination with partial pivoting, while NAS is a widely adopted benchmark suite elaborating floating point performance, data communication, matrix multiplication, etc. They indicated several problems:

- Instable communication performance. NAS Parallel Benchmark results present the substantial impact of communication has on overall performance of applications.
- Obscure policy of virtual machine management. Excess number of instances can be allocated to a physical machine, which disrupts the performance of applications.
- The uncontrollable topology of instances. Whenever a user launches a new set of instances, the topology starts to change.
- Dynamic communication bandwidth. The bandwidth is possibly shared with other users, and even inter-node communication can be regarded as public communication. This feature prevents tuning of applications with inter-node communications.

In parallel, Jackson et al. [2010] also implemented a benchmark using real applications representing the workload at a typical supercomputing center. They pointed out that conventional HPC platforms were much faster than the Amazon EC2. The main bottleneck lay in the poor performance of network communication. These issues in fact hamper an efficient implementation of point cloud services on the cloud platform, which has also been imparted by industrial experts. However, as was mentioned, cloud is a publicly available technique to tackle distributed big data problems and it incorporates HPC ability. Thus, in the research, I will pay special attention to these issues to find the solution.

3.6.2 Quantum computer

The development of actual quantum computers is still in its infancy, and experiments of quantum computational operations were executed on a very small number of qubits. For instance, in 2016, IBM launched the Quantum Experience (QX) [Alsina and Latorre, 2016] which enabled people connecting to IBM's quantum processors via the IBM Cloud, to run algorithms and experiments. The processor behind currently controls 16 qubits. In the end of 2017, Microsoft released the quantum development kit, i.e. Q#, a quantum-focused programming language with native type, operators, and other abstraction. Q# features rich integration with Visual Studio and VS Code and interoperability with the Python programming language. It is also accessible through Azure for more powerful quantum simulation. Meanwhile, Intel and Qutech started to test the new 17-qubit chip invented very recently, while at the start of 2018, Intel revealed a 49-qubit quantum chip. However, fast production of large scale quantum computers does not entail great gain in practical implementation. In fact, it is still a long way before chips can be utilized for industrial applications. Nonetheless, some specific problems, for example the integer factorization and searching a database can be solved more quickly with large-scale quantum computers than any classical computers, which have been theoretically proven.

Architecture of a quantum database has been investigated. Roy et al. [2013] proposed a quantum database plan where quantum algorithms were

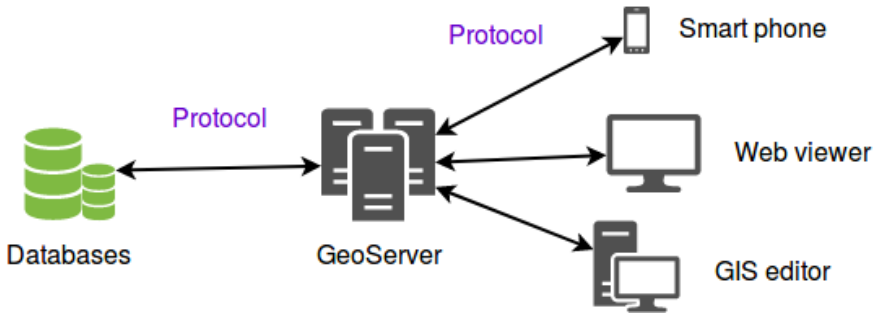

Figure 3.12: The workflow of point cloud services in a network environment

utilized for computation while the real data storage still resided in a MySQL database. Resource allocation has been improved using the new approach. Regarding basic operations on quantum databases, Younes [2007] initialized a set of quantum algorithms and suggestions to implement INSERT, UPDATE, DELETE, SELECT, backing up and restoring a database file. However, the storage model based on quantum computing is still missing. The main bottleneck lies in the short duration of a superposition of a qubit, i.e. change of state in hundreds of milliseconds due to energy loss and entanglements.

3.7 OGC PROTOCOLS

Protocol is an indispensable element to connect different platforms for point cloud services (Figure 3.12). It incorporates specifications on data models and communication. OGC has already published several standards which facilitate web spatial services. Among them, Web Processing Service (WPS), Web Coverage Service (WCS), Web Feature Service (WFS) and Web 3D Service (W3DS) present applicability in point cloud related services.

3.7.1 WPS

WPS [Müller and Pross, 2015] provides rules for standardizing requests and responses for invoking spatial processing (calculation) services, such as polygon overlay, as a web service. It defines an interface that facilitates the publishing of spatial processes and clients' discovery of and binding to those processes. WPS supports simultaneous exposure of processes via HTTP GET, HTTP POST, and SOAP [Box et al., 2000] for clients to choose. Five functions are defined in WPS:

- *GetCapabilities*: returns information about capabilities and processes offered by the server.
- *DescribeProcess*: returns detailed metadata on selected processes offered by a server
- *Execute*: executes a process comprised of a process identifier, the desired data inputs and the desired output formats
- *GetStatus*: returns status information of a processing job
- *GetResult*: returns the results of a processing job

3.7.2 WCS

WCS [Baumann, 2010] supports retrieval of spatial data as "coverages" (i.e. multidimensional array) – that is, digital spatial information representing space/time-varying phenomena. WCS provides access to coverage data in forms that are useful for client-side rendering, as input into scientific models, and for other clients. The WCS suite is organized as a core and a set of extensions defining additional functionality. The core establishes basic spatial and temporal extraction. Two types of subsetting can be combined: *Trimming* extracts a sub-area of a coverage indicated by a bounding box; the result has the same dimension as the original coverage. *Slicing* performs a cut at the position indicated, thereby reducing the dimension of the result coverage. Three operations are supported:

- *GetCapabilities*: returns an XML-encoded description of service properties and the data holdings offered by the server inquired
- *DescribeCoverage*: returns XML-encoded descriptions of coverages (such as their location in space and time)
- *GetCoverage*: delivers a coverage (or part thereof), either as original data or processed, in some suitable data format

3.7.3 WFS

WFS [Vretanos, 2010] represents a change in the way vector data is created, modified and exchanged on the Internet. Rather than sharing geographic information at the file level using File Transfer Protocol (FTP), for example, the WFS offers direct fine-grained access to geographic information at the feature and feature property level. Web feature services allow clients to only retrieve or modify the data they are seeking, rather than retrieving a file that contains the data they are seeking and possibly much more. It specifies 5 class of operations:

- *GetCapabilities and DescribeFeatureType*: returns the capability of the service and retrieves the application schema that defines the feature types that the service offers
- *GetPropertyValue and GetFeature*: allows features or values of feature properties to be retrieved from the underlying data store based upon constraints, defined by the client, on feature properties
- *GetFeatureWithLock and LockFeature*: provides exclusive access to features for the purpose of modifying or deleting features
- *Transaction*: allows features to be created, changed, replaced and deleted from the underlying data store
- *CreateStoredQuery, DropStoredQuery, ListStoredQueries and DescribeStoredQueries*: allows clients to create, drop, list and described parameterized query expressions that are stored by the server and can be repeatedly invoked using different parameter values

3.7.4 W3DS

W3DS [Schilling and Kolbe, 2010] is a portrayal service for three-dimensional geodata, delivering graphical elements from a given geographical area. 3D

scene graphs will be rendered by the client and can interactively be explored by the user. The W3DS merges different types (layers) of 3D data in one scene graph which does not contain the semantic characteristics and relations of the basic geodata. The default format it applies is X3D. Five operations are available:

- *GetCapabilities*: returns service metadata from a server
- *GetScene*: returns a 3D scene representing a subset of the natural or man-made structures on the earth surface
- *GetFeatureInfo*: provides clients with additional attribute information about features within a scene that is currently displayed
- *GetLayerInfo*: collects information on the available attribute names and the values in the attribute table of a specific layer
- *GetTile*: accesses tiled layers using tile level, row, and column indices

To summarize, WPS is suitable for processing services such as noise removal and various spatial computation on point clouds. WCS can be employed for spatio-temporal selections. Since point clouds can be regarded as a vector data (point type), WFS may be applied for selections based on any dimensions. Visualization of point clouds as images can be realized in a way through the W3DS protocol. However, none of these provide specific rules for efficient operations on massive point clouds, especially compression and streaming standards. The streaming mechanism defines the priority of data transmission and it is required in the perspective view selection, for example. The visualization based on images definitely removes lots of original information of point cloud data which becomes exceptional due to its high accuracy and thus it is not acceptable.

3.8 BENCHMARK

Van Oosterom et al. [2015] designed and implemented a comprehensive benchmark for large point clouds management. The testing dataset AHN2 elaborates totally 640 billion points and 12 TB size in LAS files. Various platforms and data organizing approaches were taken into account including PostgreSQL flat table, PostgreSQL block, Oracle flat table, Oracle block, Oracle Exadata, MonetDB and LAStools. SFC was later utilized in the flat table model and the effect was analyzed. To investigate the scalability, the benchmark was decomposed into several stages with different data sizes, i.e. mini (20 million points), medium (20 billion points), and full benchmark (640 billion points). Besides, two parallel query processing algorithms were presented and partly tested to learn the performance gain.

3.9 OUTLOOK

We are entering into the era of big data which is challenging on the one hand, while on the other hand, could cause a paradigm shift in nearly every domain. It will soon become the reality that driverless cars are moving around by scanning millions of points per second. It should be noted that big data not only refers to the data, it also defines the applications/use. This

is a crucial aspect to consider when designing new data structures as well as benchmarks. The scale of the problem is directly related to the basic unit (e.g. block size) to organize the data.

Besides, researchers from other realm indicated potential directions which also catered to the interest of this research:

- Cognition. Previous data storage addresses the efficiency and the design originates from an engineer's perspective. This somehow caused the "block pattern" (Figure 1.1 on Page 2). The cLoD proposed in the research would take the cognition/perception of points rendering into account by adopting appropriate distribution. Besides, although we have been working on spatial data for decades, the knowledge of different dimensions is rarely defined explicitly to be utilized in the data management. Nature of different dimensions needs to be explored and identified.

- New hardware platforms. The I/O bottleneck of current computer architecture constrains the efficiency for a long time and effective solutions are still required. Thus, I proposed the cloud computing which provides a scalable approach considering distributed data collection and manipulation, and the quantum computing which could leverage the parallel nature of quantum to facilitate data processing. In the research, I will investigate further what we could improve using them.

4 METHODOLOGY

The whole research project can be divided into 6 components (Figure 4.1), which are discussed specifically in following sections. While the corresponding time plan is provided in Figure 5.1 on Page 43.

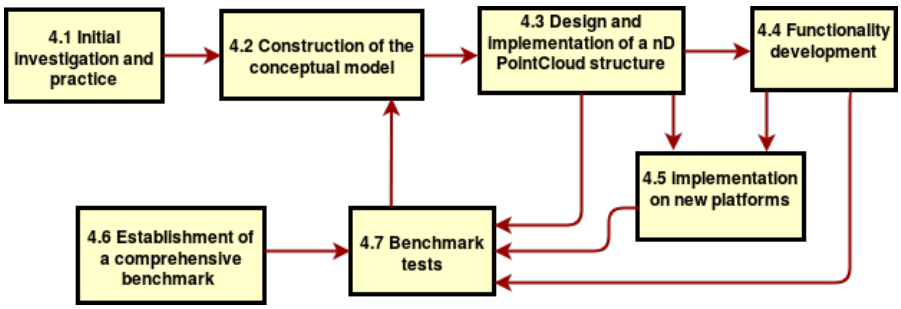

Figure 4.1: Hierarchy of sub-projects of the research

4.1 INITIAL INVESTIGATION AND PRACTICE

Prior investigation has to be conducted to study current scientific and technological progress related to large point clouds management, which helps formulating key research directions. Considering all components, the investigation (theoretical/literature study) includes:

1. Representative point cloud datasets that should be utilized in the research.
2. State-of-the-art point cloud DBMSs and file systems such as Oracle, PostgreSQL, HDF5 and Lastools:
   a) Storage structure and data access methods
   b) Previous benchmarks executed
3. Data loading techniques of high performance DBMSs and using professional software elaborating FME, PDAL and entwine.
4. Literature study and comparison of potential cloud platforms including MS Azure and AWS.

Besides, also gaining practical experience is crucial to comprehend specific problems as well as get familiar with related tools.

1. Initial benchmark of various solutions including DBMSs and file systems.
2. Oracle realization of the storage for AHN2-viewer which is currently constructed on LAS files.
3. Initial VR experiments using small datasets.

4. Cloud platform testing, focusing on mismatch between computation power and I/O, working principle of server-protocol-clients and multi-hierarchical architecture.

4.2 CONSTRUCTION OF THE CONCEPTUAL MODEL

Conceptual model can be generic to support vast range of applications such as WaterML which is devised for exchange of in-situ hydrological observations [Almoradie et al., 2013]. It could also be specific, like indoorGML intended for indoor localization and navigation purposes [Lee et al., 2014]. In this research, I will develop a model that can benefit visualization, VR simulation, change detection and GPS data analysis based on large point clouds. In fact, Poux et al. [2017] have proposed a SmartPC model that strives for point cloud management and processing. It is a 3-layer model (Figure 4.2) consisting of a bottom data description layer (level-0), a middle connection layer (level-1, Figure 4.3) and an upper domain adaption layer (level-2, Figure 4.4).

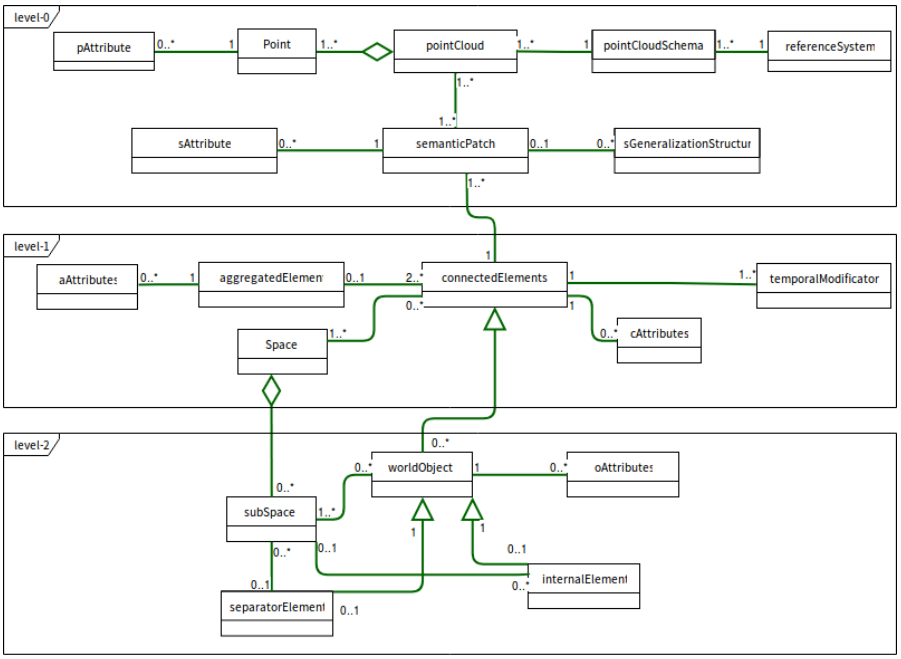

Figure 4.2: Overall SmartPC model

The conceptual model in this research will be adapted from the SmartPC model. The original level-0 layer can always be changed as it deals with storage implementation which depends on specific solutions, which can be elaborated in the nD PointCloud structure. As SmartPC model assumes that the point cloud has classification dimension which is not the general case, the *semanticPatch* class in level-0 will be changed to the *spatiotemporalPatch*. As the point cloud may be the GPS trajectories, a *timeSeries* can be added besides the *Space* class in the level-1 layer. Besides, more application modules will be added into the level-2 layer. Before this step, 3 different use cases should be implemented to specify relationships between data and functions.

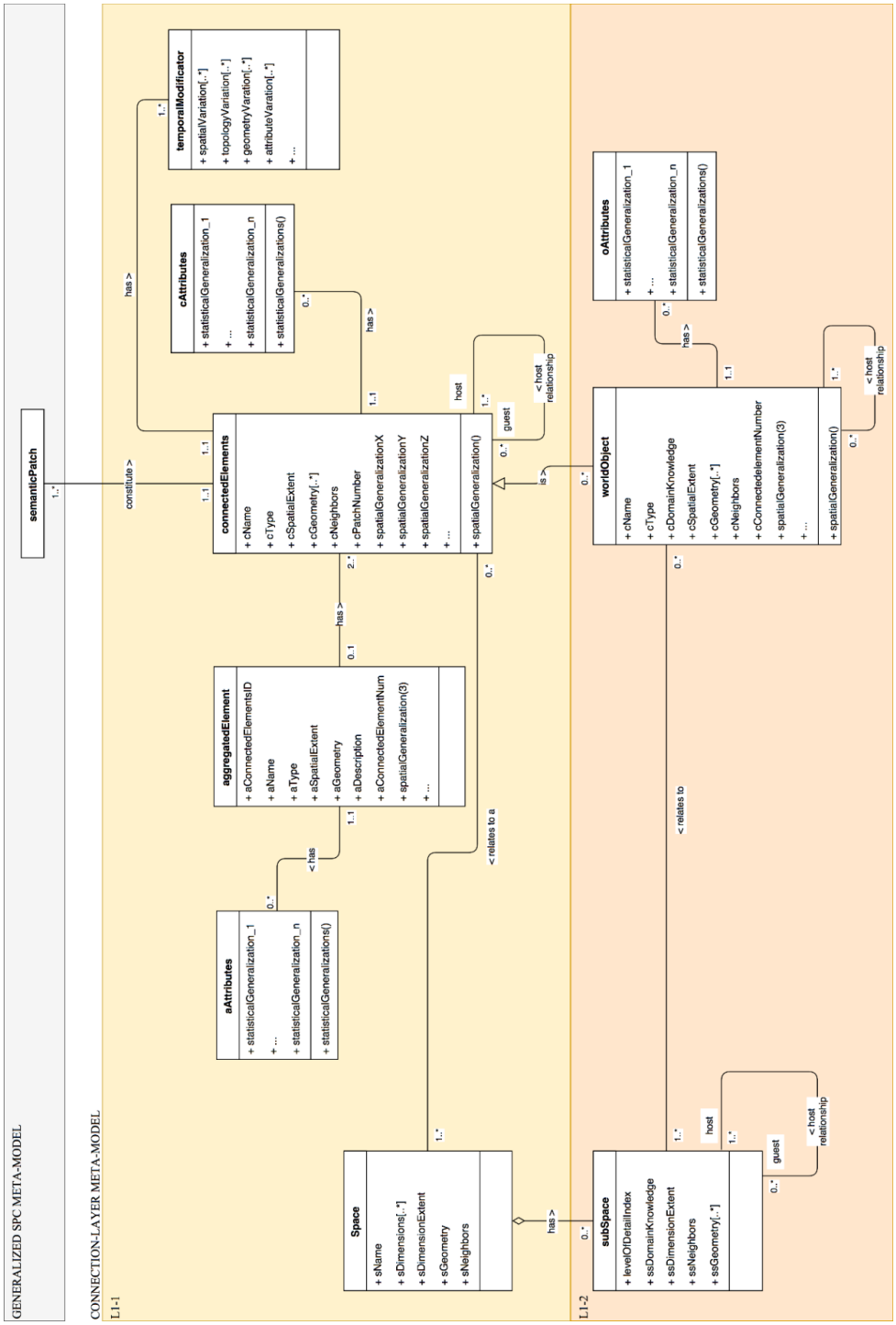

Figure 4.3: SmartPC model [Poux et al., 2017]: Level-1, data connection layer

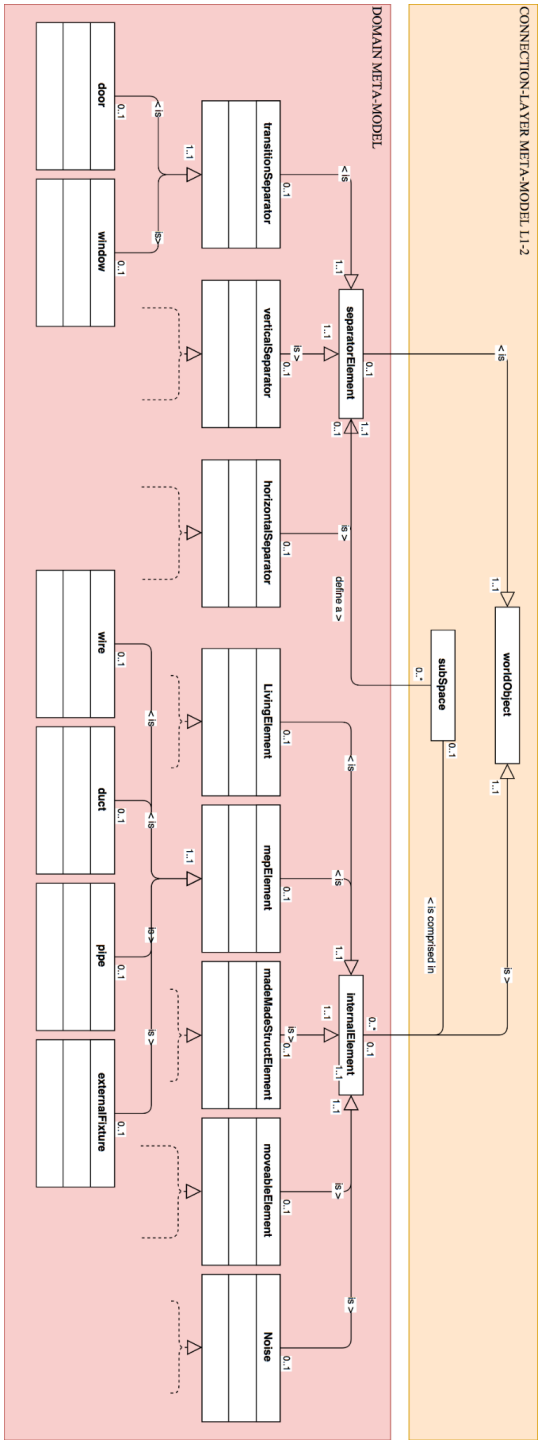

Figure 4.4: The SmartPC model [Poux et al., 2017]: Level-2, domain adaption layer

4.3 DESIGN AND IMPLEMENTATION OF AN ND POINTCLOUD STRUCTURE

As was discussed in the problem statement, a new data structure can be a way to solve the problem of inefficient point cloud management and processing. In this research, I will design and implement such an nD PointCloud structure with the cLoD dimension utilized. As was mentioned in Section 1.1 on Page 2, how to determine and use the organizing dimensions and the property dimensions for management is a key issue. Ahead of this, the nature of different dimensions should be identified. For example, basic understanding includes:

1. Traditional spatial dimensions XYZ are still the fundamental dimensions to organize data as point clouds are mostly used for spatial analysis. The values are confined in a limited scope with certain redundancy, e.g. several points may share a same X value. The most common unit is meter.

2. Temporal dimension becomes increasingly important to grasp changes of objects. It is a dynamic dimension, i.e. the range can be changed and unlimited, and is closely concerned with data updating. As it is also involved in spatio-temporal queries, the priority could be equal or above spatial dimensions.

3. LoD is an additionally introduced dimension used to express importance. LoD influences computing accuracy and efficiency significantly, e.g. classification at different scales. cLoD is consisted of a series of distinct values following a certain distribution. Considering perception [Jiang, 2013], points having small LoD values should be much less than points with large values (Figure 3.1 on Page 9). The access pattern of the cLoD dimension is normally a selection of a continuous portion, e.g. selecting points larger than some value. As one dataset may be utilized for multiple applications, there might exist several LoDs. Therefore it is a fairly flexible dimension.

4. Classification/instance dimension, as was introduced in the problem statement, should be established for semantic analytical purposes. It is represented by a limited number of integers or words. Besides, instance is closely related to spatial dimensions and each physical object such as desk and cup occupies a certain space.

5. Color and intensity are normally considered as property dimensions. But when certain applications frequently query on them, it is reasonable to convert some to the organizing dimensions to cluster or index data for the sake of efficiency.

Psomadaki [2016] proposed to map certain property dimensions to organizing dimensions depending on the context. The researcher conducted experiments, i.e. either put Z and time as part of a SFC key for management of coastal monitoring dataset (Section 3.3.2). But actually two treatments of Z tends to be stress tests as benchmark queries are not concerned with Z. Besides, the effect of multiple transformations between two types of dimensions cannot be revealed. The researcher also indicated the problem of SFC encoding that it required all dimension enrolled having similar range and interval. As a matter of fact, units of temporal dimension could variate a

lot from spatio-dimensions, not to mention the cLoD and other dimensions. Considering nD characteristic of point clouds, Guan et al. [2018] developed the SFClib[1]) applying for massive multi-scale nD point cloud management. The library provides SFC encoding/decoding, pipeline bulk loading and nD box queries.

For calculating the cLoD value, on the one hand, I will consider the geometric information and this has been addressed by the SFC code which implicitly forms an Octree structure (Figure 4.7). On the other hand, other dimensions as well as perceptual aspects should be elaborated according to various needs. An additional cLoD dimension may then be explicitly created for this. I will then construct the generic nD PointCloud structure by selecting appropriate organizing dimensions and encoding them in a SFC key. This is based on sufficient exploration of different use cases. To achieve an equilibrium in the SFC key, proper unit of each dimension during encoding should be employed. In the next step, to take advantage of the intuitive table model as well as the B-tree like structure of Oracle IOT, the SFC key together with other property dimensions will be loaded into an IOT using the SFC key as the primary key. Dimensions encoded in the key will not be restored additionally as this could cost heavily when data size is very large.

Analogous to the UMG_PC implemented by Vo [2017], the SFC key related operations including loading, indexing and computing will be realized in a registered package of Oracle, preferably in C++ classes which could be more efficient for computation. The SFClib will be further adapted and improved in the new scheme to reach the best performance. Besides, parallelization could be applied for pipelining the process. As was pointed out by Cura [2016], typical point cloud processing workflow normally handle points as a group instead of individual, which is reasonable considering popular applications such as visualization and change detection. I share the assertion. Therefore, after successfully building the full SFC key approach, I will also integrate points at a certain level to increase the efficiency.

The aim of the data structure is to elaborate full dimensions that are currently concerned in queries into the design process. PDAL solution provides such a list[2] for example. However transformation from a property dimension into an organizing dimension entails a tremendous reorganizing process (e.g. sorting and blocking). And as current knowledge and theory for an efficient solution are still limited, the data structure will be developed considering different use cases which are featured in specific dimensions. These use cases elaborate major types of point clouds:

- Case 1 (5D): Spatial dimensions, the LoD and the classification dimension are involved and they are most likely to be implemented as organizing dimensions. In this use case, semantics are intensively used. Indoor navigation and visualization will be the main application. Another option could be the visibility detection (Figure 4.5) inside buildings, where rays can pass through windows and space, but would be blocked by walls. Such a process can be integrated into a comprehensive VR simulation. Depending on time, prototypes may be developed.

1 http://sfclib.github.io/
2 https://www.pdal.io/dimensions.html

Figure 4.5: Faculty of architecture in TU Delft. Through one window, the square and another window, people and furnitures in the other part of the building can be seen

- Case 2 (4D): spatial dimensions XYZ and the LoD dimension are included. The data management of the AHN2-viewer and VR rendering will be the applications. Besides, normal spatial selections and spatial join computations like minimum clearance outline (Figure 1.3 on Page 6) and change detection will be tested as well. They constitute main applications for this research. In the implementation, different strategies to organize the data will be tested. For instance, XYLoD could be the organizing dimensions if the shape of the data is 2.5D.

- Case 3 (5D): Spatio-temporal dimensions XYZT and the LoD dimension are incorporated. Feature extraction from GPS trajectories will be applied. Data can be collected from mobile devices or vessels. Analogous to previous cases, different organizing dimensions will be experimented.

4.4 FUNCTIONALITY DEVELOPMENT

The performance of the nD PointCloud data structure will be verified and demonstrated by various functionalities. After the determination of the data structure, data loading is the first step to start all workflows and this is described in Section 4.4.1. Then, functionalities enrolled in different use cases including web-viewer rendering, VR visualization, change detection, GPS track analysis and indoor navigation will be presented from Section 4.4.2 to Section 4.4.6.

4.4.1 Data loading

As is indicated by Van Oosterom et al. [2016], although the Oracle loading process of large point clouds takes 34 hours to accomplish, the speed tends to be raw I/O with optimization. Three options are applicable:

1. Oracle external tables[3]. As was analyzed in Section 1.3, the main issue with Oracle SQL*Loader is the staging table process before final IOT tables or blocks. This can be avoided by harnessing the *external table* which allows Oracle to access data stored in files as if it were in a

3 https://docs.oracle.com/cd/B19306_01/server.102/b14215/et_concepts.htm

table in the database. Then the external table can be directly inserted into the IOT. However, binary spatial data formats like LAS/LAZ have not been supported by the Oracle external table until now. Hence, all point cloud files should first be converted to the Text format.

2. PDAL. PDAL avoids staging table by directly creating blocks outside Oracle and then storing them inside Oracle, which might be more efficient.

3. Multi-phase approach. "Is it necessary to load complete dataset into the database all at once?" This is a critical question and perhaps the whole data importing process can be divided into multiple phases. With cLoD, points of high importance should be loaded into the database first. So it is always possible to stop at a certain stage and users can still do query or computation at some LoD of point clouds. Along with the loading process, a log file which should be automatically created in the database records the cLoD completed each time. This approach is even more appropriate for dynamic data growing over time.

All these options have their shortcomings. External tables only support Text files. PDAL option needs data to be pre-sorted before loading. Yet it entails a whole data management solution based on PDAL blocks. The multi-phase approach requires prior LoD computation. Thus, in different cases, the best solution may differ. In the research, comprehensive benchmark concerning diverse applications will be executed to learn the performance of these alternatives.

4.4.2 Rendering of the AHN2-viewer

As AHN data is collected by aircrafts, a natural way to model the cLoD is to simulate the view from sky to ground gradually. That is to say, in the roughest scale, only large objects can be seen while more details appear when increasing the cLoD. However, the size of an objects is related to XY, which is a global computation problem and not a character of a single point. Nonetheless, a random sampling approach for cLoD can resolve the issue as large objects mostly have more points. Besides, decision makers might hold a different view that they may want every specific object shown even if it is a small house for example. Then the solution should be first segmenting and clustering the data. Then cLoD is computed for each object and finally all points are integrated together again.

In summary, the general methodology is as follows:

1. Simulate the behaviour of LoD (e.g. large objects appear at the highest level).

2. Determine essential information that can realize a cLoD. Current AHN2-viewer renders the AHN2 dataset with only XYZ information. If more information like color, intensity and classification is enrolled in the data, the cLoD can be calculated more comprehensively.

3. Organize data in a way that can efficiently support the cLoD application. At this stage, organizing dimensions should be distinguished for clustering in the nD PointCloud structure.

Then in the final rendering process, perspective view selection which has already been implemented in the AHN2-viewer [Van Oosterom et al., 2015]

will be performed to determine the range of data to be selected. In addition, the relationship between the cLoD value and the distance should be established for visualizing points. An initial algorithm could be linearly interpolating the cLoD values from view position to the farthest point which has the largest cLoD value. The program then selects points with cLoD values approximately equal the values interpolated.

4.4.3 VR visualization

The nD PointCloud structure which realizes cLoD dimension will be adopted for data management. Based on cLoD where every point represents a level, points are supposed to be rendered gradually and smoothly when movements happen. Additionally, since the memory of the headset is limited and real-time rendering is required, streaming, restructuring, encoding/decoding processes from the server to the headset should be specifically developed. An algorithm of motion prediction (Figure 4.6) which is utilized for buffering surrounding data according to current viewpoint, motion direction and pace will be adopted. Another function VR provides is the relocation operation which triggers shift of the scene. Three modes according to the distance, i.e. overview (high in the sky), medium (low in the sky) and local scale (inside terrain) require different data selections. In the overview mode, data with low cLoD values should be shown gradually without visual artifacts all the way until the destination has been reached; with respect to the medium mode, the principle is similar, but starting with medium cLoD values, i.e. more details; while shift of local view is basically a new spatial selection. In the research, the overview mode will be implemented combined with the motion prediction. Section 5.2 provides more information of the development environment of VR.

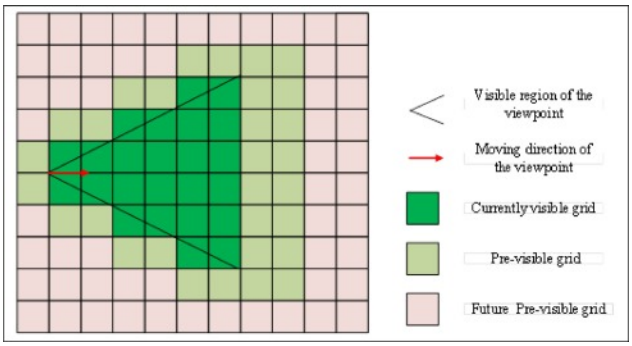

Figure 4.6: The change law of visible area in a grid demonstration [Wang et al., 2018b]

4.4.4 Change detection

The core of change detection is a kNN search problem, so it is a spatial join operation. Clearance filtering (Section 1.5 on Page 6) is a spatial join computation between points and a polyhedron. In the research, change detection is chosen as a former step before combining different data types. It is performed by comparing two point clouds of the same area but two independent time steps. To determine whether a change takes place accurately, the same area should be confined first and then, surfaces would be derived from point clouds for comparison [Qin et al., 2016]. However, this method concerns massive computation and thus I will simply calculate

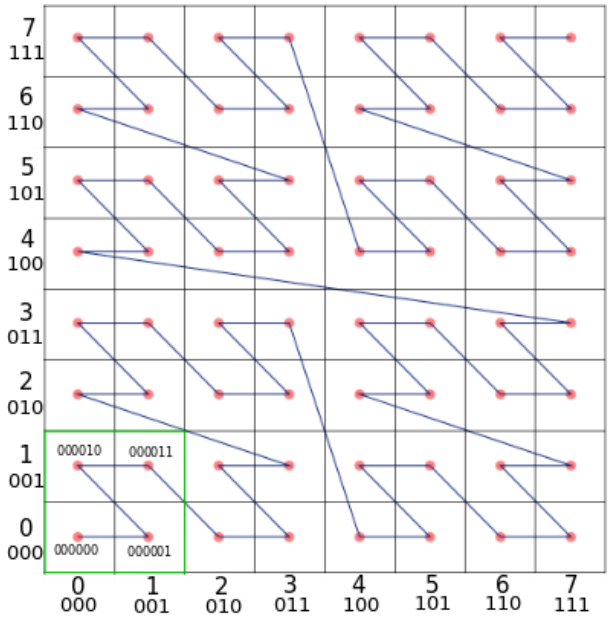

Figure 4.7: Morton code. The lower left block composed of four points which all start with 0000 will be in a same group.

the Euclidean nearest distance of the compared point to the reference point neighbor. If the distance exceeds a certain threshold, it is considered as a change point.

Using cLoD, the Euclidean NN search approach can be more efficient. That is, points at the deepest level will not be utilized for comparison first because a certain change can be reflected in the points with smaller cLoD values. With less points searched, the efficiency of NN search is improved. In fact, a dLoD approach is capable enough for the task. First, XYZ will be encoded into a Morton/Hilbert key, then by bitwise operation, points with the same first several pairs of digits of their keys will be regarded as a group (Figure 4.7). And the first point in the group will be assigned a large dLoD value. By controlling the number of digits, different layers can be constructed. The essence is that an Octree structure is embedded in the SFC implicitly. Besides efficiency, the accuracy of the change detection will be considered as a key factor which reversely determines the computation of dLoD values. In the implementation, sample data from AHN2 and AHN3 will be the input.

4.4.5 GPS track analysis

Applications of GPS tracks mainly focused on mining features, such as properties of roads [Biljecki et al., 2013; Van Winden et al., 2016] and historical trajectories [Meijers et al., 2017]. In this module, two datasets will be adopted due to different track patterns. Based on the novel data structure, I will perform feature extraction from vehicle GPS trajectories and they are average speed and importance of a road. The algorithms have been developed by Van Winden et al. [2016]. And the GPS dataset is preprocessed to match roads from OpenStreetMap [Biljecki et al., 2013]. Another dataset is the Automated Identification System (AIS) vessel tracking data from Rijk-

swaterstaat [Meijers et al., 2017]. Extraction of historical trajectories will be tested.

The average speed is calculated by selecting all the speeds of GPS points of a certain road and dividing them by the total number of GPS points. And the importance of a road is the number of passes which is counted when a vehicle travels over a certain road. To search the historical path of a vessel, first the vessel should be identified and then, all the points recording its historical locations should be selected and sorted according to the timestamps. Depending on temporal scales, different levels of these queries can be set up.

The temporal dimension will be treated as an organizing dimension and a property dimension respectively. The performance will then be evaluated. However, the cLoD concept in such an environment is fairly abstract and the rationale for defining the importance of a GPS point is lacking. Two possibilities seems applicable at present:

- Sampling of the trajectory. For instance, according to the time duration, points with long stay will be given high importance value.

- Union of similar trajectories. Several trajectories may share the same path, and the central one may be put into a high level. Isolated trajectories are less significant.

In the implementation, the GPS point cloud will be shrunk from only spatial, temporal and the combined spatio-temporal aspect. This implies that the neighbourhood relationship changes as well. The queries will be reprocessed using the simplified dataset and the distinction from the original dataset will be analyzed to get insights.

4.4.6 indoor navigation

Key needs and problems concerned with semantic objects will be analyzed to provide hints for the storage of classified indoor point clouds. To start with, I will explore the optimal data storage for visualization of the indoor environment considering navigation needs. Not only will the geometry of objects be considered, also the accessibility of objects, e.g. doors, stairs and windows will be incorporated in the navigation mode. That is to say, when less details are needed, accessible objects should be rendered first as they are more crucial for navigation. These aspects will be expressed in the cLoD computation. Besides, the visibility detection mentioned earlier is also related to the indoor navigation, and computation based on line of sight could be developed depending on the time consumption of the research.

4.5 IMPLEMENTATION ON NEW PLATFORMS

As is indicated by Wang et al. [2018a], in large-scale remote sensing (RS) applications, regional or even global coverage multidimensional datasets are used for processing to meet the growing demands for more accurate and up-to-date information. The high accurate, dynamic and informative point cloud will also be one of the major big RS datasets. During the data collecting process, point clouds are normally stored and managed in local data centers, which complexes the implementation of large-scale applications. To

conquer the issues of massive data volumes as well as the increasing complexity of data, employment of the cluster-based HPC paradigm or the next generation computing platform - quantum computer is imperative. Current cloud technology has generalized the complex HPC processing model and is accessible by large numbers of users. So the local implementation of use cases will be transplanted on a cloud system to handle big data practically. However, quantum computer is still at a developing phase at present and even in the coming years. Nonetheless, advanced quantum computing simulators as well as available quantum computing resources on the Internet makes prototype development of use cases possible.

4.5.1 Cloud

By referencing previous work [Wang et al., 2018a; Richter and Döllner, 2014], and specifically considering all use cases, I will apply the following server-protocol-clients architecture (Figure 4.8) in the research. As was mentioned, regarding wide distribution of big data, requirement of robustness of services as well as less network transmission on the Internet, cloud computing is a solution. I plan to use the Azure platform because my group has conducted research on AHN2 management on it before[4]. Azure is also one of the leading cloud services worldwide.

Data layer

Data stored are various point clouds collected for different purposes. In this research, they particularly refer to airborne laser scanning data, indoor point clouds and GPS tracks. These datasets will be managed using the nD PointCloud structure where cLoD is applied. However, data storage with traditional block and table approaches will also be created for benchmarking later (Section 4.7).

Function layer

The data management module is mainly used by administrators and developers to load, store, maintain and benchmark data stores. The preprocess is concerned with noise removal and metadata management, which are not the focus yet there are many efficient solutions for this work [e.g. Cura et al., 2017]. The data loading approach developed in Section 4.4.1 will be deployed to utilize parallel acceleration. Both dimension managing and cLoD computation are used to construct the nD PointCloud structure where optimization of storage can be made. Besides, system/query logs from the monitoring module can also be utilized to design a benchmark and optimize the data storage later.

The analysis module contains all functionalities needed for use cases proposed in this research. Spatio-temporal computation refers to property extraction from GPS tracking data (Section 4.4.5), while generalized spatio-temporal queries and analysis functions are also included. Complete AHN2 and AHN3 datasets are planned to be compared for the change detection.

The visualization poses higher requirements than other analysis jobs as it needs to be performed in real time. Limited network bandwidth as well as large data volume confines the massive data streaming approach. The cLoD plays a crucial role therein. As in visualization applications, view

4 http://www.pointclouds.nl/

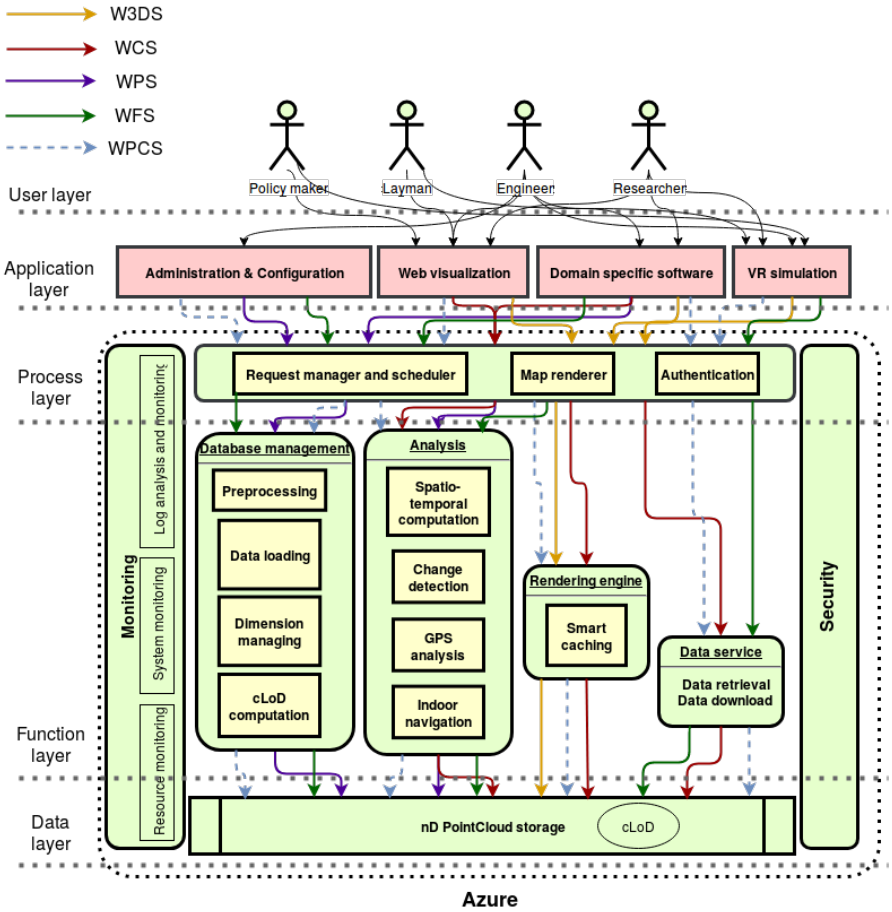

Figure 4.8: Implementation of use cases on the cloud, adapted from Richter and Döllner [2014]

perspective constantly changes, so smart caching plan is a key component to improve efficiency and smoothness. Parameters received from the map renderer which records motion of the user in the process layer will be utilized for extracting scenes as well as buffering neighbouring data (Figure 4.6). Then combined with cLoD values, points needed and mostly likely requested later will be cached. The cache cleaning should also be efficient to avoid memory crash. Regarding this point, the AHN2-viewer has already implemented a decent solution which can be referenced. AHN2 and street-view scanning data from Fugro will be tested.

Protocols

Different protocols are utilized in the communication among layers (Figure 4.8). But still they present limitations for such a relatively new data type. Visualization supported by W3DS can only be synthesized images, which implies information loss during data retrieving. 3D visualization based on raw point clouds is lacking. Also there is no specific compression, streaming, structuring (e.g. blocking) and LoD definitions in the protocols available, while those elements are indispensable for handling massive points. Communities, e.g. [Kodde, 2010] have proposed to establish a new OGC standard, the Web Point Cloud Service (WPCS) which is expected to take account of the special aspects of point clouds. In this research, key points and bottlenecks that need to be improved or resolved will be figured out to assess the necessity of publishing WPCS.

4.5.2 Quantum Computer

As I/O bottleneck of current computer architecture is hardly to break through, new types of computers might be a solution. Recent research [Wiebe et al., 2015] shows quantum k-means clustering can scale as $O(\sqrt{n}\log n)$. And the kNN method is actually the counterpart of k-means clustering as is implemented by the Fast Library for Approximate Nearest Neighbor[5] (FLANN). In the research, the algorithm proposed is decided to be repeated on a quantum computer emulator using Q#. In the later phase, a simple prototype of quantum solution which utilizes part of qubits for storing point clouds while other qubits for computation will be developed on the emulator for proof-of-concept. This may be eventually a problem-defining research and also because of lack of comparable quantum computers, the implementation will get lower priority in this research.

4.6 BENCHMARK DESIGN

There are two goals of the benchmarking process. The first is to evaluate whether the nD PointCloud structure approach can achieve comparable or superior performance than current state-of-the-art approaches. With theoretical analysis discussed in previous sections, the new approach is most likely to be more efficient. Then I will enter the next phase, tuning the nD PointCloud structure to make the solution optimal. To begin with, factors affect the query (including complex computations) performance are defined (Table 4.1).

5 https://www.cs.ubc.ca/research/flann/

Table 4.1: List of factors influencing the query performance

Category	Parameter
Data	Volume
Data structure	SFC type, dimensions, attributes, index, block size, compression
Query	Type (i.e.which dimensions or attributes are concerned), dimension span (e.g.1 $\times$ 100 or 10 $\times$ 10), output size
Software	Type, parallelization
Computer hardware	I/O, memory size, CPU
Environment	number of users, network speed, protocol type

Regarding the query type, as is discussed, a more comprehensive and realistic benchmark is expected. As a matter of fact, the query log is a valuable resource which records specifically the users' query habits. In web services, there have already been studies on this [Islamaj Dogan et al., 2009]. Query records [Van Oosterom et al., 2016] of the AHN2-viewer are available for building the visualization benchmark. However a full benchmark should include spatio-temporal selections, data loading, updating and spatial joins additionally. Multi-user accessing scenarios will also be elaborated. Previous benchmark [Van Oosterom et al., 2015, 2016] will be utilized and updated in terms of use cases.

From Table 4.1, there are many factors which have to be benchmarked to determine the effect. That is to say, a large amount of tests have to be carried out. To decrease the workload, the benchmark will be decomposed into a primary version and a full version.

4.6.1 Primary benchmark

The primary benchmark is also a local benchmark which is performed in the development phase of the nD PointCloud structure, and also for the initial comparison among different solutions. Some of the factors listed in Table 4.1 are sensitive, which means by altering the value slightly, the query performance will fluctuate dramatically, while the impact of other parameters may not be so significant. Through literature study and expert consultancy, interesting and controllable parameters will be selected for benchmark tests using the sample datasets described in Section 5.2. At this stage, the SFC type, dimension setting, compression, query type, dimension span in the query, software type and number of users will be prioritized for testing. The primary benchmark set will include spatial-temporal selections at different scales, certain aggregations and the kNN search. Every time only one factor changes while others remain their values. It is expected that after this step, sensitive factors will be identified. Besides, initial comparing results among different solutions will be produced. Essentially, the scalability will also be exploited by scaling up the number of points enrolled by factor 1 000, i.e. 1 000, 1 000 000, 1 000 000 000 etc, until the full benchmark is achieved.

4.6.2 Full benchmark

In the full benchmark, complete point cloud datasets will be loaded and tested. Besides, the query types as mentioned before should be more com-

prehensive. Only sensitive factors from primary benchmarking will be tuned to locate the optimal solution. Scenarios of different number of users will be incorporated to learn the scalability.

4.7 BENCHMARK TEST

Newly designed nD PointCloud structure will be constantly benchmarked to learn the actual performance, until the structuring scheme achieves an optimal state. Then a formal comparison among different solutions will be conducted by implementing the primary benchmark locally. After it, cloud technology will be utilized to process much larger datasets for the sake of scalability. The nD PointCloud structure in the cloud will be investigated and tuned through the full benchmark.

4.7.1 Local tests

Architecture of local tests will be constructed on the 'pakhuis' server (Section 5.2). Targets for benchmarking are the DBMS solution with newly implemented nD Point Cloud structure, conventional DBMSs such as Oracle IOT and block approaches, PDAL solutions built on top of Oracle and PostgreSQL, Lasfile and HDF5 solutions. It is essential to distinguish time consumption at each component during a query execution, e.g. I/O and call of libraries, which is crucial for the improvement. Besides, caching mechanism will be monitored specifically as it is a complex process influencing query performance significantly [Liu et al., 2016; Martinez-Rubi, 2015]. It also determines whether the data structure and algorithms behind are "smart" and scalable. The overall evaluation of performance of various solutions will be based on the wall-clock time where interventions of different queries executed by multiple users will be elaborated.

4.7.2 Cloud tests

After the realization of use cases on the cloud platform (Figure 4.8), the performance of these applications will be benchmarked. The aim is to learn the shortcomings of the implementation with focus on the data storage. Thus, as mentioned previously, different components involved in a query execution will be clearly separated for time measurements. Whenever necessary, essential tools will be developed for this purpose. VR with Unreal (having a plugin supporting point clouds) installed will be included in the test to explore the potential of a Software as a Service (SaaS) for the public. In addition, protocols including WFS, WPS and W3DS will be assessed in terms of efficiency and accuracy, which might provide indications for a new protocol. Besides, the performance of bulk loading/preprocessing is also a critical aspect which facilitates querying later on. The configuration of the cloud platform will play a significant role in the benchmarking tests and thus should also be considered.

5 PRACTICAL ASPECTS

The whole time plan of the research is depicted in Figure 5.1. This is then followed by research facilities including data, software and hardware in Section 5.2. Education and possible research visits in parallel are provided in Section 5.3 and Section 5.4 respectively. Organization matters are mentioned in Section 5.5. Possible contributions of this innovative research are indicated in Section 5.6. Publications define the milestones for the research and they are summarized in Section 5.7 and Section 5.8.

5.1 TIME PLAN

Figure 5.1: Time plan of the PhD research

The core of the research is the nD PointCloud structure (blue box in Figure 5.1). Therefore, the implementation of the data structure using the 3 use cases holds the priority in this research. Then cloud computing platform

will be utilized and tested. I will propose the comprehensive benchmark, but may not be able to fully implement it due to time limitation.

5.2 DATA, SOFTWARE AND HARDWARE

Core datasets (Table 5.1) used in the research include:

- AHN2 and AHN3, the Dutch national digital terrain models, used for visualization and change detection
- Stanford Large-Scale 3D Indoor Spaces Dataset (S3DIS) [Armeni et al., 2016], used for the VR simulation and visibility detection
- OSM GPS tracing data and AIS message from vessels [Meijers et al., 2017], used for feature extraction from GPS data
- Street-view LIDAR data from Fugro, used for visualization

Table 5.1: Overview of possible research datasets

Name	Dimension	Number of points	Format
AHN2	X, Y, Z	6.4×10^{12}	LAS
AHN3	X, Y, Z, intensity, return number, GPS time, classification	6.4×10^{12}	LAS
OSM GPS track	X, Y, Z, t	2.8×10^{9} (until 2013)	GPX
AIS	X, Y, Z, t	2.9×10^{9}	Text
S3DIS	X, Y, Z, R, G, B	2.7×10^{8}	Text

They are chosen because of open access, large data size, diverse attributes and different application purposes.

Software concerned in the research are:

- Oracle, PostgreSQL, PDAL, Lastools, HDF, Potree, Entwine and Greyhound, used for benchmarking
- Python, gcc, Microsoft Visual Studio, Inkscape, LaTeX and Microsoft Office, used for code development and writing
- Unreal, used for VR applications

Hardware:

- The VR equipments include a HTC VIVE headset with resolution 1080 x 1200 pixels per eye, refresh rate 90 Hz and memory size utmost 4 GB, and a desktop with 8-core AMD Ryzen 7 2700X processor at 3.7GHz, a GPU of MSI GeForce GTX 1080 TI - Gaming X - 8GB GDDR5, 16GB of main memory, a 512MB SSD using interface of SATA 6Gb/s.
- A ThinkPad T550 laptop with Dual-Core Intel i5-5200U at 2.2 GHz, 12GB of main memory, a NVIDIA GeForce 940M GPU and a WIN10 operating system. The disk storage is a 1TB SATA 7200 rpm in RAID5 configuration.

- 'Pakhuis': A HP DL380p Gen8 server with 2 × 8-core Intel Xeon processors, E5-2690 at 2.9 GHz, 128GB of main memory, and a RHEL6 operating system. The disk storage which is directly attached consists of a 400GB SSD, 5TB SAS 15 K rpm in RAID5 configuration (internal), and 2 × 41TB SATA 7200 rpm in RAID5 configuration.

5.3 EDUCATION

For a PhD certificate, 3 types of skills are required from TU Delft and they are research skills, discipline related skills and transferable skills. A minimum of 15 Graduate School credits (GSc) should be obtained per category. Considering relativity to the PhD research topic, i.e. management of massive point cloud data, I made the following plan:

Category	GSc
Research skills	21.5
Discipline related skills	18
Transferable skills	16.5
Total GSc	56

5.3.1 Research skills

Learning on-the-job activity

1. Scientific presenting & interacting

Type	GSc indicator	Times planned	Times finished
Addressing a small audience (examples: speaker at national (or minor international) / conference / workshop)	0.5	2	1
Addressing a large audience (examples: speaker at major international conference / workshop incl. conf. paper)	1	2	2
Total GSc	3		

2. Writing & Publishing

Type	GSc indicator	Times planned	Times finished
Paper review	1	1	0
Writing the first conference paper	1	1	1
Writing an international, peer-reviewed journal article	4	1	0
Total GSc	6		

3. Teaching & Supervison

Type	GSc indicator	Times planed	Times finished
Supervising a MSc student (incl. correcting master thesis)	4	1	0
Teaching assistance: assisting in laboratory course/tutorial	3	1	1
Peer-review & peer-learning meetings ($6 \times 0,5$ day), for example as follow up of the PhD Start Up	3	1	1
Total GSc	10		

In total, from learning on the job activity, I plan to acquire 19 GSc.

Graduate school courses

R1.A1 Research Design (2.5 GSc), I have already finished the course.

In total, 21.5 GSc are planned. Until the Go-no-Go meeting, I have acquired 12 GSc.

5.3.2 Discipline related skills

Workshops/seminars

Summer school in Wuhan University from 2nd July 2018 to 6th July 2018 (4 GSc)

The topic is Advanced Point Clouds Processing covering collection, registration, plane fitting, segmentation, classification and object identification. A synthesis project has to be done by a group of 5 people and in the end, it would be presented. I have finished this summer school.

Master courses

Code	Name	GSc	Faculty	Period	Time needed (Hour)
AP3421D	Fundamentals of quantum information	4	TNW	2017/18 Q1	112
IN4392	Cloud computing	5	EWI	2018/19 Q1	140
Total GSc		9			

I have passed the course Fundamentals of quantum information.

Graduate school courses

ABE018 — GIS applied in various domains (5 GSc)

Geographic information system (GIS) applications, with specialized topics related to current societal and environmental issues. I selected 4 modules and they are Sustainable Water Environments, Sustainable Agriculture & Farming and Forest, Smart Sustainable Cities and Climate Change and Renewable Energy Sources. I have accomplished the course.

In total, 18 GSc are planned for discipline related skills. Until Go-no-Go meeting, I have got 13 GSc.

5.3.3 Transferable skills

Courses from the graduate school of TU Delft

Code	Name	GSc	Period	Time needed (Hour)	Whether finished
T1.B2	Writing a Scientific Article in English (PROM-4)	3	Autumn, 2018	24	N
T1.D2	English for Academic Purposes (EAP 3)	3	second semester, 2018	110	Y
T2.A1	How to become effective in a network conversation	1	6/4/2018	8	N
T2.A2	The PhD Network Hub	1	22/3/2018	8	Y
T2.C1	Effective Negotiation: win-win communication	2.5	Summer, 2018	20	N
T2.D1	Leadership, teamwork and group dynamics	1.5	January, 2018	12	Y
T3.B1	Coaching Individual Students and Project Groups	2	May, 2019	16	N
T4.B2	Time Management - first things first	1.5	Spring, 2018	12	Y
T4.B3	Self-management II - Self-management to the max	1	March, 2018	8	Y
Total GSc		16.5			

In total, 16.5 GSc are planned for transferable skills. I have obtained 8 GSc in this category.

On the whole, I have got 33 GSc and there are still 23 GSc to be acquired according to my plan.

5.4 RESEARCH VISITS

It is expected to have at least one longer visit to a university or company. Possible places can be Institut Géographique National (IGN), France or the Universite Paris-Est where Rémi Cura and his team developed an advanced point cloud server, which is closely related to this PhD research. Computer Graphics group of TU Wien is also an ideal choice to learn more about the visualization of massive points. They have conducted intensive research

on rendering algorithms. Another choice can be the Joint Research Centre (JRC) of Wuhan University – Delft University of Technology on Spatial Information where the two parties have approved several cooperative research topics including point clouds. As the research will elaborate Oracle DBMS and Azure, research visits can also be arranged at Oracle or Microsoft. Final decisions will be defined later depending on the progress of the research.

5.5 ORGANIZATION

The research is never an individual work, several groups should be established to assure a success.

5.5.1 Supervision

The supervision team consists of a promotor, Prof.dr. Peter van Oosterom, a co-promotor, Dr. Martijn Meijers and an advisor Ir. Edward Verbree. A progress meeting will be held every two weeks to discuss main results from last two weeks, problems encountered, the plan for the next two weeks and the initial plan to overcome the potential hurdles. Before every meeting, I will write and distribute a progress monitor composed of these four parts, together with related files. And after each meeting, I will summarize key points from the meeting for a consensus. Besides, additional informal meetings may be held when necessary. Supervisors are expected to spend 8 hours per month on the guidance.

5.5.2 Cooperations

Fugro

The research is partially supported by Fugro. And there is an agreement to work at Fugro (Leidschendam) 40 hours per year, mainly for presentations and discussions. It is planned in the beginning of the research, meetings with professionals from Fugro should be more frequently arranged to collect problems and set up the schedule.

VR team, Architecture Faculty, TU Delft

The VR team can provide prevalent VR equipment including both hardware and software (Section 5.2). Besides, Mr. Arno Freeke, the coordinator has plenty experience and could offer technical support during the implementation.

TU Wien

Mr. Markus Schütz from TU Wien is the founder of the WebGL based point cloud renderer Potree, which is the core of the AHN2-viewer discussed in the research. Octree data structure of Potree is also the source for the "block" effect. To tackle the issue, i.e. embedding the nD PointCloud structure into Potree, close cooperation with Mr. Schütz is planed.

Wuhan University

Dr. Xuefeng Guan comes from Wuhan University and he is the developer of the SFClib. Together with Dr. Guan, we plan to enhance the library to make it more flexible and robust.

5.5.3 MSc students

I will acquire information and knowledge from MSc synthesis projects and thesis conducted in parallel, e.g. massive point cloud management in a distributed environment using the Hadoop framework.

Guidance of MSc thesis research related to this topic is also considered during the PhD period. Possible directions can be developing one or two functionalities based on the nD PointCloud structure, e.g. change detection and visibility detection. Specific topics will be defined after certain amount of exploration. Then I will post the information on the website of MSc Geomatics in TU Delft.

5.6 CONTRIBUTIONS AND BENEFITS

5.6.1 Adding knowledge

The original goal of a PhD research is to discover new things, so does this research. The main development will include:

- cLoD concept in a high dimensional space and the method to compute it
- Nature and management of dimensions that can be utilized for structuring data

5.6.2 Contribution to industry

Although only 3 use cases will be implemented in the research, the nD PointCloud structure created tends to benefit more point cloud applications. As a matter of fact, the research will at least provide a solution of rendering massive point clouds in a VR environment.

5.6.3 Societal benefits

Contribution to a new OGC standard "Web Point Cloud Service" is expected in the end. The research is also aimed to facilitate the construction of large data centers of point clouds in Europe with improved viewing or download services for the public.

5.7 PUBLICATION GOALS

5.7.1 Journal paper

1. ND PointCloud structure for indoor point data where classification is involved. Benchmark will be executed to evaluate the performance.

Use cases can be indoor navigation, visualization and visibility detection. Possible solutions include the IOT approach and the nested table approach.

2. IOT approach with in-database encoding and decoding functions for managing AHN2. Use case elaborates visualization and change detection. The "block" effect should be eliminated. Benchmark test will be performed.

3. Adapted nD PointCloud structure to handle GPS data featuring in the temporal dimension. Application could be feature extraction or mining.

4. The adoption of nD PointCloud on a cloud computing platform. Protocols will be explored together with an implementation of VR applications.

5. Comprehensive point cloud benchmark: simulating users' query habit together with a cost model. The benchmark should also be applicable in a cloud environment. The benchmark will be defended by performing less tests.

6. A generic conceptual model covering nD point cloud and corresponding use cases. Connect the model to different use cases could be a demonstration.

5.7.2 Conference paper/abstract

1. Problems derived from literature and experiments on management of massive points. Initial results will be presented.

2. Initial benchmark tests based on possible point cloud management solutions inside Oracle.

3. Test of specific database processes using the nD PointCloud structure, e.g. data loading with cLoD utilized based on a simple SFC storage.

4. DBMS to solve discrete visualization of point clouds/VR with new data structure.

5. Server-protocol-clients architecture for point cloud management and computation. Initial results will be discussed.

6. Conceptual model + nD PointCloud structure + implementation in a database + demonstration of certain spatial operators, e.g.spatial join, nearest neighbor search.

7. Design of a comprehensive benchmark for point cloud management.

8. Improvement of use cases with new platforms, e.g. parallelization/-cloud computation + optimization + benchmark.

9. Point cloud management strategy on the quantum platform and possible simulations.

Topics listed above are possible directions. In fact, due to time limitation, some of them may not be realized. However, the graduation requires at least 3 journal papers and 6 conference papers/abstracts.

5.8 JOURNALS AND CONFERENCES

5.8.1 Journals

Journals that can be considered to publish the results include:

1. ISPRS Journal of Photogrammetry and Remote Sensing (ISPRS), IF=6.387
2. International Journal of Geographical Information Science (IJGIS), IF=2.502
3. ISPRS International Journal of Geo-Information (IJGI), IF=1.502
4. Computers, Environment and Urban Systems, IF=3.724
5. Transactions in GIS (TGIS), IF=2.252
6. Computers & Geosciences, IF=2.533
7. Computers & Graphics, IF=1.176
8. Journal of Digital Earth (TJDE), IF=2.292

5.8.2 Conferences

Recent conferences will include, for example:

1. AGILE 2018, 12-15 June, Lund, Sweden
2. ISPRS Technical Commission IV Symposium 2018, 1-5 October, Delft, the Netherlands
3. AGILE 2019, June, Cyprus
4. ISPRS Geospatial Week 2019, 10-14 June, Enschede, the Netherlands
5. COSIT 2019, 9-13 September, Regensburg, Germany

BIBLIOGRAPHY

Agarwal, D. and Prasad, S. K. (2012). Azurebench: Benchmarking the storage services of the azure cloud platform. In *2012 IEEE 26th International Conference on Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW)*, pages 1048–1057. IEEE. [Cited on Page 21]

Akioka, S. and Muraoka, Y. (2010). HPC benchmarks on Amazon EC2. In *2010 IEEE 24th International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, pages 1029–1034. IEEE. [Cited on Page 21]

Almoradie, A., Jonoski, A., Popescu, I., and Solomatine, D. (2013). Web based access to water related data using OGC WaterML 2.0. *International Journal of Advanced Computer Science and Applications, EnviroGRIDS Special Issue on "Building a Regional Observation System in the Black Sea Catchment*, pages 83–89. [Cited on Page 28]

Alsina, D. and Latorre, J. I. (2016). Experimental test of Mermin inequalities on a five-qubit quantum computer. *Physical Review A*, 94(1):012314. [Cited on Page 22]

Armeni, I., Sener, O., Zamir, A. R., Jiang, H., Brilakis, I., Fischer, M., and Savarese, S. (2016). 3D semantic parsing of large-scale indoor spaces. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*. [Cited on Page 44]

Bailey, D. H., Barszcz, E., Barton, J. T., Browning, D. S., Carter, R. L., Dagum, L., Fatoohi, R. A., Frederickson, P. O., Lasinski, T. A., Schreiber, R. S., et al. (1991). The NAS parallel benchmarks. *The International Journal of Supercomputing Applications*, 5(3):63–73. [Cited on Page 21]

Baumann, P. (2010). OGC WCS 2.0 interface standard—core. *Open Geospatial Consortium: Wayland, MA, USA*. [Cited on Page 24]

Biljecki, F., Ledoux, H., and Van Oosterom, P. (2013). Transportation mode-based segmentation and classification of movement trajectories. *International Journal of Geographical Information Science*, 27(2):385–407. [Cited on Page 36]

Box, D., Ehnebuske, D., Kakivaya, G., Layman, A., Mendelsohn, N., Nielsen, H. F., Thatte, S., and Winer, D. (2000). Simple object access protocol (SOAP) 1.1. Available at https://www.w3.org/TR/2000/NOTE-SOAP-20000508/, accessed 28-February-2018. [Cited on Page 23]

Cura, R. (2016). *Inverse procedural Street Modelling: from interactive to automatic reconstruction*. PhD thesis, Université Paris-Est Marne-la-Vallée, France. [Cited on Pages 10 and 32]

Cura, R., Perret, J., and Paparoditis, N. (2017). A scalable and multi-purpose point cloud server (PCS) for easier and faster point cloud data management and processing. *ISPRS Journal of Photogrammetry and Remote Sensing*, 127:39–56. [Cited on Pages 17 and 38]

Dobos, L., Csabai, I., Szalai-Gindl, J. M., Budavári, T., and Szalay, A. S. (2014). Point cloud databases. In *Proceedings of the 26th International Conference on Scientific and Statistical Database Management*, page 33. ACM. [Cited on Page 17]

Elseberg, J., Borrmann, D., and Nüchter, A. (2013). One billion points in the cloud–an octree for efficient processing of 3D laser scans. *ISPRS Journal of Photogrammetry and Remote Sensing*, 76:76–88. [Cited on Page 9]

Guan, X., van Oosterom, P., and Cheng, B. (2018). A parallel n-dimensional space-filling curve library and its application in massive point cloud management. *ISPRS International Journal of Geo-Information*, 7(8):19. [Cited on Pages 7, 10, 14, and 32]

Isenburg, M. (2013). LASzip: lossless compression of LiDAR data. *Photogrammetric Engineering and Remote Sensing*, 79(2):209–217. [Cited on Pages 15 and 16]

Islamaj Dogan, R., Murray, G. C., Névéol, A., and Lu, Z. (2009). Understanding PubMed® user search behavior through log analysis. *Database*, 2009. [Cited on Page 41]

ISPRS (2011). LAS 1.4 format specification. Technical report, The American Society for Photogrammetry And Remote Sensing. Available at http://www.asprs.org/a/society/committees/standards/LAS_1_4_r13.pdf, accessed 22-February-2018. [Cited on Page 15]

Jackson, K. R., Ramakrishnan, L., Muriki, K., Canon, S., Cholia, S., Shalf, J., Wasserman, H. J., and Wright, N. J. (2010). Performance analysis of high performance computing applications on the amazon web services cloud. In *2010 IEEE Second International Conference onCloud Computing Technology and Science (CloudCom)*, pages 159–168. IEEE. [Cited on Page 22]

Jiang, B. (2013). Head/tail breaks: A new classification scheme for data with a heavy-tailed distribution. *The Professional Geographer*, 65(3):482–494. [Cited on Pages 7 and 31]

Kodde, M. (2010). The art of collecting and disseminating point clouds. *Management of Massive Point Cloud Data: Wet and Dry*, pages 9–15. [Cited on Page 40]

Lee, J., Li, K.-J., Zlatanova, S., Kolbe, T., Nagel, C., and Becker, T. (2014). OGC IndoorGML. *Open Geospatial Consortium standard*. [Cited on Page 28]

Liu, H., van Oosterom, P., Hu, C., and Wang, W. (2016). Managing large multidimensional array hydrologic datasets: a case study comparing NetCDF and SciDB. *Procedia Engineering*, 154:207–214. [Cited on Pages 5 and 42]

Ma, Y., Wu, H., Wang, L., Huang, B., Ranjan, R., Zomaya, A., and Jie, W. (2015). Remote sensing big data computing: Challenges and opportunities. *Future Generation Computer Systems*, 51:47–60. [Cited on Page 5]

Manning, C. (2017). Trillions of points - massive point clouds as infrastructure. Available at https://s3.amazonaws.com/entwine.io/slides/foss4g2017/index.html, accessed 28-February-2018. [Cited on Page 11]

Martinez-Rubi, O. (2015). Massive point clouds for eSciences. Presentation. [Cited on Pages 6 and 42]

Martinez-Rubi, O., Verhoeven, S., Van Meersbergen, M., Schûtz, M., Van Oosterom, P., Gonçalves, R., and Tijssen, T. (2015). Taming the beast: Free and open-source massive point cloud web visualization. In *Capturing Reality Forum 2015, 23-25 November 2015, Salzburg, Austria*. The Servey Association. [Cited on Page 11]

Meijers, M., Quak, W., and van Oosterom, P. (2017). Archiving AIS messages in a Geo-DBMS. In Arnold Bregt, Tapani Sarjakoski, R. v. L. and Rip, F., editors, *Societal Geo-Innovation : short papers, posters and poster abstracts of the 20th AGILE Conference on Geographic Information Science*, Wageningen, the Netherlands. Wageningen University and Research. Available at https://agile-online.org/index.php/conference/proceedings/proceedings-2017, accessed 24-June-2018. [Cited on Pages 36, 37, and 44]

Müller, M. and Pross, B. (2015). OGC WPS 2.0 interface standard. *Open Geospatial Consortium Inc.: Wayland, MA, USA*. [Cited on Page 23]

NGA (2015). Sensor independent point cloud (SIPC) volume 1 design and implementation description document (DIDD) dated 2015-08-31 (LIWG ver 1.02). Technical report, National Center for Geospatial Intelligence Standards (NCGIS), NGA. Available at https://nsgreg.nga.mil/doc/view?i=4206&month=2&day=5&year=2016, accessed 22-February-2018. [Cited on Page 16]

NIST (2018). Notice of funding opportunity (nofo) NIST public safety innovation accelerator program (PSIAP)– point cloud city. Technical report, National Institute of Standards and Technology. Available at https://www.nist.gov/sites/default/files/documents/2018/01/05/2018-nist-psiap-pc2_nofo.pdf, accessed 22-February-2018. [Cited on Page 1]

Ohori, K. A., Ledoux, H., Biljecki, F., and Stoter, J. (2015). Modeling a 3D city model and its levels of detail as a true 4D model. *ISPRS International Journal of Geo-Information*, 4(3):1055–1075. [Cited on Page 2]

Otepka, J., Ghuffar, S., Waldhauser, C., Hochreiter, R., and Pfeifer, N. (2013). Georeferenced point clouds: A survey of features and point cloud management. *ISPRS International Journal of Geo-Information*, 2(4):1038–1065. [Cited on Page 1]

Otepka, J., Mandlburger, G., and Karel, W. (2012). The OPALS data manager — efficient data management for processing large airborne laser scanning projects. *Proceedings of the ISPRS Annals of the Photogrammetry, Melbourne, Australia*, 25:153–159. [Cited on Page 15]

Petitet, A., Whaley, R., Dongarra, J., and Cleary, A. (2004). HPL-a portable implementation of the high-performance linpack benchmark for distributed-memory computers. Technical report, University of Tennessee. Available at http://www.netlib.org/benchmark/hpl/. [Cited on Page 21]

Poux, F., Neuville, R., Hallot, P., and Billen, R. (2017). Model for reasoning from semantically rich point cloud data. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 4:107–115. [Cited on Pages 28, 29, and 30]

Psomadaki, S. (2016). Using a space filling curve for the management of dynamic point cloud data in a relational DBMS. Master's thesis, Delft University of Technology, the Netherlands. [Cited on Pages 6, 14, and 31]

Qi, C. R., Su, H., Kaichun, M., and Guibas, L. J. (2017). Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 77–85. IEEE. [Cited on Page 3]

Qin, R., Tian, J., and Reinartz, P. (2016). 3d change detection–approaches and applications. *ISPRS Journal of Photogrammetry and Remote Sensing*, 122:41–56. [Cited on Page 35]

Ramsey, P. (2014). A PostgreSQL extension for storing point cloud (LIDAR) data. Available at `https://github.com/pgpointcloud/pointcloud`, accessed 22-February-2018. [Cited on Page 20]

Ravada, S., Horhammer, M., and Kazar, B. M. (2010). Point cloud: Storage, loading, and visualization. In *Proceedings of National Science Foundation Tera Grid Workshop on Cyber-GIS, Washington DC, USA*. Available at `http://www.cigi.illinois.edu/cybergis/docs/Kazar_Position_Paper.pdf`, accessed 28-February-2018. [Cited on Page 20]

Richter, R. and Döllner, J. (2014). Concepts and techniques for integration, analysis and visualization of massive 3D point clouds. *Computers, Environment and Urban Systems*, 45:114–124. [Cited on Pages 17, 38, and 39]

Rieg, L., Wichmann, V., Rutzinger, M., Sailer, R., Geist, T., and Stötter, J. (2014). Data infrastructure for multitemporal airborne LiDAR point cloud analysis–examples from physical geography in high mountain environments. *Computers, Environment and Urban Systems*, 45:137–146. [Cited on Page 17]

Roy, S., Kot, L., and Koch, C. (2013). Quantum databases. In *Proceedings of CIDR 2013: The 6th Biennial Conference on Innovative Data Systems Research*. [Cited on Page 22]

Samet, H. (1984). The Quadtree and related hierarchical data structures. *ACM Computing Surveys (CSUR)*, 16(2):187–260. [Cited on Page 9]

Schilling, A. and Kolbe, T. H. (2010). Draft for candidate OpenGIS® web 3D service interface standard. *OpenGeospatial Consortium*. [Cited on Page 24]

Schütz, M. and Wimmer, M. (2015). Rendering large point clouds in web browsers. In *Proceedings of CESCG 2015: The 19th Central European Seminar on Computer Graphics*, pages 83–90. Vienna University of Technology. [Cited on Page 11]

Van Oosterom, P. (2016). nD-PointClouds: a model for deeply integrating space, time and scale. Presentation. Available at `http://www.gdmc.nl/3DCadastres/workshop2016/programme/Pres2016_31.pdf`, accessed 15-August-2018. [Cited on Page 2]

Van Oosterom, P., Martinez-Rubi, O., Ivanova, M., Horhammer, M., Geringer, D., Ravada, S., Tijssen, T., Kodde, M., and Gonçalves, R. (2015). Massive point cloud data management: Design, implementation and execution of a point cloud benchmark. *Computers & Graphics*, 49:92–125. [Cited on Pages 1, 5, 6, 14, 15, 25, 34, and 41]

Van Oosterom, P., Martinez-Rubi, O., Tijssen, T., and Gonçalves, R. (2016). Realistic benchmarks for point cloud data management systems. In *Advances in 3D Geoinformation*, pages 1–30. Springer International Publishing. [Cited on Pages 2, 4, 16, 33, and 41]

Van Oosterom, P. and Meijers, M. (2014). Vario-scale data structures supporting smooth zoom and progressive transfer of 2D and 3D data. *International Journal of Geographical Information Science*, 28(3):455–478. [Cited on Page 2]

Van Winden, K., Biljecki, F., and Van der Spek, S. (2016). Automatic update of road attributes by mining GPS tracks. *Transactions in GIS*, 20(5):664–683. [Cited on Page 36]

Verma, U. and Butler, H. (2014). Plasio. Available at `https://github.com/verma/plasio`, accessed 28-February-2018. [Cited on Page 11]

Vo, A.-V. (2017). *Spatial data storage and processing strategies for urban laser scanning*. PhD thesis, University College Dublin, Ireland. [Cited on Pages 18 and 32]

Vretanos, P. A. (2010). OpenGIS web feature service 2.0 interface standard. *OpenGIS Project Document: OGC*. [Cited on Page 24]

Wang, L., Ma, Y., Yan, J., Chang, V., and Zomaya, A. Y. (2018a). PipsCloud: High performance cloud computing for remote sensing big data management and processing. *Future Generation Computer Systems*, 78:353–368. [Cited on Pages 14, 21, 37, and 38]

Wang, L., Ma, Y., Zomaya, A. Y., Ranjan, R., and Chen, D. (2015). A parallel file system with application-aware data layout policies for massive remote sensing image processing in digital earth. *IEEE Transactions on Parallel and Distributed Systems*, 26(6):1497–1508. [Cited on Page 14]

Wang, W., Lv, Z., Li, X., Xu, W., Zhang, B., Zhu, Y., and Yan, Y. (2018b). Spatial query based virtual reality GIS analysis platform. *Neurocomputing*, 274:88–98. [Cited on Page 35]

Wang, Y. and Guo, M. (2012). A combined 2D and 3D spatial indexing of very large point-cloud datasets. *Acta Geodaetica et Cartographica Sinica (in Chinese)*, 41(4):605–612. [Cited on Page 13]

Wiebe, N., Kapoor, A., and Svore, K. M. (2015). Quantum algorithms for nearest-neighbor methods for supervised and unsupervised learning. *Quantum Information and Computation*, 15(3):318–358. [Cited on Page 40]

Wijga-Hoefsloot, M. (2012). Point clouds in a database: Data management within an engineering company. Master's thesis, Delft University of Technology, the Netherlands. [Cited on Page 1]

Xie, H., Wu, B., and Zhao, Z. (2013). A novel organization method of massive point cloud. *Remote Sensing Information (in Chinese)*, 28(6):26–32. [Cited on Page 9]

Yang, J. and Huang, X. (2014). A hybrid spatial index for massive point cloud data management and visualization. *Transactions in GIS*, 18(S1):97–108. [Cited on Page 11]

Younes, A. (2007). Database manipulation on quantum computers. *arXiv preprint arXiv:0705.4303*. [Cited on Page 23]

Zhang, Y. (2016). The D-FCM partitioned D-BSP tree for massive point cloud data access and rendering. *ISPRS Journal of Photogrammetry and Remote Sensing*, 120:25–36. [Cited on Page 13]

Zhao, H., Wang, J., Wang, Y., and Guo, M. (2015). A new multi-level spatial index of scattered point cloud data. *Journal of Geo-information Science (in Chinese)*, 17(12):1450–1455. [Cited on Page 13]

Zheng, Y., Ou, Y., Lex, A., and Phillips, J. M. (2017). Visualization of big spatial data using coresets for kernel density estimates. *arXiv preprint arXiv:1709.04453*. [Cited on Page 10]

COLOPHON

This document was typeset using LaTeX. The document layout was generated using the `arsclassica` package by Lorenzo Pantieri, which is an adaption of the original `classicthesis` package from André Miede.

Reports published before in this series

1. GISt Report No. 1, Oosterom, P.J. van, Research issues in integrated querying of geometric and thematic cadastral information (1), Delft University of Technology, Rapport aan Concernstaf Kadaster, Delft 2000, 29 p.p.
2. GISt Report No. 2, Stoter, J.E., Considerations for a 3D Cadastre, Delft University of Technology, Rapport aan Concernstaf Kadaster, Delft 2000, 30.p.
3. GISt Report No. 3, Fendel, E.M. en A.B. Smits (eds.), Java GIS Seminar, Opening GDMC, Delft 15 November 2000, Delft University of Technology, GISt. No. 3, 25 p.p.
4. GISt Report No. 4, Oosterom, P.J.M. van, Research issues in integrated querying of geometric and thematic cadastral information (2), Delft University of Technology, Rapport aan Concernstaf Kadaster, Delft 2000, 29 p.p.
5. GISt Report No. 5, Oosterom, P.J.M. van, C.W. Quak, J.E. Stoter, T.P.M. Tijssen en M.E. de Vries, Objectgerichtheid TOP10vector: Achtergrond en commentaar op de gebruikersspecificaties en het conceptuele gegevensmodel, Rapport aan Topografische Dienst Nederland, E.M. Fendel (eds.), Delft University of Technology, Delft 2000, 18 p.p.
6. GISt Report No. 6, Quak, C.W., An implementation of a classification algorithm for houses, Rapport aan Concernstaf Kadaster, Delft 2001, 13.p.
7. GISt Report No. 7, Tijssen, T.P.M., C.W. Quak and P.J.M. van Oosterom, Spatial DBMS testing with data from the Cadastre and TNO NITG, Delft 2001, 119 p.
8. GISt Report No. 8, Vries, M.E. de en E. Verbree, Internet GIS met ArcIMS, Delft 2001, 38 p.
9. GISt Report No. 9, Vries, M.E. de, T.P.M. Tijssen, J.E. Stoter, C.W. Quak and P.J.M. van Oosterom, The GML prototype of the new TOP10vector object model, Report for the Topographic Service, Delft 2001, 132 p.
10. GISt Report No. 10, Stoter, J.E., Nauwkeurig bepalen van grondverzet op basis van CAD ontgravingsprofielen en GIS, een haalbaarheidsstudie, Rapport aan de Bouwdienst van Rijkswaterstaat, Delft 2001, 23 p.
11. GISt Report No. 11, Geo DBMS, De basis van GIS-toepassingen, KvAG/AGGN Themamiddag, 14 november 2001, J. Flim (eds.), Delft 2001, 37 p.
12. GISt Report No. 12, Vries, M.E. de, T.P.M. Tijssen, J.E. Stoter, C.W. Quak and P.J.M. van Oosterom, The second GML prototype of the new TOP10vector object model, Report for the Topographic Service, Delft 2002, Part 1, Main text, 63 p. and Part 2, Appendices B and C, 85 p.
13. GISt Report No. 13, Vries, M.E. de, T.P.M. Tijssen en P.J.M. van Oosterom, Comparing the storage of Shell data in Oracle spatial and in Oracle/ArcSDE compressed binary format, Delft 2002, .72 p. (Confidential)
14. GISt Report No. 14, Stoter, J.E., 3D Cadastre, Progress Report, Report to Concernstaf Kadaster, Delft 2002, 16 p.
15. GISt Report No. 15, Zlatanova, S., Research Project on the Usability of Oracle Spatial within the RWS Organisation, Detailed Project Plan (MD-NR. 3215), Report to Meetkundige Dienst – Rijkswaterstaat, Delft 2002, 13 p.
16. GISt Report No. 16, Verbree, E., Driedimensionale Topografische Terreinmodellering op basis van Tetraëder Netwerken: Top10-3D, Report aan Topografische Dienst Nederland, Delft 2002, 15 p.
17. GISt Report No. 17, Zlatanova, S. Augmented Reality Technology, Report to SURFnet bv, Delft 2002, 72 p.
18. GISt Report No. 18, Vries, M.E. de, Ontsluiting van Geo-informatie via netwerken, Plan van aanpak, Delft 2002, 17p.
19. GISt Report No. 19, Tijssen, T.P.M., Testing Informix DBMS with spatial data from the cadastre, Delft 2002, 62 p.

20. GISt Report No. 20, Oosterom, P.J.M. van, Vision for the next decade of GIS technology, A research agenda for the TU Delft the Netherlands, Delft 2003, 55 p.
21. GISt Report No. 21, Zlatanova, S., T.P.M. Tijssen, P.J.M. van Oosterom and C.W. Quak, Research on usability of Oracle Spatial within the RWS organisation, (AGI-GAG-2003-21), Report to Meetkundige Dienst – Rijkswaterstaat, Delft 2003, 74 p.
22. GISt Report No. 22, Verbree, E., Kartografische hoogtevoorstelling TOP10vector, Report aan Topografische Dienst Nederland, Delft 2003, 28 p.
23. GISt Report No. 23, Tijssen, T.P.M., M.E. de Vries and P.J.M. van Oosterom, Comparing the storage of Shell data in Oracle SDO_Geometry version 9i and version 10g Beta 2 (in the context of ArcGIS 8.3), Delft 2003, 20 p. (Confidential)
24. GISt Report No. 24, Stoter, J.E., 3D aspects of property transactions: Comparison of registration of 3D properties in the Netherlands and Denmark, Report on the short-term scientific mission in the CIST – G9 framework at the Department of Development and Planning, Center of 3D geo-information, Aalborg, Denmark, Delft 2003, 22 p.
25. GISt Report No. 25, Verbree, E., Comparison Gridding with ArcGIS 8.2 versus CPS/3, Report to Shell International Exploration and Production B.V., Delft 2004, 14 p. (confidential).
26. GISt Report No. 26, Penninga, F., Oracle 10g Topology, Testing Oracle 10g Topology with cadastral data, Delft 2004, 48 p.
27. GISt Report No. 27, Penninga, F., 3D Topography, Realization of a three dimensional topographic terrain representation in a feature-based integrated TIN/TEN model, Delft 2004, 27 p.
28. GISt Report No. 28, Penninga, F., Kartografische hoogtevoorstelling binnen TOP10NL, Inventarisatie mogelijkheden op basis van TOP10NL uitgebreid met een Digitaal Hoogtemodel, Delft 2004, 29 p.
29. GISt Report No. 29, Verbree, E. en S.Zlatanova, 3D-Modeling with respect to boundary representations within geo-DBMS, Delft 2004, 30 p.
30. GISt Report No. 30, Penninga, F., Introductie van de 3e dimensie in de TOP10NL; Voorstel voor een onderzoekstraject naar het stapsgewijs introduceren van 3D data in de TOP10NL, Delft 2005, 25 p.
31. GISt Report No. 31, P. van Asperen, M. Grothe, S. Zlatanova, M. de Vries, T. Tijssen, P. van Oosterom and A. Kabamba, Specificatie datamodel Beheerkaart Nat, RWS-AGI report/GIST Report, Delft, 2005, 130 p.
32. GISt Report No. 32, E.M. Fendel, Looking back at Gi4DM, Delft 2005, 22 p.
33. GISt Report No. 33, P. van Oosterom, T. Tijssen and F. Penninga, Topology Storage and the Use in the context of consistent data management, Delft 2005, 35 p.
34. GISt Report No. 34, E. Verbree en F. Penninga, RGI 3D Topo - DP 1-1, Inventarisatie huidige toegankelijkheid, gebruik en mogelijke toepassingen 3D topografische informatie en systemen, 3D Topo Report No. RGI-011-01/GISt Report No. 34, Delft 2005, 29 p.
35. GISt Report No. 35, E. Verbree, F. Penninga en S. Zlatanova, Datamodellering en datastructurering voor 3D topografie, 3D Topo Report No. RGI-011-02/GISt Report No. 35, Delft 2005, 44 p.
36. GISt Report No. 36, W. Looijen, M. Uitentuis en P. Bange, RGI-026: LBS-24-7, Tussenrapportage DP-1: Gebruikerswensen LBS onder redactie van E. Verbree en E. Fendel, RGI LBS-026-01/GISt Rapport No. 36, Delft 2005, 21 p.
37. GISt Report No. 37, C. van Strien, W. Looijen, P. Bange, A. Wilcsinszky, J. Steenbruggen en E. Verbree, RGI-026: LBS-24-7, Tussenrapportage DP-2: Inventarisatie geo-informatie en -services onder redactie van E. Verbree en E. Fendel, RGI LBS-026-02/GISt Rapport No. 37, Delft 2005, 21 p.
38. GISt Report No. 38, E. Verbree, S. Zlatanova en E. Wisse, RGI-026: LBS-24-7, Tussenrapportage DP-3: Specifieke wensen en eisen op het gebied van plaatsbepaling, privacy en beeldvorming, onder redactie van E. Verbree en E. Fendel, RGI LBS-026-03/GISt Rapport No. 38, Delft 2005, 15 p.

39. GISt Report No. 39, E. Verbree, E. Fendel, M. Uitentuis, P. Bange, W. Looijen, C. van Strien, E. Wisse en A. Wilcsinszky en E. Verbree, RGI-026: LBS-24-7, Eindrapportage DP-4: Workshop 28-07-2005 Geo-informatie voor politie, brandweer en hulpverlening ter plaatse, RGI LBS-026- 04/GISt Rapport No. 39, Delft 2005, 18 p.
40. GISt Report No. 40, P.J.M. van Oosterom, F. Penninga and M.E. de Vries, Trendrapport GIS, GISt Report No. 40 / RWS Report AGI-2005-GAB-01, Delft, 2005, 48 p.
41. GISt Report No. 41, R. Thompson, Proof of Assertions in the Investigation of the Regular Polytope, GISt Report No. 41 / NRM-ISS090, Delft, 2005, 44 p.
42. GISt Report No. 42, F. Penninga and P. van Oosterom, Kabel- en leidingnetwerken in de kadastrale registratie (in Dutch) GISt Report No. 42, Delft, 2006, 38 p.
43. GISt Report No. 43, F. Penninga and P.J.M. van Oosterom, Editing Features in a TEN-based DBMS approach for 3D Topographic Data Modelling, Technical Report, Delft, 2006, 21 p.
44. GISt Report No. 44, M.E. de Vries, Open source clients voor UMN MapServer: PHP/Mapscript, JavaScript, Flash of Google (in Dutch), Delft, 2007, 13 p.
45. GISt Report No. 45, W. Tegtmeier, Harmonization of geo-information related to the lifecycle of civil engineering objects – with focus on uncertainty and quality of surveyed data and derived real world representations, Delft, 2007, 40 p.
46. GISt Report No. 46, W. Xu, Geo-information and formal semantics for disaster management, Delft, 2007, 31 p.
47. GISt Report No. 47, E. Verbree and E.M. Fendel, GIS technology – Trend Report, Delft, 2007, 30 p.
48. GISt Report No. 48, B.M. Meijers, Variable-Scale Geo-Information, Delft, 2008, 30 p.
49. GISt Report No. 48, Maja Bitenc, Kajsa Dahlberg, Fatih Doner, Bas van Goort, Kai Lin,Yi Yin, Xiaoyu Yuan and Sisi Zlatanova, Utilty Registration, Delft, 2008, 35 p.
50. GISt Report No 50, T.P.M. Tijssen en S. Zlatanova, Oracle Spatial 11g en ArcGIS 9.2 voor het beheer van puntenwolken (Confidential), Delft, 2008, 16 p.
51. GISt Report No. 51, S. Zlatanova, Geo-information for Crisis Management, Delft, 2008, 24 p.
52. GISt Report No. 52, P.J.M. van Oosterom, INSPIRE activiteiten in het jaar 2008 (partly in Dutch), Delft, 2009, 142 p.
53. GISt Report No. 53, P.J.M. van Oosterom with input of and feedback by Rod Thompson and Steve Huch (Department of Environment and Resource Management, Queensland Government), Delft, 2010, 60 p.
54. GISt Report No. 54, A. Dilo and S. Zlatanova, Data modeling for emergency response, Delft, 2010, 74 p.
55. GISt Report No. 55, Liu Liu, 3D indoor " door-to-door" navigation approach to support first responders in emergency response – PhD Research Proposal, Delft, 2011, 47 p.
56. GISt Report No. 56, Md. Nazmul Alam, Shadow effect on 3D City Modelling for Photovoltaic Cells – PhD Proposal, Delft, 2011, 39 p.
57. GIST Report No. 57, G.A.K. Arroyo Ohori, Realsing the Foundations of a Higher Dimensional GIS: A Study of Higher Dimensional Data Models, Data Structures and Operations – PhD Research Proposal, Delft, 2011, 68 p.
58. GISt Report No. 58, Zhiyong Wang, Integrating Spatio-Temporal Data into Agent-Based Simulation for Emergency Navigation Support – PhD Research Proposal, Delft, 2012, 49 p.
59. GISt Report No. 59, Theo Tijssen, Wilko Quak and Peter van Oosterom, Geo-DBMS als standard bouwsteen voor Rijkswaterstaat (in Dutch), Delft, 2012, 167 p.
60. GISt Report No. 60, Amin Mobasheri, Designing formal semantics of geo-information for disaster response – PhD Research Proposal, Delft, 2012, 61 p.
61. GISt Report No. 61, Simeon Nedkov, Crowdsourced WebGIS for routing applications in disaster management situations, Delft, 2012, 31 p.

62. GISt Report No. 62, Filip Biljecki, The concept of level of detail in 3D city models – PhD Research Proposal, Delft, 2013, 58 p.
63. GISt Report No. 63, Theo Tijssen & Wilko Quak, GISt activiteiten voor het GeoValley project – Projectnummer: GBP / 21F.005, Delft, 2013, 39 p.
64. GISt Report No. 64, Radan Šuba, Content of Variable-scale Maps – PhD Proposal, Delft, 2013, 36 p.
65. GISt Report No. 65, Ravi Peters, Feature aware Digital Surface Model analysis and generalization based on the 3D Medial Axis Transform, - PhD Research Proposal, Delft 2014, 55 p.
66. GISt Report No. 66, Sisi Zlatanova, Liu Liu, George Sithole, Junqiao Zhao and Filippo Mortani, Space subdivision for indoor applications, Delft, 2014, 38 p.
67. GISt Report No. 67, Sisi Zlatanova, Jakob Beetz, Joris Goos, Albert Mulder, Anne-Jan Boersma, Hans Schevers, Marian de Vries and Tarun Ghawana, Spatial Infrastructure for the Port of Rotterdam, Delft, 2014, 35 p.
68. GISt Report No. 68, Edward Verbree en Peter van Oosterom, Exploratieve Puntenwolken, Delft, 2015, 60 p.
69. GISt Report No. 69, Wilko Quak, Storage of spatial properties of NWB-vaarwegen in Neo4j, Delft, 2016, 13 p.
70. GISt Report No. 70, Edward Verbree and Peter van Oosterom, Standaardisatie van Puntenwolken, Delft, 2016, 66 p.
71. GISt Report No. 71, Martijn Meijers, Peter van Oosterom and Wilko Quak, Management of AIS messages in a Geo-DBMS, Delft, 2018, 33 p.
72. GISt Report No. 72, Martijn Meijers and Peter van Oosterom, Clustering and Indexing historic AIS data with Space Filling Curves, Delft, 2018, 33 p.
73. GISt Report No. 73, Abdullah Allatas, Supporting indoor navigation using access rights to spaces based on an integrated IndoorGML and LADM model, - PhD Research Proposal, Delft, 2019, 54 p.
74. GISt Report No. 74, Mingxue Zheng, Classification of Point Clouds of Urban Scenes exploiting a hierarchy of Point Densities, - PhD Research Proposal, Delft, 2019, 59 p.

OTB – Research for the built environment
Faculty of Architecture and the Built Environment, TU Delft
Julianalaan 134, 2628 BL Delft
Postbus 5043, 2600 GA Delft

Phone: +31 (0)15 278 30 05
E-mail: OTB-bk@tudelft.nl
www.otb.bk.tudelft.nl