OTB – Onderzoek voor
de gebouwde omgeving
Faculteit Bouwkunde

STANDAARDISATIE VAN PUNTENWOLKEN

Edward Verbree & Peter van Oosterom

December, 2016
GISt Rapport No. 70 – Versie 1.1

TU Delft Delft University of Technology

Challenge the future

STANDAARDISATIE VAN PUNTENWOLKEN
Dit project is uitgevoerd in het kader van de RWS-TUD Raamovereenkomst betreffende Samenwerking en Kennisuitwisseling op gebied van Ruimtelijke Informatievoorziening (zaaknummer 31103836)

Auteurs:
Edward Verbree & Peter van Oosterom
Contactpersoon Rijkswaterstaat: Dick ten Napel

GISt Rapport No. 70 – Versie 1.1

December, 2016

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

Inhoud

1 Inleiding

In het kader van de Raamovereenkomst Rijkswaterstaat – TU Delft is eind 2015 het vliegende start project 'Exploratieve Puntenwolken' naar tevredenheid afgerond. Dat project richtte zich op het onderkennen, onderzoeken en het bieden van oplossingsrichtingen voor het direct en exploratief gebruik van puntenwolkdata. De resultaten van dat project zijn beschreven in het GISt Rapport No. 68 "Exploratieve Puntenwolken" d.d. 21 december 2015. Hoofdstuk 3 van dat rapport biedt een aantal aanbevelingen voor mogelijk vervolgonderzoek:

- *Multi Level of Detail (LOD) puntenwolken*
- *Standaardisatie puntenwolken*
- *Directe berekening en analyse puntenwolken*
- *Interactieve visualisatie en visuele interactie puntenwolken*
- *Spatio-temporele puntenwolken*

Dit project richt zich op een verdere verdieping van één van de bovengenoemde aanbevelingen: standaardisatie puntenwolken.

Daarbij is voor gekozen voor de volgende aanpak. Deelname aan Open Geospatial Consortium (OGC) Point Cloud Domain Working Group (DWG).
'The Point Cloud DWG is being established to address interoperability challenges with point clouds. This group will facilitate discussion of the requirements that define different point cloud encoding and/or optimization formats and how interoperability between content can be better achieved. Specifically, the Point Cloud DWG will pursue the following activities:

- *discuss different point cloud encoding and optimization formats to understand the pros and cons for each;*
- *discuss the types of OGC services that might be useful in providing point cloud data through a standardized interface;*
- *define best practices for managing, processing and sharing point cloud data;*
- *define areas for standardization and create necessary Standards Working Groups to address the gaps in the OGC standards baseline.'*

Verder is er en duidelijke link met onderwijs: discussies en presentaties tijdens OGC Point Cloud DWG bijeenkomsten vormen bronmateriaal voor colleges van de Master of Science Geomatics (www.geomatics.tudelft.nl).

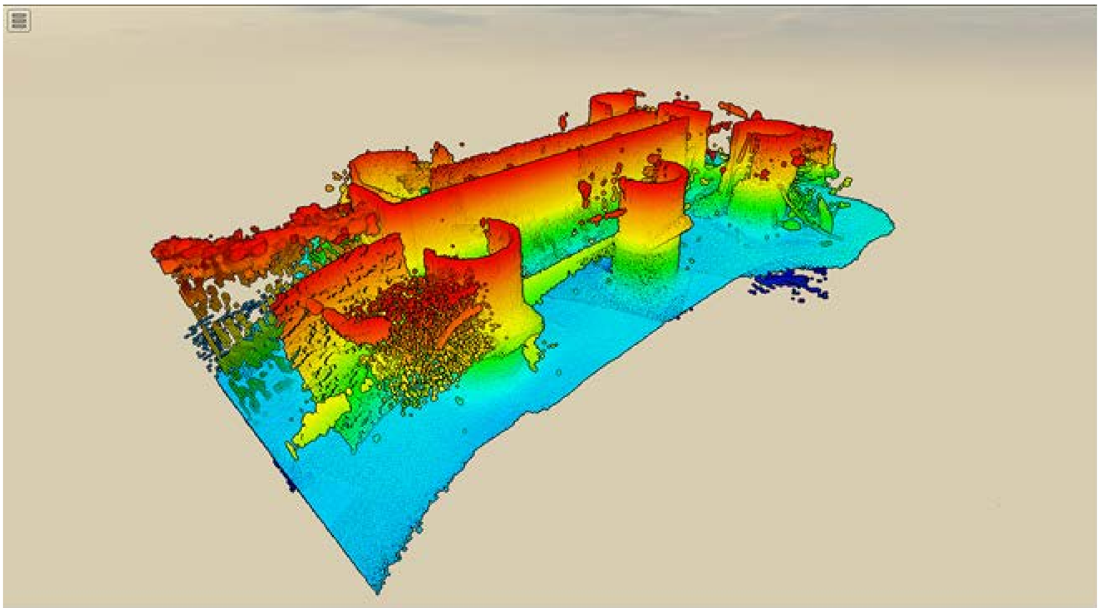
Daarnaast is Stella Psomadaki in november 2016 in de praktijk afgestudeerd bij Deltares op het onderwerp 'Using a Space Filling Curve for the Management of Dynamic Point Cloud Data in a Relational DBMS'.

De OpenGeospatial Technical Commission heeft in 2016 een viertal bijeenkomsten gehad. De 98ste in maart in Washington DC, de 99ste in juni in Dublin, de 100ste in september in Orlando, en de 101ste in december in Taichung. Peter van Oosterom heeft een voordracht gehouden in Dublin. Deze bijeenkomst in Dublin, maar ook de andere bijeenkomsten (met uitzondering van Taichung) zijn door Edward Verbree op afstand gevolgd via een 'GotoMeeting'. Alle voordrachten (ppt/pdf), maar ook de recordings (MP4) van de presentaties zijn vrij beschikbaar voor OGC leden via het OGC portal (https://portal.opengeospatial.org; Membership; Member Login; Technical Committee; Meetings

2016). Het verdient aanbeveling een aantal van deze presentaties te bestuderen; met name degenen die hieronder in de 'aantekeningen' verder besproken worden.
Daarnaast zijn er ook voor andere gremia voordrachten verzorgd op basis van reeds eerder verricht onderzoek 'Exploratieve Puntenwolken' en de resultaten van andere sub-thema's binnen het oorspronkelijke onderzoeksvoorstel.
Voordrachten op basis van reeds eerder verricht onderzoek ('Exploratieve Puntenwolken') en resultaten van andere subthemas binnen dit onderzoeksvoorstel 'Direct gebruik van puntenwolken'.
Zo heeft Peter van Oosterom tijdens de 'Joint 3D Athens Conference' 18-21 oktober, Athene, Griekenland een plenaire keynote presentatie verzorgd (zie Sectie 5 van dit rapport).

Verantwoording:

Dit verslag is een 'mengelmoes' van Nederlands en Engels en rijkelijk voorzien van de onvermijdelijke drie- en vier-letterige acroniemen. Het materiaal is afkomstig van de genoemde OGC portal en vanuit eigen aantekeningen.



Figuur 1: 'Direct gebruik van puntenwolken'

2 OGC Point Cloud DWG[1] bijeenkomsten

De OGC Point Cloud DWG is gestart in 2015, vanuit het gevoel dat: "time was ripe and that stars aligned".

- Technology is rapidly democratizing.
- Consensus that the focus should be on Point Cloud access and processing:
  - with a focus on the latter, that is where the power/business will be;
  - (data acquisition is not the money maker, it's the added value you can bring to the Point Cloud.
- Market players were aligning on the message and way forward:
  - beginning of overlap with existing standards;
  - need for a consensus process to manage the Point Cloud roadmap.

De Point Cloud DWG is opgericht vanuit de volgende gedachte:

- Probably better not try to standardize point clouds at database level, but rather focus on webservices level (more support/ partners expected).
- Problem Statement OGC Point Cloud DWG:
  - point cloud data has often been overlooked;
  - stored in many formats;
  - many domains such as LiDAR, Elevation, Seismic, Bathymetric, Meteorological, and Fixed/Mobile consumer sensors.
- Examples de facto standards: ASPRS LAS, Sensor Independent Point Cloud (SIPC) based on HDF5.
- Greater interoperability between point cloud datasets and interoperate with other OGC standards.

2.1 98ste OGC Technical Meeting[2]

Agenda:

1. Update on the Point Cloud Survey - Stan Tillman, Intergraph;
2. Improving Access to Point Cloud Data - Keith Ryden, Esri;
3. Visibility analysis with the 3D skeleton of a point cloud - Jantien Stoter, TUDelft;
4. Precise spatial and radiometric processing of the Level 0 point cloud - Chuck Heazel, WiSC Enterprises;
5. Advanced Format Standards for Point Cloud Data - Michael Ingram, Harris;
6. Serving LiDAR Through Existing OGC Standards - Scott Pakula, Pixia.

Aantekeningen:

Ad 1. Update on the Point Cloud Survey - Stan Tillman, Intergraph;

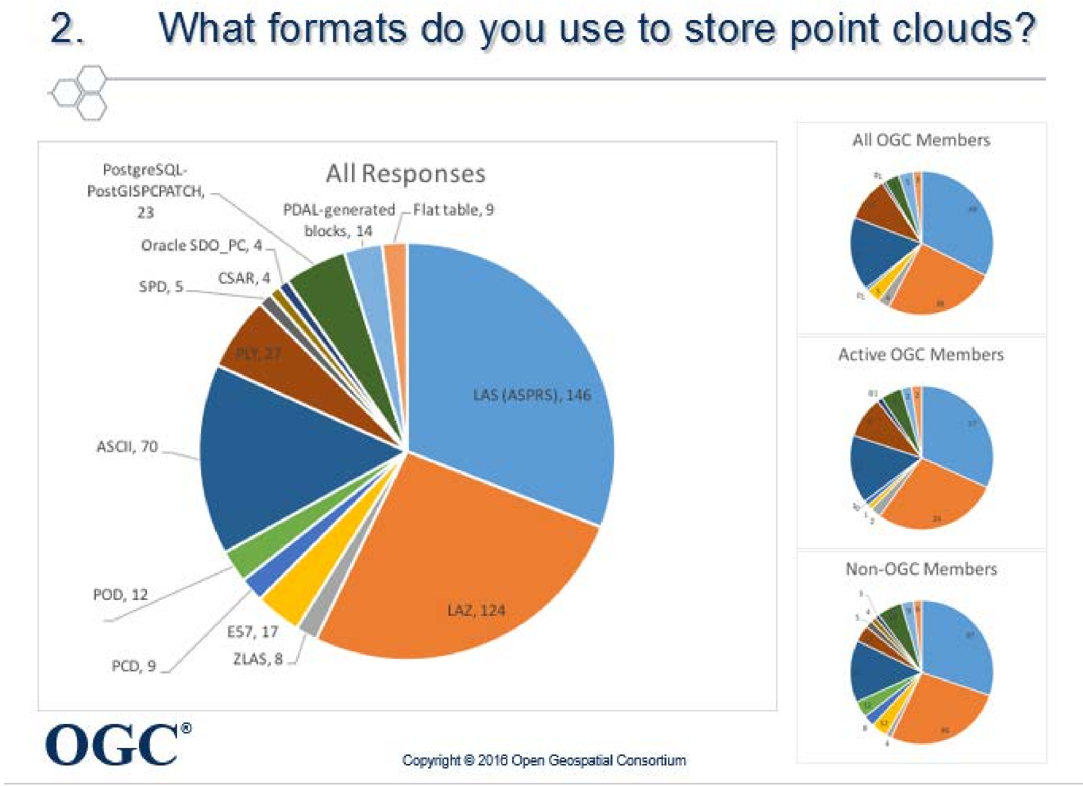
Point Cloud Survey 163 responses. Voornaamste conclusies:

1. What are your major sources for surface scan point clouds?

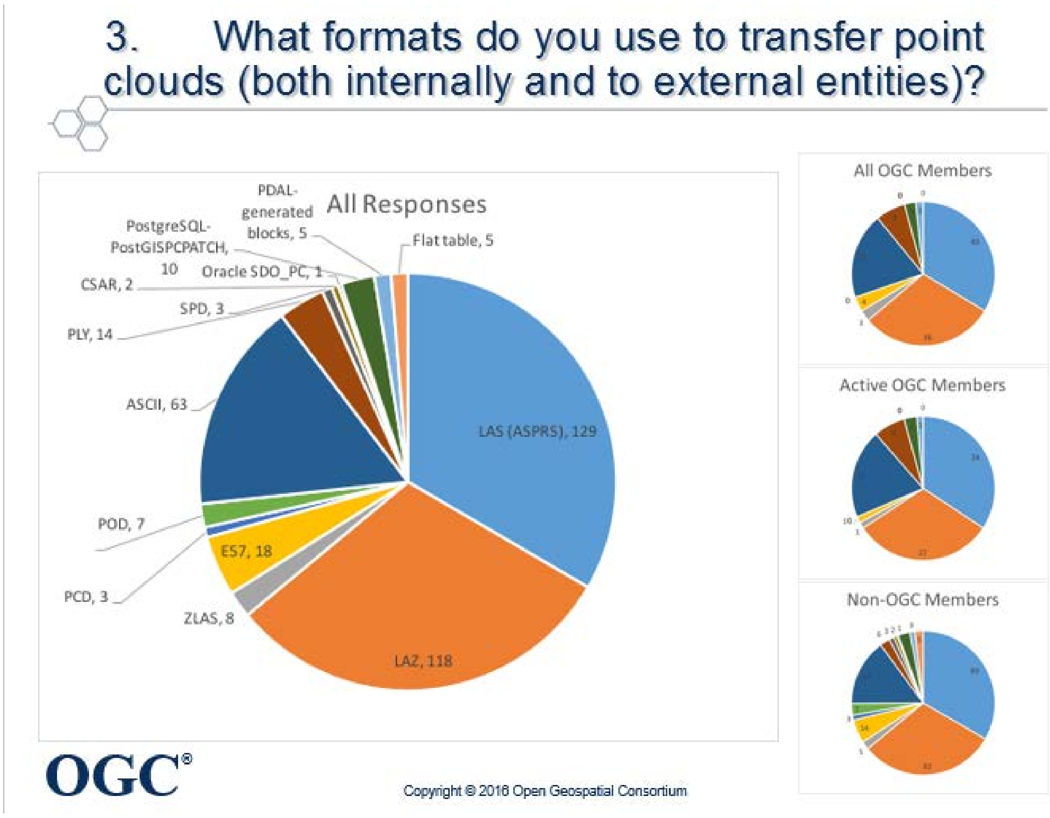
   Top of the list: Terrestrial Lidar (including Mobile Mapping); Indoor Laser Scanning; SONAR (single and multi-beam echo's); Photogrammetry; RADAR (PS-InSAR)
2. What formats do you use to store point clouds?

[1] http://www.opengeospatial.org/projects/groups/pointclouddwg

[2] Washington, USA (07-11 maart 2016): http://ogcmeet.org/pastevents/1603tc/



3. What formats do you use to transfer point clouds (both internally and to external entities)?



4. What are your most common use cases for point clouds?

Visualization:	111
GIS:	87
Digital Terrain Modelling:	132
Feature Extraction:	104

Ad 2. Improving Access to Point Cloud Data - Keith Ryden, Esri

Some nice statements

- Point Cloud data is typically Big Data LiDAR data in a collection of LAS datasets are one example; It's big if you don't want to move it ….; Bring the processing to the data….
- Start simple
  - No changes to the LAS file.
  - Add a "sidecar" file that has all the optimization information in it: Metadata, Classification, Statistics, Spatial Indexing, Reorganize records, Get more complicated later.
  - Redefine the LAS file storage: SQL is your friend – RDBMS and SQLite; Make it continuous and scalable; Compression.
- LAS files consist of variable length records full of individual fixed format point records.
- Each record has its length and ID encoded in its' header.
- End up scanning the file to find stuff…
- We can optimize access to LAS data [ZLAS] by spatially indexing the variable length point records based on the extent of the points, using the point record ID as the key: Know you know which variable length point records are of interest, Read only those, and scan the internal points.
- Hash or checksum on the LAS data file? – Might be a good idea – lets you know if the file has been changed
- If individual points are scattered through the LAS file, you can optimize access by rearranging the points; Cluster points spatially related together, Reorganize points based on application requirements, Moves points into different records, but doesn't change the LAS format.

Ad 3. Visibility analysis with the 3D skeleton of a point cloud - Jantien Stoter, TUDelft

- Jantien Stoter presented PhD work Ravi Peters: Visibility analysis with the 3D skeleton of a point cloud.

Ad 4. Precise spatial and radiometric processing of the Level 0 point cloud - Chuck Heazel, WiSC Enterprises

- No remarks

Ad 5. Advanced Format Standards for Point Cloud Data - Michael Ingram, Harris

- Michael Ingram addressed some problems with LAss/Laz, albeit this is the only approved point cloud compression format by the US DoD. He showed the advantages of Sensor Independent Point Cloud (SIPC) based on HDF5. Conclusion: HDF5optimal format for any point cloud data: self-discoverable, hierarchical, Object oriented, Used by NASA; File converters: MATLAB and C++ executables; SIPC Viewers: IDL, PointVIS, Viper; SIPC Benefits: Data is in ECEF 3D Cartesian coordinate system

Ad 6. Serving LiDAR Through Existing OGC Standards - Scott Pakula, Pixia.

- No remarks

2.2 99ste OGC Technical Meeting[3]

Agenda:

1. Update on the Point Cloud Survey and way forward - Stan Tillman, Intergraph;
2. OGC Testbed-12 LiDAR Streaming ER - Simon Jirka & Christoph Stasch, 52°North;
3. Managing dynamic point clouds - Peter van Oosterom, TUDelft;
4. Sorted Pulse Data (SPD) File Format: Thoughts and Opinions - Pete Bunting, Aberystwyth University;
5. On-going work for extending LASzip - Martin Isenburg, rapidlasso;
6. SpectrumML: Data Model and Encoding for Electromagnetic Fields - Lance McKee, Independent Consultant.

Tijdens de 99ste OGC TC heeft de Point Cloud DWG haar top-3-activiteiten voor de nabije toekomst bepaald. De bijeenkomst vond plaats op 21 juni 2016 bij UCD (University College Dublin) in Ierland. Als eerste presenteerde Stan Tillman (Intergraph) namens zijn medevoorzitters Jan Boehm (University College London) en Peter van Oosterom (TU Delft) de resultaten van de recent gehouden Point Cloud Survey over onder meer opslag, uitwisseling, omvang, toepassingen en gebruikte software voor point cloud data. Hiervoor wordt verwezen naar Sectie 5. In totaal zijn er bij de OGC 188 ingevulde enquêtes ontvangen van 175 verschillende organisaties. Deze zijn allemaal online beschikbaar via de (niet zo fijne) URLs in de spreadsheet op

- https://docs.google.com/spreadsheets/d/1_6389UlkIblWyneY5WbbONMMJZiNaeUmcs_iG6olS0/edit?usp=sharing

en de analytics op

- https://docs.google.com/forms/d/19eQVHYaylKwBHOQTlVrlitxk8AXm1xbzmfCWyeHKICs/viewanalytics

Samenvattend kan worden gesteld dat er een grote variatie aan formats en tools waarneembaar is en dat er ondanks de al wel beschikbare de facto ‘standaarden’ bij veel organisaties behoefte is aan verdere officiële standaardisatie. Op basis van de enquête heeft de Point Cloud DWG de volgende drie prioriteiten aangeven:

1. Verdere samenwerking met de American Society of Photogrammetry and Remote Sensing (ASPRS) op het gebied van het LAS-formaat;
2. Verkennen van HDF5 als formaat voor Point Cloud-data; en
3. Interoperabele streaming van Point Cloud webservices.

Naast het uitstippelen van de toekomstige speerpunten van de DWG waren er verschillende andere point cloud presentaties in deze sessie, waaronder: ‘On-going work for extending LASzip’ (Martin Isenburg, rapidlasso), ‘OGC Testbed-12 LiDAR Streaming ER’ (Simon Jirka & Christoph Stasch, 52°North), ‘Sorted Pulse Data (SPD) File Format: Thoughts and Opinions’ (Pete Bunting, Aberystwyth University) en een bijdrage uit Nederland ‘Towards vario-scale and dynamic point clouds’ (Peter van Oosterom, Stella Psomadaki, Theo Tijssen/TU Delft & Fedor Baart Deltares). Hiervoor wordt verwezen naar Sectie 5. Alle presentaties zijn beschikbaar op de OGC-website. Al met al een zeer vruchtbare bijeenkomst en het begin van echte standaardisatie op het gebied van point clouds.

[3] Dublin, Ierland (20-24 juni 2016): http://ogcmeet.org/pastevents/1606tc/

Figuur 2: University College Dublin, gastheer van het 99ste Open Geospatial Consortium

2.3 100ste Technical Meeting[4]

Agenda:
1. Update on LAS as a Community Standard - Stan Tillman, Intergraph;
2. Streaming large point clouds - Howard Butler, Hobu, Inc.;
3. Techniques for Point Cloud Compression - Robert A. Cohen, Anthony Vetro (Mitsubishi Electric Research Labs) & Koji Wakimoto (Mitsubishi Electric Corporation);
4. Visualizing massive point clouds with 3D Tiles - Patrick Cozzi, AGI;
5. 3D data compression and streaming - Eric Robertson, PAR Government Systems Corporation
6. OGC Assessment of the Point Cloud survey - Scott Simmons, OGC.

Aantekeningen:
- Streaming large point clouds - Howard Butler, Hobu, Inc.: zeer inzichtelijke presentatie; ook voor colleges e.d.;
- Techniques for Point Cloud Compression - Robert A. Cohen, Anthony Vetro (Mitsubishi Electric Research Labs) & Koji Wakimoto (Mitsubishi Electric Corporation): Extend concepts from block-based image and video compression to compress point cloud attributes; Interessante, maar lastig te volgen presentatie;
- Visualizing massive point clouds with 3D Tiles - Patrick Cozzi, AGI: Zeer interessante spreker, oprichter van Cesium. Heeft tal van boeken geschreven over Virtual Globes, WebGL en OpenGL. Goede presentatie met o.a. goede intro over 3D Tiles;
- 3D data compression and streaming - Eric Robertson, PAR Government Systems Corporation: goede, maar lastig te volgen presentatie.

2.4 101ste Technical Meeting[5]

Agenda:
1. Testbed-12 LiDAR Streaming Engineering Report (ER) - Simon Jirka, 52 North;
2. Point clouds as a data service using WPS - Adam Steer, The Australian National University;
3. Update on the LAS 1.4 submission as a Community Standard - Stan Tillman, Hexagon Geospatial (Intergraph).

[4] Orlando, Florida (19-24 september 2016): http://ogcmeet.org/pastevents/1609tc/index.php
[5] Taichung, Taiwan (04-08 december 2016): http://ogcmeet.org/

Bijeenkomst niet via GotoMeeting gevolgd (3.00 uur lokale tijd ..), aantekeningen op basis van presentaties:

Ad 1. Testbed-12 LiDAR Streaming Engineering Report (ER) - Simon Jirka, 52 North Testbed-12:

- ➢ Determine the optimal method to deliver LiDAR data through an OGC Sensor Observation Service;
- ➢ Provide a streaming mechanism
  - – Three deliverables: LiDAR Streaming ER; SOS server for streaming LiDAR data; LiDAR streaming client that interacts with the streaming SOS server;
- ➢ Recommendations:
  - – SOS interface is suited for serving LiDAR data in a JPIP-like streaming approach: Pre-processing needed, Rely on the GetResult operation;
  - – Recommended change: support native (binary) data formats for LiDAR data as output of the GetResult operation: Vavoid overhead for en-/de-coding the binary LiDAR data into a text-based XML document;
  - – LiDAR SOS requires slightly more business logic on the client side compared to other approaches (e.g. JPIP): client needs to identify patches and no stateful server-side business logic;
  - – Further performance evaluation;
  - – LiDAR Streaming Profile for the SOS 2.0 standard.

Ad 2. Point clouds as a data service using WPS - Adam Steer, The Australian National University

- ➢ Point data services with a WPS/PDAL stack:
  - – Why use WPS?: WPS provides a standard model for transactions; Server-side operations can be built using any system tools available; WPS-based services on other NCI data holdings are emerging (interface consistency);
  - – Why use PDAL?: Is not prescriptive about storage formats; Is well supported, extensible, flexible; Is a C library set as well as command-line applications; Gives us access to more ideas than our own about point handling.

Ad 3. Update on the LAS 1.4 submission as a Community Standard - Stan Tillman, Hexagon Geospatial (Intergraph)

- ➢ Why are we bringing LAS 1.4 into OGC?
  - – LAS is already a specification within American Society of Photogrammetry and Remote Sensing (ASPRS);
  - – But we currently cannot reference it in a normative fashion within our documents;
  - – Based on our recent survey, it is clear that LAS is still widely used both inside and outside of OGC.

Action was taken to approach ASPRS to gauge interest in obtaining a “Community Standard” stamp of approval from OGC

It was presented to the ASPRS Board for a vote.

A note from Charles Toth (President of ASPRS) that the ASPRS BOD approved, on September 11, 2016 to proceed with the community standard for LAS.

3 MSc Thesis Styliani Psomadaki

Op 8 november 2016 is Styliani (Stella) Psomadaki in november 2016 in de praktijk afgestudeerd bij Deltares op het onderwerp 'Using a Space Filling Curve for the Management of Dynamic Point Cloud Data in a Relational DBMS' (MSc Programma Geomatics for the Built Environment van de Faculteit Bowukunde). Zij is vanuit de TUD begeleidt door Peter van Oosterom (tevens afstudeerhoogleraar) en Theo Tijssen. Vanuit Deltares is zij begeleidt door Fedor Baart.
Overigens is Stella cum laude afgestudeerd.

De volledige afstudeerscriptie is te vinden op:
http://www.gdmc.nl/publications/2016/Using_SFC_for_Dynamic_Point_Clouds_thesis.pdf

Abstract
The rapid developments in the field of point cloud acquisition technologies have allowed point clouds to become an important source of information for many applications. One of the newest applications of point clouds concerns the monitoring of the coast. Many countries, among which the Netherlands, use this source of data in order to determine the changes in coastal elevations. This means that point clouds are collected every hour, day, month, year; ultimately talking about dynamic point clouds. To be able to efficiently use this plethora of data, the management of those point clouds, dynamic or not, is proven to be crucial. Point clouds, like the majority of geodata, have been traditionally managed using file-based solutions. Nevertheless, the last years database solutions have emerged. Typical examples are the point cloud extensions for PostgreSQL and the Oracle Database. Both options use a similar block-based organisation. In addition to the block based organisations, point clouds can also be managed using a flat table where each point is stored in a separate row. While the first approach is very scalable and efficient, the second is easier to implement and to update. To make the flat model scalable, a Space Filling Curve (SFC) can be used to cluster the data. Nonetheless, both approaches in their current forms, are not suited for the management of dynamic points. The reason for this is the fact that they do not consider the time dimension as part of the organisation and further insertions for the block-based approaches are not straightforward. Within this thesis a SFC approach for managing dynamic point clouds is investigated. For this, the flat model approach using an Index Organised Table (IOT) within a Relational Database Management System (RDBMS) is used. Two variants coming from two extremes of the space - time continuum are then taken into account. In the first approach, space and time are both used within the SFC (integrated approach), while in the second one, time dominates over space (non-integrated approach). Along these two approaches, two treatments of the z dimension are, also, studied: as attribute or as part of the SFC. In addition to that, building on the coastal monitoring applications, the most important queries are identified: space - time, only time, only space. The efficiency of the implemented methodology is tested through the execution of a benchmark. Using two use cases coming from coastal applications, the benchmark is executed once for daily and once for yearly data. The results show that the SFC approach is an appropriate method for managing dynamic point clouds. Furthermore, the integrated approach is the most suitable way to proceed. Achieving scalability, time efficiency and dynamic insertions can be achieved for various use cases.

Haar afstudeerpresentatie is opgenomen in Sectie 5.

4 Conclusies

Op basis van eerder verricht onderzoek omtrent ‘exploratieve puntenwolken’ en de bevindingen van de OGC Point Cloud DWG hebben wij aan Rijkswaterstaat de volgende aanbevelingen:

- De eerste aanbeveling aan Rijkswaterstaat is bij één van de komende Point Cloud DWG Meetings in 2017 de relevantie van standaardisatie van Point Clouds binnen de werkprocessen van Rijkswaterstaat te benadrukken. Een goed handvat biedt de opzet van de Point Cloud Survey, uitgevoerd door de DWG co-chairs Stan Tillman, Peter van Oosterom en Jan Boehm. Min of meer dezelfde vragenlijst, maar dan specifiek gericht op de inwinning, opslag, verspreiding, en presentatie van puntenwolk data binnen Rijkswaterstaat, de toeleveranciers, en de afnemers. De eerst volgende OGC Technical Commitee bijeenkomst vindt plaats van 20-24 maart in Delft, zie: http://www.opengeospatial.org/event/1703tc.

- De tweede aanbeveling aan Rijkswaterstaat is om, vanuit deze betrokkenheid, veel meer sturend op te treden bij het belang van directe benutting van de puntenwolkdata binnen de werkprocessen van Rijkswaterstaat. Tijdens de Point Cloud DWG zijn tal van voorbeelden getoond waarbij directe (visuele) analyse en (interactieve) presentatie van puntenwolkdata een direct antwoord biedt op de informatiebehoefte. Dit verbetert de doorlooptijd, inzichtelijkheid en hergebruik van data.

- De derde aanbeveling aan Rijkswaterstaat is om, vanuit dit inzicht, veel meer sturend te zijn op de inzet en ontwikkeling van Point Cloud standaardisatie activiteiten.

Bijlagen

A. Plenaire keynote Peter van Oosterom, ‘Joint 3D Athens Conference’, 18-21 oktober, Athene, Griekenland

B. Point Cloud Survey – OGC Point Cloud DWG Presentatie Stan Tillman (Intergraph) namens zijn medevoorzitters Jan Boehm (University College London) en Peter van Oosterom (TU Delft), 21 juni 2016, Dublin, Ierland

C. ‘Towards vario-scale and dynamic point clouds’ Peter van Oosterom, Stella Psomadaki, Theo Tijssen/TU Delft & Fedor Baart/Deltares, 21 juni 2016, Dublin, Ierland

D. Afstudeerpresentatie Styliani (Stella) Psomadaki, 8 november 2016, Delft

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.

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

Bijlage A

Plenaire keynote Peter van Oosterom 'Joint 3D Athens Conference',
18-21 oktober, Athene, Griekenland

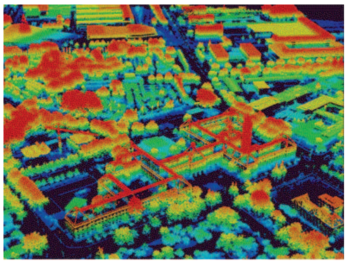
nD-PointClouds
a model for deeply integrating space, time and scale

14-12-2016

Peter van Oosterom

Plenary lecture at the Joint 3D Athens Conference,
18-21 October, Athens, Greece

Overview



- motivation
- scale as dimension
- functionality
- data management
- standardization (if time allows)
- conclusion

acknowledgements: based on joint work with Edward Verbree, Theo Tijssen, Oscar Martinez-Rubi, Mike Horhammer, Stella Psomadaki, Xuefeng Guan, …

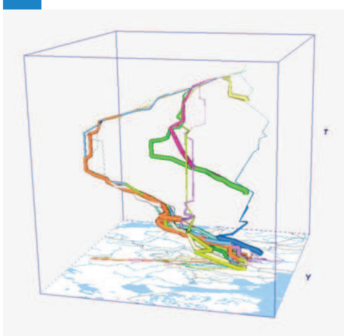
kind of sequel on last years keynote at JIGC 2015, Kuala Lumpur: Realistic benchmarks for point cloud data management systems

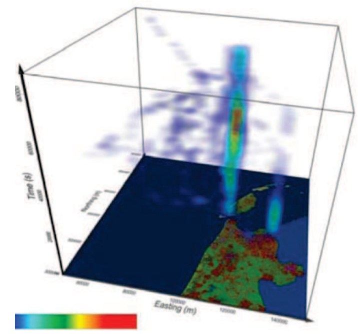
Motivation

- point cloud data sets are often used for monitoring
  → dynamic point clouds
  → time as additional organizing dimension
- organizing point cloud data in LoD's/importance levels is an approach to manage large data sets
  → LoD: discrete (multi-scale) or continuous (vario-scale)
  → scale treated as additional organizing dimension
- how to manage higher dimensional point clouds (4D, 5D)

Time as dimension

time more obvious: well-known space-time cubes

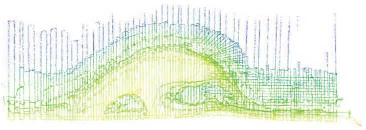




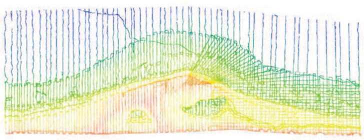
Dynamic Point Clouds

- point clouds are generated every day, hour, minute
- repeated scans of the same area → dynamic
- time as selective as the spatial component or needed in integrated space – time selections
- current DBMS solutions designed for static point clouds
- management is still a challenge
- example Sand Engine, time series Dutch coast, Deltares (see Psomadaki et al, Friday)

2011

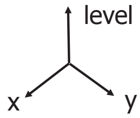


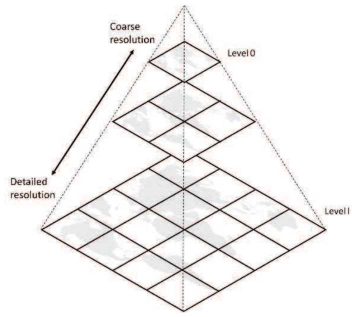
2015



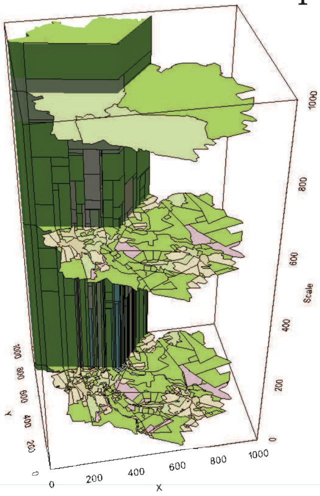
Scale as dimension

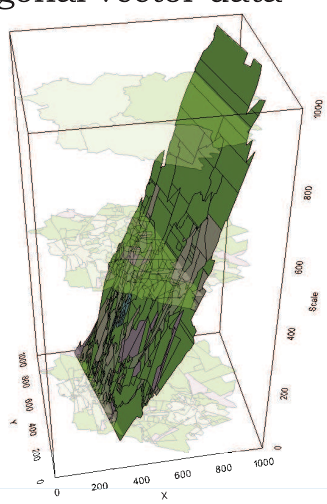
- less obvious than time
- data pyramids (Level of Detail/ Multi-scale)
- well-known from raster data
- results in discrete number of levels (multi-scale)
- level could be considered as additional dimension





Vario-scale with polygonal vector data





Representations of space, time, scale ...after grid/voxel or object/vector

- new 3rd representation: nD point cloud (PC)
- many scientific domains (spatial): geography, medicine, physics, astronomy, hydrology, architecture, archaeology, arts, CAD, social media/ moving objects, gaming…
- deep integration space/time/scale
  1. more efficient, store, exchange, compute
  2. more functionality (smooth zoom/ analysis)
- nD PC in whole processing chain: acquisition, DBMS, analysis, simulation, dissemination, visualization,…
- BIG spatial data: 35 trillion points (in astronomy, geo-info)

Overview

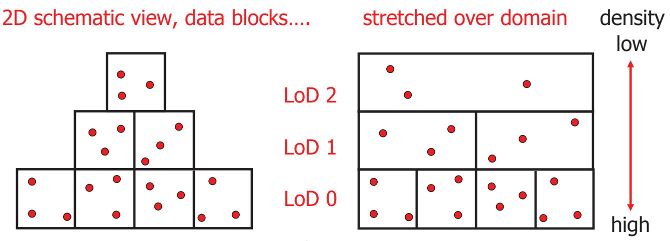
- motivation
- scale as dimension
- functionality
- data management
- standardization
- conclusion

LoD/multi-scale Point Cloud

- data pyramid (Level of Detail/ Multi-scale) in analogy with raster
- imagine a fine 2D (or 3D) bottom level grid to organize the points
- option is after every 4 points in cell move 5th point to parent cell (for 2D organization and every 9th point in case of 3D), recursively bottom-up filling the cell/blocks at higher levels
- results in data pyramid → discrete number of levels (multi-scale)
- Note: depending on input data distribution, some areas my reach higher levels than others

Point cloud data pyramid

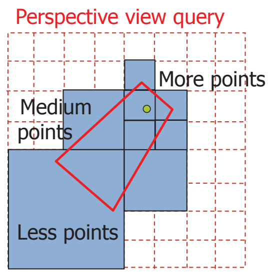
- overview queries just want top-subset
- detailed queries part of bottom-subset
- organize in data pyramid



every next higher level, density 2^k times less (2D → 4, 3D → 8)

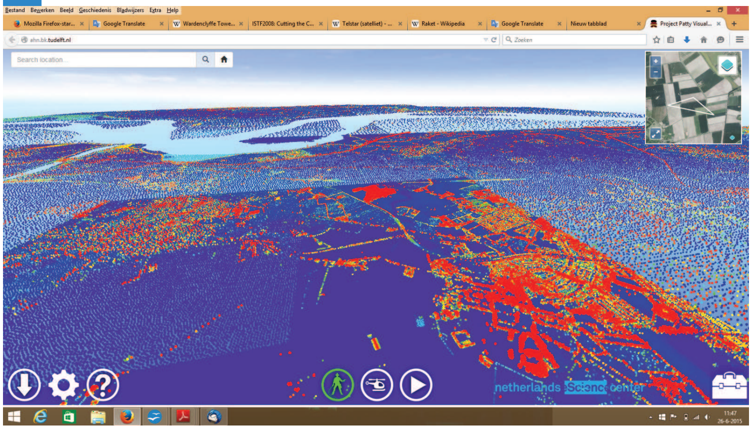
Data pyramid/multi-scale

- allows fast spatial searching including LoD selection
- the further away from viewer the lesser points selected (i.e. the higher level blocks/points)
- drawbacks:
  1. discrete number of levels
  2. bottom-up filling, unbalanced top
  3. point random assigned to level



Discrete LoD's are visible…

http://ahn2.pointclouds.nl: 640.000.000.000 points on-line 3D viewer



Data pyramid alternatives

- not random points, but more characteristic points move up (more important), some analysis needed; e.g.:
  1. compute local data density → more dense less important
  2. compute local surface shape → more flat less important
  3. other criteria, data collection/application dependent (intensity)

  (combine into) one imp_value of point → better than random

- not bottom-up, but top-down population, make sure that top levels are always filled across complete domain (lower levels may not be completely filled)

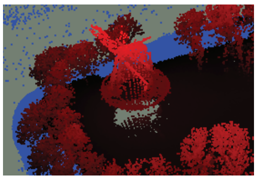
Further improvements … beyond discrete levels

- might result in artefacts when looking at perspective view image (possible 'see' blocks of different levels)
- also not optimal within block (near viewer perhaps not enough points, further from viewer perhaps too much points)
- would a true vario-scale option possible?

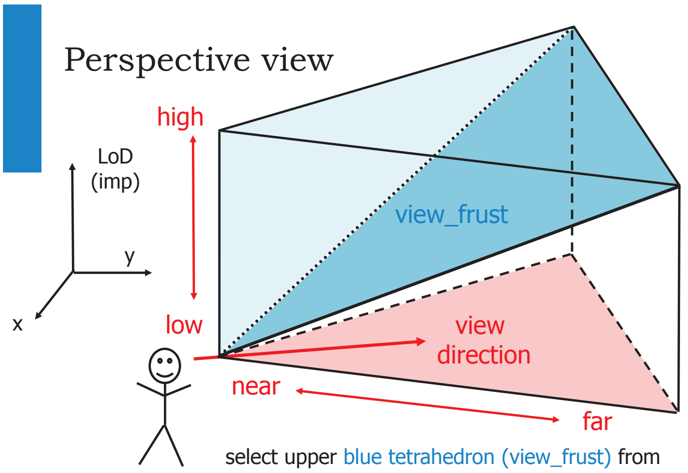
→ Vario-scale geo-info research at TU Delft

Vario-scale for point cloud data

- lesson from vario-scale research: add one continuous dimension to the geometry to represent scale (2D data vario-scale represented by 3D geometry)
- apply this to point cloud data
  1. compute the imp value
  2. add this as dimension, either x,y,imp (z and others attributes) or x,y,z,imp (and others as attributes)
  3. Cluster/index the 3D or 4D point
  4. Define perspective view selections, view frustum with one more dimension: the further, the higher imp's



Perspective view



select upper blue tetrahedron (view_frust) from prism-part of vario-scale x,y,imp point cloud data cube

Normal view frustum selection and streaming based on importance

- view frustum selection

```
select point
from point_cloud
where overlaps (point, view_frust)
```

- ordered on importance for streaming

```
select point
from point_cloud
where overlaps (point, view_frust)
order by imp desc;
```

(or distance from tilted plane)

Delta queries for moving and zoom in/out

- select and send new points:
  `point in new_frust and point not in old_frust`
- find and drop old points:
  `point in old_frust and not in new_frust`
- note this works form both
  1. changing view position x,y(,z)
  2. zooming in or out ('view from above', imp-dimension)
- optional to work at point or block granularity
  (in selection and server-client communication)

Overview

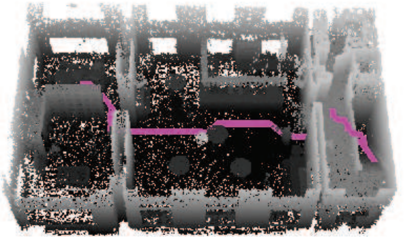
- motivation
- scale as dimension
- functionality
- data management
- standardization
- conclusion

Point cloud analysis

- benefits:
  - no conversion time
  - no data loss
  - analysis may be better
  - LoD continuous (raster pixels factor 2, vector hard)
  - very realistic representations (e.g tree with leaves)
- drawbacks:
  - lot of data
  - redevelop algorithms
- have it as option
  together with conversions PC ←→vector, PC←→ raster

Types of analysis, direct point clouds

- solar energy potential
- viewshed/ line-of-sight
- 3D routing (e.g. drone; see Rodenberg et al, Friday)
- change detection (deformations)
- volume analysis computations
- hydrology/ flow over surface
- vegetation analysis
- continuous LoD
  also for analysis
  not only visualization



Point cloud base functionality (1/2)

1. simple range/rectangle filters (of various sizes)
2. selections based on points along a linear route (with buffer)
3. selections of points overlapping a 2D polygon
4. selections based on the attributes such as intensity I (/RGB)
5. multi-resolution/LoD selection (select top x%)
6. sort points on relevance/importance (support streaming)
7. slope orientation or steepness computation
8. compute normal vector of selected points
9. convert point cloud to TIN representation
10. convert point cloud to Grid (DEM)
11. convert point cloud to contours
12. k-nearest neighbor selection (approx or exact)
13. selection based on point cloud density
14. spatial join with other table; e.g. 100 building polygons
15. spatiotemporal selection queries (specify space+time range)
16. temporal differences computations and selection
17. compute min/max/avg/median height in 2D/3D area

Point cloud base functionality (2/2)

18. hill shading relief (image based on point cloud/DEM/TIN)
19. view shed analysis (directly on point cloud with fat points)
20. flat plane detection (and segmentation point, add plane_id)
21. curved surface detection (cylinder, sphere patches, freeform)
22. compute area of implied surface (by point cloud)
23. compute volume below surface
24. select on address/postal code/geographic names (gazetteer)
25. coordinate transformation RD-NAP - ETRS89
26. compute building height using point cloud (diff in/outside)
27. compute cross profiles (intersect with vertical plane)
28. combine multiple point clouds (Laser+MBES)
29. volume difference between design (3D polyhedral) surface and point could
30. detect break line point cloud surface
31. selection based on perspective view (point cloud density)
32. delta selection of query 31, moving to new position

Overview

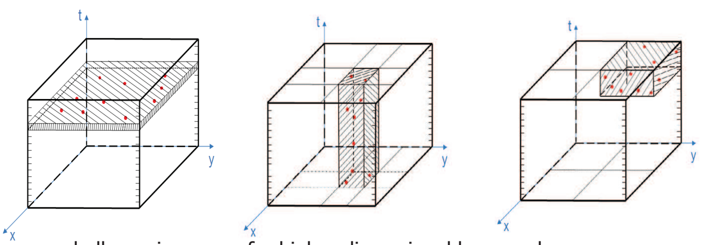
- motivation
- scale as dimension
- functionality
- data management
- standardization
- conclusion

nD PC data management

- management of nD PC data, starts by defining
  - dimensions (and their roles/priorities in the points)
  - associated attributes
- dimensions are main drivers for data organization, clustering, indexing, subdivision (for parallel processing), compression, blocking/ caching and streaming of data
- investigate various data management options
  - kd-tree based organization (no scaling issues of different dimensions)
  - organization based on simplices (e.g. triangle/tet bins, Sierpinski)
  - *integrate dimension values in 1 value via Space Filling Curve (SFC): Morton, Hilbert, and relation to quadtree*

Different blocking scheme's for space-time (or space-scale) cube

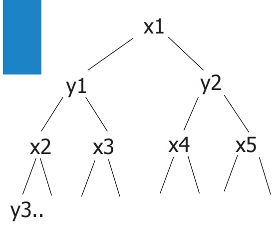
16x16x1 4x4x16 8x8x4

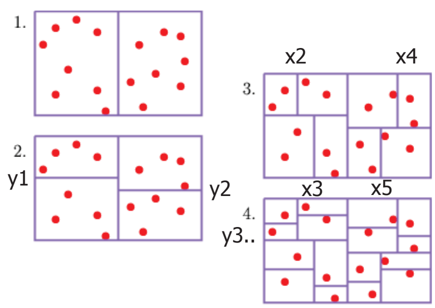


- challenge increases for higher dimensional hyper-cubes:
  - 4D: 2D space-time-scale, 3D space-time, 3D space-scale
  - 5D: 3D space-time-scale

kd-tree

http://graphics.stanford.edu

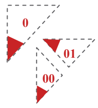


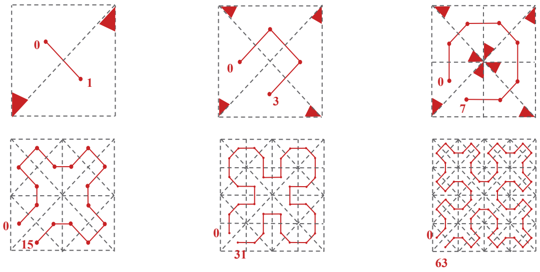


- alternating x, y split
- needs resorting (again and again)
- works in nD (alternative x,y,z split, or x,y,z,t split, or ..)
- may get unbalanced, not dynamic
- dimensions metric independent (scaled, distributed differently)
- used by László Dobos et al (cosmological particles, Bridget Falck)

Simplices based

- Sierpinski Curves: start with two triangles (2D) and split recursively
- works in 3D (tets) and higher?





- Elliot Sefton-Hash uses triangle bins with quadsplit (planetary data)

nD-PointClouds data management

- modelling theory for nD point cloud data
- tools to support modelers, developers and users in point cloud data organization design decisions for (given 1. data sets and 2. required functionalities in applications):
  - what are the dimensions,
  - what are the attributes,
  - what type of organization: Morton-code/ kd-tree/ nD simplices-part,
  - what relative scale of various dimensions,
  - parameters such as clustering/ blocking size,
  - what compression,
  - what approach and level of parallelism (incl. hardware aspects),

→ Modeling workbench

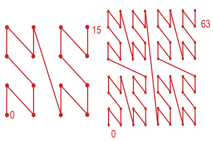
In detail: Space Filling Curves (SFCs)

- apply linear ordering to a multidimensional domain (spatial clustering)
- organize a flat table efficiently
- full resolution keys: avoid storing x,y[,z] + t/l → recovered from SFC key

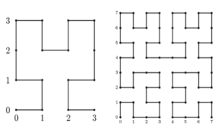
- use Index Organized Table (data stored in the B-Tree index)
- queries need to be re-written to SFC-ranges, benefit from spatial clustering → efficient

- SFCs based on hyper-cubes
  - Morton/Hilbert both **nD and quadrant recursive**
  - Consider relative scaling of dimensions
  - Space reserved on the hypercube for future data

Morton (Peano)



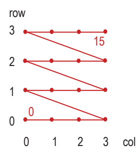
Hilbert



Some Space Filling Curves

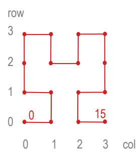
space filling curve used for block/cell creation
ordering or numbering of cells in kD into 1D using bi-jective mapping

Row (first y, then x)

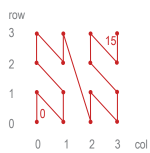


Default nD-array
(non clustering)

Hilbert



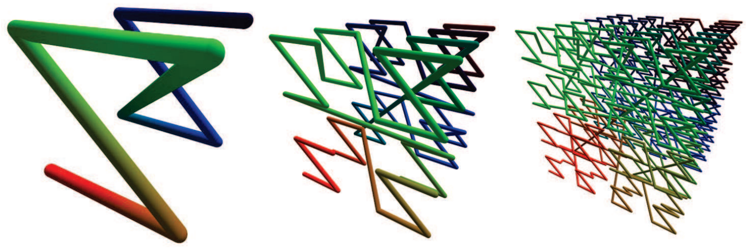
Peano



3D Morton curve

illustrations from http://asgerhoedt.dk

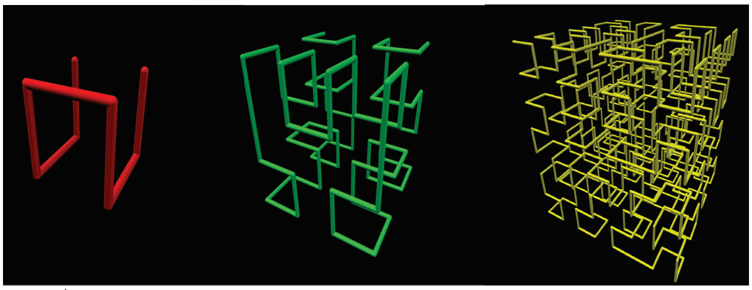
2x2x2 4x4x4 8x8x8



3D Hilbert curve

illustrations from Wikimedia Commons

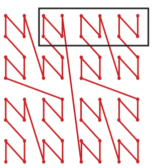
2x2x2 4x4x4 8x8x8



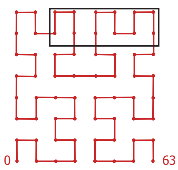
Average number of clusters for all possible range queries

- Faloutsos and Roseman, 1989
- N=8, number of Clusters for a given range query:

N*N GRID	HILBERT	PEANO	I
2*2	1.11	1.22	.11
4*4	1.64	2.16	.52
8*8	2.93	4.41	1.48
16*16	5.60	9.29	3.69



Peano (3 ranges)



Hilbert (2 ranges)

Use Hilbert/Morton code

- two options:
  1. flat table model create b-tree index on SFC code
  2. walk the curve create point cloud blocks
- better flat table model (tested with Oracle):
  - not use the default heap-table, but an indexed organized table IoT (issue with duplicate values → CTAS distinct)
  - no separate index structure needed → more compact, faster
- best (as no redundancy):
  - not x, y, z, time, LoD attributes, but just high-res SFC code (as x, y, z coordinates and time, LoD can be obtained from code)

SQL DDL for index organized table

- Oracle:

```
CREATE TABLE PC_demo (hm_code NUMBER PRIMARY KEY)
    ORGANIZATION INDEX;
```

- PostgreSQL, pseudo solution, not dynamic (better available?):

```
CREATE TABLE PC_demo (hm_code BIGINT PRIMARY KEY);
CLUSTER pc_demo ON pc_demo_pkey;
```


SFC code technique outline

A. define functions for given square/cubic/… nD domain:

1. Compute_Code(point, domain) → Code; (for storage)
2. Overlap_Codes(query_geometry, domain) → Ranges; (for query)

B. add SFC Code during bulk load or afterwards

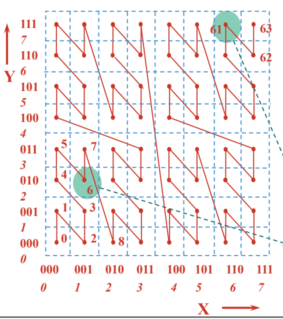
- or even replace point coordinates

C. modify table from default heap to b-tree on Code

SFC code (corresponds to Quadtree in 2D, Octree in 3D, …)

Compute_Code (point, domain) → Morton_code / Peano key / Z-order

- bitwise interleaving x-y coordinates
- also works in higher dimensions (nD)



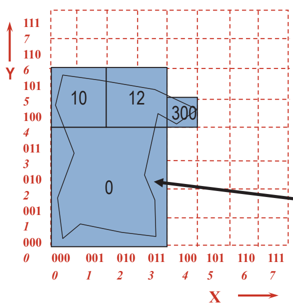
two examples of Morton code:

x= 110, y=111 → xy= 111101 (decimal 61)

x= 001, y=010 → xy= 000110 (decimal 6)

Overlap_Codes (query_geometry, domain) → Morton_code_ranges

- based on concepts of Region Quadtree & Quadcodes
- works for any type of query geometry (point, polyline, polygon)
- also works in 3D (Octree) and higher dimensions



Quadcode 0: Morton range 0-15
Quadcode 10: Morton range 16-19
Quadcode 12: Morton range 24-27
Quadcode 300: Morton range 48-48
(Morton code gaps resp. 0, 4, 20)

query_geometry, polygon

Note : SW=0, NW=1, SE=2, NE=3

Overlap_Codes(), recursive function Pseudo code

```
Overlap_Codes(query_geometry,domain,parent_quad) def
for quad = 0 to 3 do
  quad_domain = Split(domain, quad);
  curr_quad_code = parent_quad+quad;
  case Relation(query_geometry, quad_domain) is
    quad_covered:  write Range(curr_quad_code);
    quad_partly:   Overlap_Codes(query_geometry,
                     quad_domain, curr_quad_code);
    quad_disjoint: done;
```

notes:
- number of quads 2^k (for 2D: 4, for 3D: 8, etc.)
- quad_covered with resolution tolerance
- Range() translates quadcode to Morton range: start-end
- above algorithm writes ranges in sorted order (eg linked list)

Create ranges & post process (glue)

```
Overlap_codes(the_query, the_domain, '');
Glue_ranges(max_ranges);
```

Overlap_codes() creates the sorted ranges (in linked list).
result can be large number of ranges, not pleasant for DBMS
query optimizer gets query with many ranges in where-clause

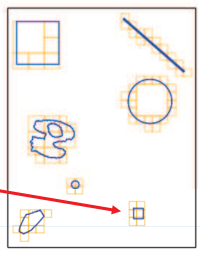
reduce the number of ranges to 'max_ranges' with Glue_ranges()
(which also adds unwanted codes)

```
Glue_ranges(max_ranges) def
  Num_ranges = Count_ranges();
  Remove_smallest_gaps(num_ranges – max_ranges);
```

notes:
- gaps size between two ranges may be 0 (no codes added)
- efficient to create gap histogram by Count_ranges()

Quadcells / ranges and queries

```
CREATE TABLE query_results_1 AS (
SELECT * FROM
  (SELECT x,y,z FROM ahn_flat WHERE
  (hm_code between 1341720113446912 and 1341720117641215) OR
  (hm_code between 1341720126029824 and 1341720134418431) OR
  (hm_code between 1341720310579200 and 1341720314773503) OR
  (hm_code between 1341720474157056 and 1341720478351359) OR
  (hm_code between 1341720482545664 and 1341720503517183) OR
  (hm_code between 1341720671289344 and 1341720675483647) OR
  (hm_code between 1341720679677952 and 1341720683872255)) a
WHERE (x between 85670.0 and 85721.0)
  and (y between 446416.0 and 446469.0))
```



Query 1 (small rectangle)

Drawback of high dimensional SFC?

- nD SFC keys have benefits: space-time-scale (and perhaps even other attributes) in compact organization
- may select on multiple dimensions at same time efficiently

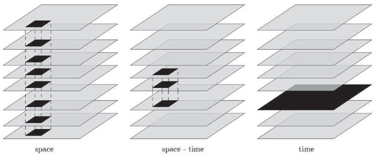
- possible drawbacks of high dimensional point cloud:
  1. need big SFC code (128 bits number or other encoding, like varchar)
  2. if just limited number of dimensions are specified for selection → other dimensions then range form min-to-max: 'tall prisms' many (empty?) cells, what are the query performance consequences

- needs further exploration
  (as the relative scaling of dimensions need attention → basis for defining cross-dimension distance → actual grouping/ clustering)

Storage model balancing

'best' organization is dependent on data and queries; e.g.

- asking for time slice (map of one moment in time)
- performing time needle query (one location trough time)
- selecting data for time interval in limited area



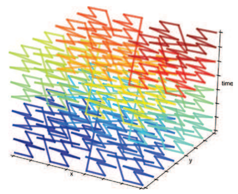
dynamic data optimizing for space/time queries contradicts:

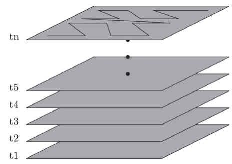
1. Points close in space and time should be stored (to some extent) close in memory for fast spatio-temporal retrieval
2. Already organized points should not be reorganized when inserting new data to achieve fast loading

Storage Model

storage of space and time:

1. integrated space and time approach: space and time have an equal role in the SFC code
2. non-integrated space and time approach: time dominates over space (and used first in organization)





second option easier to add data (dynamic scenario), no reorganization

Overview

- motivation
- scale as dimension
- Functionality
- data management
- standardization
- conclusion

OGC Domain Working Group PC

OGC Making location count.

DWG PC is active for about 1 year
chairs: Stan Tillman (Intergraph), Jan Boehm (UCL), myself

first, conducted Point Cloud Survey (use, tools, needs,…)
received 188 responses:
https://docs.google.com/spreadsheets/d/1_6389UlkIblWyneY5WbbONMMJ-ZiNaeUmcs_iG6olS0/edit?usp=sharing

next, following priorities for the DWG are identified:

1. further collaborate with ASPRS on LAS (OGC community standard)
2. explore HDF5 as format for Point Cloud data
3. interoperable steaming Point Cloud webservices

Standardization of point clouds?

- ISO/OGC spatial data:
  - at abstract/generic level, 2 types of spatial representations: features and coverages
  - at next level (ADT level), 2 types: vector and raster, but perhaps points clouds should be added
  - at implementation/ encoding level, many different formats (for all three data types)
- nD point cloud:
  - points in nD space and not per se limited to x,y,z (n ordinates of point which may also have m attributes)
  - make fit in future ISO 19107
  - note: nD point clouds are very generic; e.g. also cover moving object point data: x,y,z,t (id) series.

Characteristics of possible standard point cloud data type

1. xyz (a lot, use SRS, various base data types: int, float, double,..)
2. attributes per point (e.g. intensity I, color RGB or classification, or imp or observation point-target point or…)
   → correspond conceptually to a higher dimensional point
3. fast access (spatial cohesion) → blocking scheme (in 2D, 3D, …)
4. space efficient storage → compression (exploit spatial cohesion)
5. data pyramid (LoD, multi-scale/vario-scale, perspective) support
6. temporal aspect: time per point (costly) or block (less refined)
7. query accuracies (blocks, refines subsets blocks with/without tolerance value of on 2D, 3D or nD query ranges or geometries)
8. operators/functionality (next slides)
9. options to indicate use of parallel processing

Grouping of functionality

a. loading, specify conversion / organization
b. selections
c. LoD use/access
d. analysis I (not assuming 2D surface in 3D space)
e. analysis II (some assuming a 2D surface in 3D space)
f. conversions (some assuming 2D surface in 3D space)
g. towards reconstruction, classification, segmentation
h. updates: insert, delete, modify

(grouping of functionalities from user requirements)

Loading, specify conversion / organization

- input format
- storage blocks based on which dimensions (2, 3, 4,…)
- data pyramid, block dimensions (level: discrete or continuous)
- compression option (none, lossless, lossy)
- spatial clustering (morton, hilbert,…) within and between blocks
- spatial indexing (rtree, quadtree) within and between blocks
- validation (more format, e.g. no attributes omitted, than any geometry or topological validation; perhaps outlier detection)?

Webservices

- better not try to standardize point clouds at database level (not much support/ partners expected), but rather focus on webservices level (more support/ partners expected)
- there is overlap between WMS, WFS and WCS...
- OGC point cloud DWG should explore if WCS is good start for point cloud services:
  - If so, then analyse if it needs extension
  - If not good starting point, consider a specific WPCS, web point cloud service standards (and perhaps further increase the overlapping family of WMS, WFS, WCS,...)

Overview

- motivation
- scale as dimension
- functionality
- data management
- standardization
- conclusion

Related projects and PhD theses

- Massive Point Clouds (NL): NL eScience Center, Oracle, RWS, Fugro, CWI/MonetDB, TUD Harvest4D (EU): Uni Wien, TUD computer graphics
- IQmulus (EU): UCL, TUD, many more
- Ahn Vu Vo: Spatial Data Storage and processing Strategies for Urban Laser Scanning, PhD thesis, University College Dublin, October 2016.
- Remi Cura: Inverse Procedural Street Modelling from interactive to automatic reconstruction, PhD thesis, University Paris Est (IGN/Thales), September 2016.

Conclusion

- nD-PointClouds as 3rd representation: direct use (storage, analysis, visualization) or conversation to vector or raster
- develop functionality inside the database: encoding and decoding SFC, SFC ranges generation
- investigate different space-time-scale relative dimension representations in hypercube (for surface PC data, but also for more dynamic data: moving object trajectories)
- investigate other SFCs (Morton/Hilbert, less ranges) and/or other organizations (kd-tree, simplex based)
- generation of blocks using the same integrations of space, time and scale (more efficient: less rows, block compression, ...)

Implementation / code

- Python code Dynamic Point Cloud available at:
  https://github.com/stpsomad/DynamicPCDMS
- C++ code for Morton/Hilbert encode/decode/range generation
  https://github.com/kwan2004/SFCLib
- eScience Massive Point Cloud code (database/ viewer) & docu
  http://pointclouds.nl
- Oracle Database 12c
  (Enterprise Edition Release 12.1.0.1.2 – 64 bit)
  - Use of Index Organized Table (IOT)
  - NUMBER data type for 128 bit Morton/Hilbert keys

Thanks for your attention

- time for questions?

OGC actions in more detail ASPRS: LAS file format

- American Society for Photogrammetry and Remote Sensing (ASPRS) developed LAS 1.4; https://www.asprs.org/committee-general/laser-las-file-format-exchange-activities.html (with Domain Profile)
- 2 nov'15: OGC and ASPRS to *collaborate* on geospatial standards, invite participation in Point Cloud work; http://www.opengeospatial.org/pressroom/pressreleases/2313
- Ongoing effort to bring the LAS 1.4 point cloud format into the OGC as a community standard
- Attention points (for the future): Attribute flexibility, Other sources than laser, Compression, Organization (clustering)

HDF5 for Point Cloud data

- *Explore capabilities*: test/ benchmarks, assess tools
- Hierarchical Data Format (HDF): file format to store /organize large amounts of data, originally by National Center for Supercomputing Applications (NCSA)
- Hierarchical, filesystem-like data format, 2 types of objects:
  - Datasets: nD arrays
  - Groups: container structures for datasets and other groups
- See HDF5 for point cloud data
  - Chauhan et al (jun' 15): National Geospatial Intelligence Agency (NGA) Sensor Independent Point Cloud (SIPC)
  - Ingram (mar'16): Advanced Point Cloud Format Standards
- Note: also NetCDF 4 (more grid oriented) is based on HDF5

Steaming Point Cloud webservices

- Web services protocol (request/selection, response)
- Data format
- Streaming, ordering, compression
- Caching
- Progressive refinement
- Support LoD's
- Visualization

- Fitting in existing WXS (WCS, WFS) or new service needed (WPCS)?
- Earlier work of OS Geo pointdown
  - https://lists.osgeo.org/mailman/listinfo/pointdown
  - https://github.com/pointdown/protocol

Bijlage B

Point Cloud Survey – OGC Point Cloud DWG Presentatie
Stan Tillman (Intergraph) namens zijn medevoorzitters Jan Boehm (University College London) en Peter van Oosterom (TU Delft),
21 juni 2016, Dublin, Ierland

Sponsored by

COBWEB
Citizen Observatory Web

Funded by the European Union

Hosted by

UCD DUBLIN

Presentation Template

99th OGC Technical Committee
Dublin, Ireland
Stan Tillman, Jan Boehm, Peter van Oosterom
21 June 2016

OGC® Making location count.

Agenda

- How we got here
- Review of the Point Cloud Survey results
- What's next?

OGC®

How we got here...

- Discussions regarding where Point Clouds fit within OGC have been occurring for some time
- At the Boulder meetings, we held an ad hoc and quickly discovered that the interest level was very high
- We created an charter and formalized the group during the Nottingham meetings
- We elected the following chairs:
  - Jan Boehm, University College London
  - Peter van Oosterom, TU Delft
  - Stan Tillman, Intergraph

OGC®

How we got here...

- Quickly we began to have concerns and emails questioning whether the work done would be fair to all.
- So the chairs met and decided before we begin with a bunch of controversial meetings, we would conduct a survey to determine the topics and interoperability issues most pressing in the community.

OGC®

Review of the Point Cloud Survey results

- 13 Questions ranging from storage, transfer, size, tools, etc.
- 188 responses from 175 different organizations
- The following slide results:
  - Are not filtering the duplicate responses
  - Groups a lot of individual responses
  - Does not attempt to consolidate similar answers.
  - But we do not feel any of this skews the overall results of the survey

OGC®

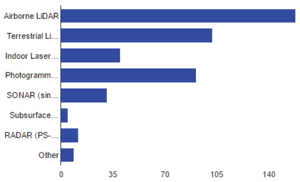
Links to the Survey Results

- Spreadsheet: https://docs.google.com/spreadsheets/d/1_6389UlkIblWyneY5WbbONMMJ-ZiNaeUmcs_iG6olS0/edit?usp=sharing
- Analytics: https://docs.google.com/forms/d/19eQVHYaylKwBHOQTlVrlitxk8AXm1xbzmfCWyeHKICs/viewanalytics

OGC®

Question #1

1. What are your major sources for surface scan point clouds?

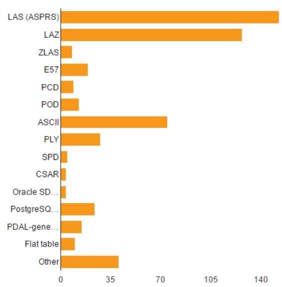


Airborne LiDAR	**158**	82.3%
Terrestrial Lidar (including Mobile Mapping)	**102**	53.1%
Indoor Laser Scanning	**40**	20.8%
Photogrammetry	**91**	47.4%
SONAR (single and multi-beam echo's)	**31**	16.1%
Subsurface Point Cloud from Seismic	**5**	2.6%
RADAR (PS-InSAR)	**12**	6.3%
Other	**9**	4.7%

OGC®

Question #2

2. What formats do you use to store point clouds?



LAS (ASPRS)	**153**	79.7%
LAZ	**127**	66.1%
ZLAS	**8**	4.2%
E57	**19**	9.9%
PCD	**9**	4.7%
POD	**13**	6.8%
ASCII	**75**	39.1%
PLY	**28**	14.6%
SPD	**5**	2.6%
CSAR	**4**	2.1%
Oracle SDO_PC	**4**	2.1%
PostgreSQL-PostGIS PCPATCH	**24**	12.5%
PDAL-generated blocks	**15**	7.8%
Flat table	**10**	5.2%
Other	**41**	21.4%

OGC®

Question #2 - Other

LAS (ASPRS)	149
LAZ	125
ASCII	73
PLY	27
PostgreSQL-PostGIS PCPATCH	23
E57	17
PDAL-generated blocks	14
POD	12
Flat table	9
PCD	7
ZLAS	7
SPD	4
bpf	4
CSAR	3
Oracle SDO_PC	3
Pts	3
RCS	2
netCDF	2
ptx	2
.bpf	1
.fls	1
.zfs	1
3D shapefile	1
Any of various vendor formats	1
BAG Bathymetric Attributed Grid	1
Don't use. Just interested in what others do.	1

ESRI LASD	1
FLS	1
GMP	1
GSF	1
H5	1
Hexagon Point Cloud (HPC) format	1
IMP	1
Indexed database with file pointers to las/laz	1
Kongsberg ALL	1
La20 (Trimble format)	1
MATLAB .mat	1
MB-System supported sonar formats	1
OPC : Orbit Point Cloud	1
Own format	1
PCG	1
PTX and various vendor specific formats	1
PulseWaves	1
RDB	1
RXP	1
SAGA .spc	1
SQLite	1
Terrasolid Fast Binary	1
WMO GRIB & BUFR	1
after bringing in then it is stored as northing	1
and lat	1
custom format	1
database	1
easting	1
fls fws pcg rcp	1
long	1
shapefile	1
x3d	1

OGC®

Question #3

3. What formats do you use to transfer point clouds (both internally and to external entities)?

LAS (ASPRS)	136	70.8%
LAZ	122	63.5%
ZLAS	9	4.7%
E57	21	10.9%
PCD	5	2.6%
POD	9	4.7%
ASCII	67	34.9%
PLY	16	8.3%
SPD	4	2.1%
CSAR	3	1.6%
Oracle SDO_PC	2	1%
PostgreSQL-PostGIS PCPATCH	11	5.7%
PDAL-generated blocks	6	3.1%
Flat table	7	3.6%
Other	30	15.6%

OGC®

Question #3 - Other

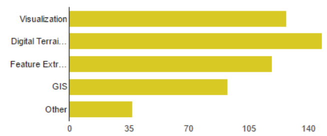
LAS (ASPRS)	133
LAZ	120
ASCII	65
E57	19
PLY	14
PostgreSQL-PostGIS PCPATCH	10
ZLAS	8
POD	7
Flat table	5
PDAL-generated blocks	5
BPF	3
PCD	3
Rcs	3
SPD	3
CSAR	2
.PTX	1
.bpf	1
.fls	1
.zfs	1
Bag	1
Don't use. Just interested in what others do.	1
FLS	1

GMP	1
GSF	1
H5	1
Hexagon Point Cloud (HPC) format	1
LODTree	1
MB-System	1
MB-System supported sonar formats	1
Oracle SDO_PC	1
Own format	1
PCG	1
PulseWaves	1
SAGA .spc	1
SQLite	1
WMO GRIB & BUFR	1
custom format	1
netCDF	1
own progressive streaming approach	1
pts	1
ptx	1
shapefile	1
x3d	1

OGC®

Question #4

4. What are your most common use cases for point clouds?



Visualization	127	66.1%
Digital Terrain Modelling	148	77.1%
Feature Extraction	119	62%
GIS	93	48.4%
Other	37	19.3%

OGC®

Question #4 - Other

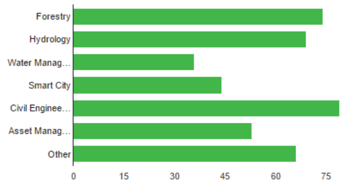
Digital Terrain Modelling	147
Visualization	124
Feature Extraction	116
GIS	92
Volumetric Analysis	2
3D BUILDING MODELS	1
3D Modeling	1
3D Modelling and Analysis	1
BIM	1
Building Modelling	1
CAD/CAM	1
City modelling	1
Clash detection	1
Construction Quality Control	1
Coordination	1
Deformation analysis	1
Distribution to users.	1
Don't use. Just interested in what others do.	1
Fabrication measurements	1
Feature Detection	1
Forest biomass assessments	1
Fully 3D object modeling	1
Intermediate product for 3D textured mesh reconstruction	1

Measurement	1
Metrics extrapolation	1
Modelling of forest variables	1
Numerical Weather Prediction	1
Profiling	1
Remote sensing research	1
Slope analysis	1
Vegetation Metrics	1
Vegetation modelling	1
Vol Calculations	1
Volumetrics	1
as a start point for estimating characteristics of sea ice	1
calculation of descriptive statistics of return heights for modeling vegetation characteristics	1
fls fws pcg rcp	1
forest canopy modelling	1
forestry	1
geological interpretation	1
geological/sedimentary description/quantification	1
land forming possibilities	1
object detection	1
software development	1
surface reconstruction	1
tree extraction	1
viewshed	1
watershed	1

OGC®

Question #5

5. What are your main application areas for point clouds?



Forestry	74	38.5%
Hydrology	69	35.9%
Water Management	36	18.8%
Smart City	44	22.9%
Civil Engineering	79	41.1%
Asset Management	53	27.6%
Other	66	34.4%

OGC®

Copyright © 2016 Open Geospatial Consortium

Question #5 - Other

Civil Engineering	77
Forestry	72
Hydrology	67
Asset Management	52
Smart City	42
Water Management	35
Archaeology	5
Hydrography	3
Bathymetry	2
Defense	2
Geology	2
Mining	2
Utilities	2
geomorphology	2
terrain analysis	2
Agribusiness	1
Agriculture	1
Analysis	1
BIM	1
Clash Detection	1
Coastal zone management	1
Construction	1
Contours	1
Cultural Heritage	1
DAYLIGHT PLANNING	1
Defense/Intel	1
Design	1
Distribution to users.	1
Don't use. Just interested in what others do.	1

Environment	1
Environmental management	1
Environmental monitoring	1
GIS management	1
Geodesy	1
Geology and general Terrain Analysis	1
Geophysics	1
Heritage Documentation	1
Industrial	1
Intelligence Community (IC) operations	1
Land Surveying Topo	1
Marine geology and geophysics	1
Marine seabed	1
Meteorology	1
Metrology	1
NMA	1
Object detection and recognition	1
Oceanography	1
Oil and Gas	1
Orienteering mapping	1
Planning	1
Production of DTM	1
Profiling	1
Public Safety	1
R&D	1
Rail	1
Road Admin	1
Seafloor Mapping	1
Site Characterization	1
Survey replacement	1

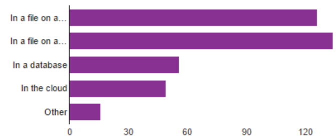
Surveying	1
Tunneling	1
VFX	1
View analysis	1
Volumetric Analysis	1
anything	1
archeology	1
as-built documentation	1
buildings	1
cryosphere and climate change	1
ecology	1
education/teaching	1
mapping	1
marine cryosphere characterisation	1
natural hazards	1
nature cons.	1
plants	1
province level	1
research	1
seafloor habitat mapping	1
sedimentology	1
surveying and mapping	1
trees	1
underwater cultural heritage	1
variety (education)	1

OGC®

Copyright © 2016 Open Geospatial Consortium

Question #6

6. How do you store point clouds?



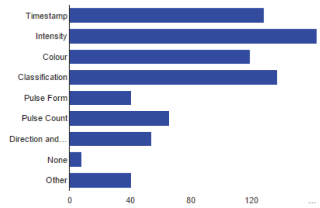
In a file on a computer	126	65.6%
In a file on a network drive	134	69.8%
In a database	56	29.2%
In the cloud	49	25.5%
Other	16	8.3%

OGC®

Copyright © 2016 Open Geospatial Consortium

Question #7

7. What attributes do your point clouds contain besides XYZ coordinates?



Timestamp	128	66.7%
Intensity	163	84.9%
Colour	119	62%
Classification	137	71.4%
Pulse Form	41	21.4%
Pulse Count	66	34.4%
Direction and Length of Scanline	54	28.1%
None	8	4.2%
Other	41	21.4%

OGC®

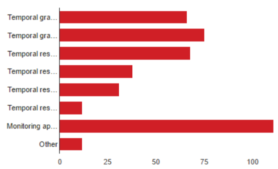
Question #7 - Other

Intensity	160
Classification	136
Timestamp	126
Colour	118
Pulse Count	65
Direction and Length of Scanline	53
Pulse Form	40
None	8
normals	4
scan angle	2
3d)	1
Accuracy (stddev)	1
Derived structural features	1
Don't use. Just interested in what others do.	1
Feature extraction	1
Flightline ID	1
Full LAS1.3 gamut	1
GPM error model data	1
Images	1
Location of sensor	1
Many attributes	1
Measurement angle	1
NIR	1
Normal Vector	1
Reflectivity	1
Return attributes from waveform processing	1
Return number	1
Scanner #	1
Statistical Uncertainty	1
TPU	1
Time Change	1
Various Tags	1
Volume Calculations	1
Waveform derived return metrics	1
We keep everything we receive	1
X-Sections	1
acoustic backscatter	1
acoustic variables such as pulse length	1
all las attributes and occasionally some custom attributes.	1
amplitude	1
background radiation	1
beam direction	1
depends on the use case and the data we get	1
derived features	1
direction	1
edge of flightline	1
estimated surface normals	1
etc. VVV imp for sonar processing that all these things are kept together	1
full suite of ASPRS LAS attributes	1
full-waveform analysis	1
mirror facet	1
number of returns	1
other user data	1
overlap flag	1
phi	1
position uncertainty (x	1
profiling	1
pulse index	1
pulse shape deviation	1
pulse source ID (flightline ID)	1
range	1
reflectance	1
return count	1
scan position	1
sensor position !!!	1
sometimes trajectory information	1
sonar attributes	1
sourceID	1
surface normal	1
temperature	1
theta	1
user data	1
varies according to source	1
various additional derived data	1
various arbitrary attributes	1
vehicle location/attitude	1
visible	1
withheld flag	1
y	1
z	1

OGC®

Question #8

9. Which temporal aspect of point clouds are relevant for you?

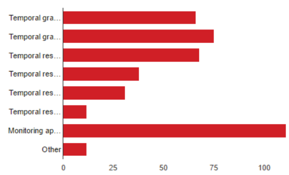


Temporal granularity at point level	66	34.6%
Temporal granularity at data set (a 'point cloud') level	75	39.3%
Temporal resolution / update frequency years	68	35.6%
Temporal resolution / update frequency months	38	19.9%
Temporal resolution / update frequency days	31	16.2%
Temporal resolution / update frequency seconds	12	6.3%
Monitoring applications, change detection	111	58.1%
Other	12	6.3%

OGC®

Question #9

9. Which temporal aspect of point clouds are relevant for you?

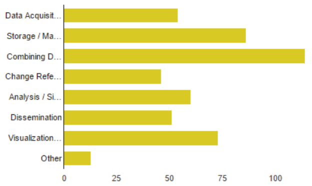


Temporal granularity at point level	66	34.6%
Temporal granularity at data set (a 'point cloud') level	75	39.3%
Temporal resolution / update frequency years	68	35.6%
Temporal resolution / update frequency months	38	19.9%
Temporal resolution / update frequency days	31	16.2%
Temporal resolution / update frequency seconds	12	6.3%
Monitoring applications, change detection	111	58.1%
Other	12	6.3%

OGC®

Question #10

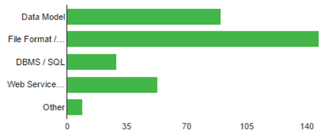
10. During what phase do you encounter interoperability challenges?



Data Acquisition	54	28.4%
Storage / Management	86	45.3%
Combining Data from multiple source	114	60%
Change Reference System	46	24.2%
Analysis / Simluation	60	31.6%
Dissemination	51	26.8%
Visualization / Interaction	73	38.4%
Other	13	6.8%

Question #11

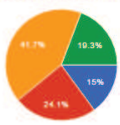
11. What do you consider the most important area of point cloud standardization?



Data Model	90	46.9%
File Format / Encoding	147	76.6%
DBMS / SQL	29	15.1%
Web Service (WxxS) protocol	53	27.6%
Other	9	4.7%

Question #12

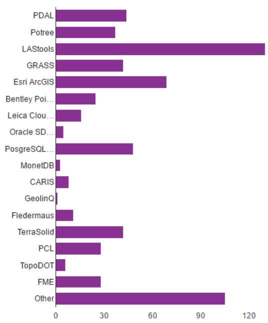
12. What volume of point clouds have you managed/processed/stored/etc. in the last 12 month?



Less than 100 million (10^6) points	28	14.6%
More than 100 million (10^6) points	45	23.4%
More than 1 billion (10^9) points	78	40.6%
More than 1 trillion (10^12) points	36	18.8%

Question #13

13. What tools do you use?



PDAL	44	22.9%
Potree	37	19.3%
LAStools	130	67.7%
GRASS	42	21.9%
Esri ArcGIS	69	35.9%
Bentley Pointools	25	13%
Leica CloudWorx	16	8.3%
Oracle SDO_PC	5	2.6%
PosgreSQL/PostGIS	48	25%
MonetDB	3	1.6%
CARIS	8	4.2%
GeolinQ	1	0.5%
Fledermaus	11	5.7%
TerraSolid	42	21.9%
PCL	28	14.6%
TopoDOT	6	3.1%
FME	28	14.6%
Other	105	54.7%

Question #13 - Other

LAStools	128
Esri ArcGIS	67
PosgreSQL/PostGIS	47
PDAL	42
GRASS	41
Potree	35
FME	27
PCL	26
Bentley Pointools	23
Leica CloudWorx	15
Fledermaus	10
CARIS	7
Global Mapper	7
LP360	6
FUSION	5
Oracle SDO_PC	4
FARO Scene	3
Orbit	3
Python	3
Cloud Compare	2
ENVI	2
Envi LiDAR	2
FugroViewer	2
GMT	2
GSTDHMQC	2
HYPACK	2
Inpho	2
Leica Cyclone	2
MARS	2

MATLAB	2
MB-System	2
Meshlab	2
MonetDB	2
AGIS	1
AMW	1
AutoDesk ReCap	1
Autodesk Civil 3D	1
Autodesk products like ReCap and Infraworks	1
BayesStripAlign	1
Blue Marble and MARS7	1
C++	1
CloudPro	1
Custom SW	1
Custom written LASF access	1
Cyclone	1
DTMaster	1
Don't use. Just interested in what others do.	1
EIVA NaviModel	1
ENVI/IDL	1
ERDAS IMAGINE	1
Fortran!	1
Fusion LDV	1
GDAL	1
GeoMedia	1
GeoVisionary	1
Geocue	1
GlobalMapper	1
GoTools	1
Hydrographic software (e.g. QINSy	1

Hypack. Plus I developed my own tools: http://dbuscombe-usgs.github.io/pysesa/	1
ISATIS	1
ImageStation	1
InstantReality	1
Johns Hopkins APL tools for US Govt	1
LIDARServer	1
Laserdata LIS	1
Leica LIDAR Survey Studio LSS	1
Leica cyclone and map factory	1
LidarViewer	1
MCC lidar	1
MICRODEM	1
MapInfo Advanced	1
Maptek's I-Site Studio	1
Matlab etc.	1
Numpy	1
OPALS	1
Optech LMS 3.0	1
Opti	1
OrbitMM	1
Own software	1
PDS)	1
Pix4D	1
Plotly	1
Point tools	1
PolyWorks	1

OGC®

What's next?

Based on the Point Cloud Survey the following priorities for the DWG are identified:

1. Further collaborate with ASPRS on LAS
2. Explore HDF5 as format for Point Cloud data
3. Interoperable steaming Point Cloud webservices

OGC®

ASPRS: LAS file format

- American Society for Photogrammetry and Remote Sensing (ASPRS) developed LAS 1.4; https://www.asprs.org/committee-general/laser-las-file-format-exchange-activities.html (with Domain Profile)
- 2 nov'15: OGC and ASPRS to *collaborate* on geospatial standards, invite participation in Point Cloud work; http://www.opengeospatial.org/pressroom/pressreleases/2313
- Attention points:
  - Attribute flexibility
  - Other sources than laser
  - Compression
  - Organization (clustering)

OGC®

HDF5 for Point Cloud data

- *Explore capabilities*: test/ benchmarks, assess tools
- Hierarchical Data Format (HDF): file format to store /organize large amounts of data, originally by National Center for Supercomputing Applications (NCSA)
- Hierarchical, filesystem-like data format, 2 types of objects:
  - Datasets: nD arrays
  - Groups: container structures for datasets and other groups
- See HDF5 for point cloud data
  - Chauhan et al (jun' 15): National Geospatial Intelligence Agency (NGA) Sensor Independent Point Cloud (SIPC)
  - Ingram (mar'16): Advanced Point Cloud Format Standards
- Note: also NetCDF 4 (more grid oriented) is based on HDF5

OGC®

Steaming Point Cloud webservices

- Web services protocol (request/selection, response)
- Data format
- Streaming, ordering, compression
- Caching
- Progressive refinement
- Support LoD's
- Visualization

- Fitting in existing WXS (WCS, WFS) or new service needed (WPCS)?
- Earlier work of OS Geo pointdown
  - https://lists.osgeo.org/mailman/listinfo/pointdown
  - https://github.com/pointdown/protocol

OGC®

Bijlage C

‘Towards vario-scale and dynamic point clouds’
Peter van Oosterom, Stella Psomadaki, Theo Tijssen/TU Delft &
Fedor Baart/Deltares
21 juni 2016, Dublin, Ierland

Sponsored by

Hosted by

COBWEB
Citizen Observatory Web

Funded by the European Union

UCD DUBLIN

Towards vario-scale and dynamic point clouds

99th OGC Technical Committee, Dublin, Ireland

Peter van Oosterom[a], Stella Psomadaki[a], Theo Tijssen[a], Fedor Baart[b]

[a] TU Delft, [b] Deltares

21 June 2016

OGC® Making location count.

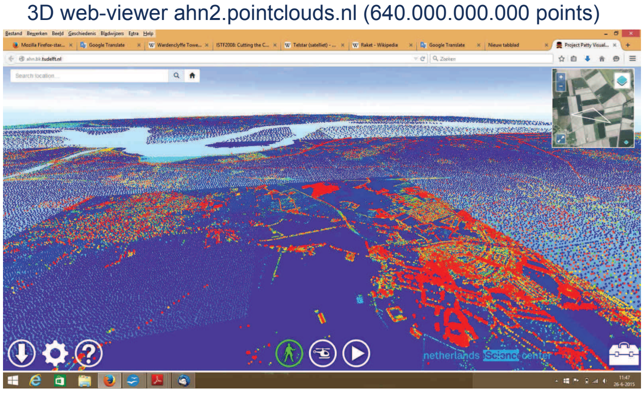
Motivation

- Point cloud data sets are often used for monitoring
  → dynamic point clouds
  → time as additional organizing dimension
- Organizing point cloud data in LoD's/importance levels is an approach to manage large data sets
  → LoD: discrete (multi-scale) or continuous (vario-scale)
  → scale treated as additional organizing dimension
- How to manage higher dimensional point clouds (4D, 5D)
  → Explore with dynamic point clouds

OGC®

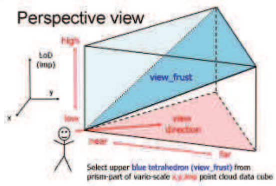
Discrete LoD's are visible…

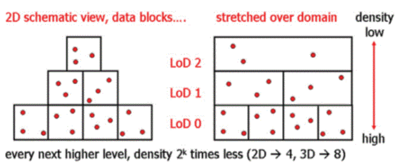
3D web-viewer ahn2.pointclouds.nl (640.000.000.000 points)

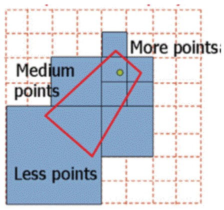


Time and scale as dimensions

- Time more obvious: space-time data cubes
- Scale as dimension: discrete or continuous







Contents

1. Dynamic Point Clouds

- Point clouds are generated every day, hour, minute
- Repeated scans of the same area → dynamic
- Time as selective as the spatial component or needed in integrated space – time selections
- Current DBMS solutions designed for static point clouds
- Management is still a challenge

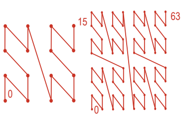
2. Space Filling Curves (SFCs)

- Apply linear ordering to a multidimensional domain (spatial clustering)
- Organise a flat table efficiently
- Full resolution curve: avoid storing x,y[,z] + t → recovered from SFC key → less storage

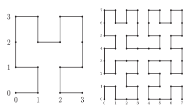
- Use Index Organized Table (data stored in the B-Tree index)
- Queries need to be re-written to SFC-ranges, benefit from spatial clustering → efficient

- SFCs based on hyper-cubes
  - Morton/Hilbert both **nD and quadrant recursive**
  - Consider relative scaling of space and time
  - Space reserved on the hypercube for future data

Morton (Peano)

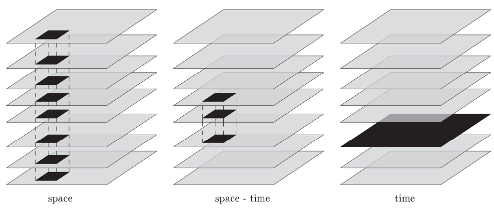


Hilbert



Important Queries

- **Space queries** all points located in a specific area
- **Space-time queries** all points located in a specific area during a specific time range
- **Time queries** all points at specific moment in time (range)



3. Storage Model

Optimizing for space/time queries in case of dynamic data

Contradiction:

1. Points close in space and time should be stored (to some extent) close in memory for fast spatio-temporal retrieval
2. Already organised points should not be reorganised when inserting new data to achieve fast loading

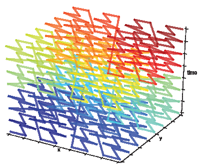
Optimisation/ tuning is needed

OGC®

Storage Model

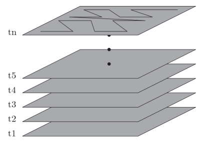
Storage of space and time:

1. Integrated space and time approach: space and time have an equal role in the Morton code
2. Non-integrated space and time approach: time dominates over space and is stored as a separate column (and used first in organisation)



Two treatments of z can be identified:

- z stored as an attribute
- z part of the Morton code (as dimension)



OGC®

Storage model – Table definitions

- Time non-integrated (z as attribute) → 2D Morton

```
CREATE TABLE xy (time, morton, z,
CONSTRAINT xy_PK PRIMARY KEY (time, morton)) ORGANIZATION INDEX
```

- Time non-integrated (z in Morton) → 3D Morton

```
CREATE TABLE xyz (time, morton,
CONSTRAINT xyz_PK PRIMARY KEY (time, morton)) ORGANIZATION INDEX
```

- Time integrated (z as attribute) → 3D Morton

```
CREATE TABLE xyt (morton, z,
CONSTRAINT xyt_PK PRIMARY KEY (morton)) ORGANIZATION INDEX
```

- Time integrated (z in Morton) → 4D Morton

```
CREATE TABLE xyzt (morton,
CONSTRAINT xyzt_PK PRIMARY KEY (morton)) ORGANIZATION INDEX
```

OGC®

4. Implementation

- Python code available at: https://github.com/stpsomad/DynamicPCDMS
- Oracle Database 12c (Enterprise Edition Release 12.1.0.1.2 – 64 bit)
  - Use of Index Organized Table (IOT)
  - NUMBER data type for 128 bit Morton keys

OGC®

Loading Procedure

Divided into following phases:

1. **Preparation**: Depending on the integration of space and time and treatment of z: LAZ files converted to Morton key. Data bulk loaded into a normal heap table.
2. **Loading**: Sort the heap table based on the key and store in an IOT
3. **Post-processing** (optional): gather optimiser statistics

Note:

- Incremental loading used to simulate growing data
- Adding new data by combining the **old IOT** and the new heap table into one **new IOT** table

OGC®

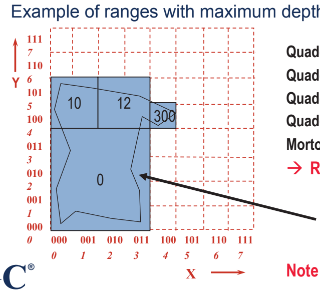
Query procedure

- Use the spatial clustering by utilising the relationship between the Quad/Oc/…-tree and the SLC (next slide)
- The maximum depth of the tree affects:
  - The number of Morton ranges that compose the query
  - The approximation of the query geometry
- Multi-step query procedure:
  - Filtering based on SFC ranges
  - Decoding to original dimensions (x,y,z,t) and storing (no functions available inside the database, now performing this step with Python)
  - Refinement (Point in Polygon, time and z refinement)

OGC®

Query geometry (domain) → SFC ranges

- based on concepts of Region Quadtree & Quadcodes
- works for any type of query geometry (point, polyline, polygon)
- also works in 3D (Octree) and higher dimensions
- Example of ranges with maximum depth of 3 level:



Quadcode 0:	Morton range 0-15
Quadcode 10:	Morton range 16-19
Quadcode 12:	Morton range 24-27
Quadcode 300:	Morton range 48-48

Morton code gaps resp. 0, 4, 20

→ Range merging possible

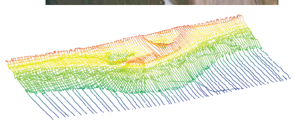
query_geometry, polygon

Note : SW=0, NW=1, SE=2, NE=3

OGC®

5. Benchmark design

- Sand Engine (Netherlands)
  - 4.5 x 4.5 km2
  - 100,000 points per file
  - **mm spatial** resolution
  - datasets **per day**
  - mix of actual measurements and simulated data

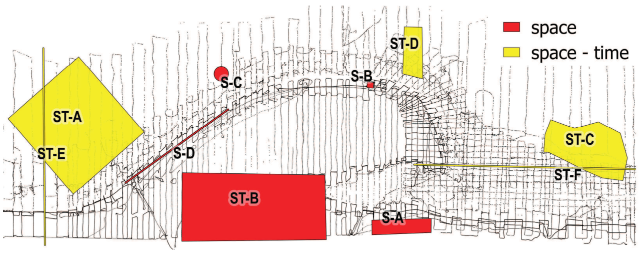


Benchmark	Points	Days	Size (MB)	Description
Small	20M	230	346	From 2000-02
Medium	40M	554	835	From 2003-06
Large	80M	931	1410	From 2007-15

OGC®

Benchmark Design

- Queries
  - Space and time ranges (ST)
  - Only space (S)
  - Only time (T)
- Different query geometries
  - rectangles
  - polygons
  - lines + buffer
  - point + buffer



6. Results – incremental loading

	Approach	Preparation (s) Morton conversion	Preparation (s) heap loading	loading IOT (s)	Total (s)	Size (MB)	points
non-integrated + z as attribute	xy-small	102.34	12.20	15.38	129.92	471	18,147,709
	xy-medium	144.76	17.18	45.27	207.21	1130	43,708,815
	xy-large	170.31	20.25	82.62	273.18	1897	73,913,926
non-integrated + z in Morton	xyz-small	324.48	10.40	12.13	347.01	368	18,147,709
	xyz-medium	450.23	14.60	34.66	499.49	885	43,708,815
	xyz-large	536.27	17.22	66.18	619.67	1495	73,913,926
integrated + z as attribute	xyt-small	331.80	10.97	33.51	376.28	471	18,147,709
	xyt-medium	487.14	14.97	47.92	550.03	1130	43,708,815
	xyt-large	569.47	17.17	82.34	668.98	1897	73,913,926
integrated + z in Morton	xyzt-small	406.31	9.55	10.96	426.82	386	18,147,709
	xyzt-medium	577.02	12.20	39.49	628.71	927	43,708,815
	xyzt-large	685.38	14.77	62.33	762.48	1566	73,913,926

Note: Scaling of 10,000: 1 mm = 10,000 days (t first in Morton)
Not so deep space-time integration: 1 m equals 10,000,000 days

Note: medium/large times for additional points, conversion+heap

Results – query via ranges

- Only **first fetching** presented: directly related to the depth of the tree and the maximum number of ranges specified (degree of merging)
- Two methods for posing the query:
  - Integrated approach: load the ranges in an IOT and perform **join**:

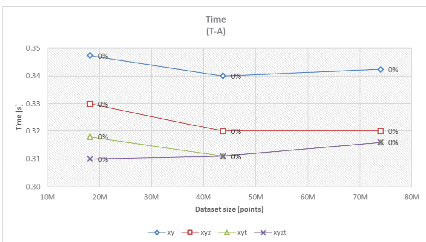
```
SELECT t.morton , t.z
  FROM table_IOT t, ranges r
  WHERE t.morton BETWEEN r.low AND r.upper;
```

  - Non-integrated approach: previous solution not efficient (**open issue**); the keys are specified in the WHERE clause – threshold of 200 ranges

```
SELECT morton, z FROM table_IOT
  WHERE ((morton BETWEEN 181664219136000 AND 181664231718911)
  OR ... (morton BETWEEN 181675304681472 AND 181675313070079));
```


Results – query first fetching (query T-A)

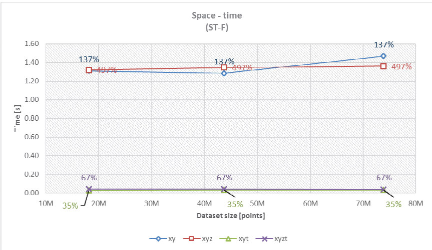
- **Time queries** have similar query response times between the four approaches
- No extra points between filtering step and refinement



All data between 25-10-02 and 26-10-02 → 1 day found

Results - first fetching (query ST-F)

- **Space – time queries** perform better for the integrated approach
- Between the two treatments of z, including it in the Morton increases the % of extra points



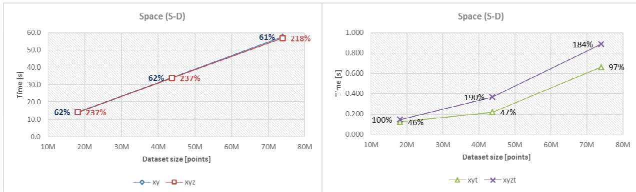
Line buffer and time between 01-01-02 and 15-01-02

OGC®

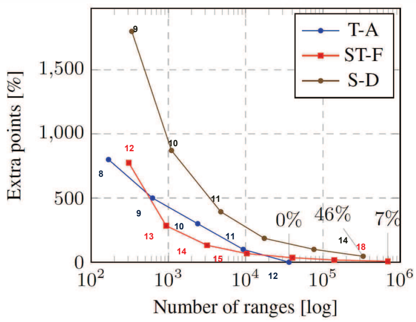
Results - first fetching (query S-D)

- **Space queries** perform better in the integrated approach, BUT seams no linear growth
- More points when z in the Morton key (different ranges)

Note: in general the space query response times increase due to that more points need to be selected between the three benchmark sizes (every day more data).



Results – depth of tree and % of extra points



The effect of increasing the depth of the tree on the percentage (%) of extra points obtained during the filtering phase of the integrated approach when z is an attribute (small dataset)

OGC®

Results – magnitude of merging ranges and % of extra points

query	maximum ranges	Depth	actual ranges	fetching [s]	extra [%]
Space - time (ST-D)	10	20	10306	0.077	4636%
	100	20	10306	0.099	384%
	1000	20	10306	20.298	18%
	10000	20	10306	69.381	9%
Space (S-B)	10	22	7458	3.653	4943%
	100	22	7458	34.367	458%
	1000	22	7458	390.246	54%
	10000	22	7458	1131.950	28%

Using specified maximum ranges in the **WHERE-clause** statement of the non-integrated approach (z attribute, small data set).

Note: for larger number of ranges the fetching is getting too slow, **open issue...**

OGC®

7. Conclusion

- Designed and executed benchmark dynamic point clouds
- Two integrations of space and time and two treatments of z
- Integrated approach provides better query response times compared to the non-integrated for the specific use case (Both treatments of z are possible, higher dimensions work)
- Key aspect of our implementation: Index Organized Table of Oracle
- Work in progress, more at 3D GeoInfo, Athens

OGC®

8. Future Work

- Use larger dataset; e.g. Coastline of Netherlands (2 trillion points), change/delta queries
- Parallel generation of the SFC and parallel loading and querying (Python → C++ code)
- Develop functionality inside the database: encoding and decoding SFC, SFC ranges generation
- Investigate different time-scalings (for surface PC data, but also for more dynamic data: moving object trajectories)
- Investigate other SFCs; Hilbert curve better, less ranges?
- Generation of blocks using the same integrations of space and time (more efficient: less rows, block compression, …)

OGC®

→ Vario-scale dynamic point clouds

Bijlage D

Afstudeerpresentatie Styliani (Stella) Psomadaki, 8 november 2016, Delft

Using a Space Filling Curve for the Management of Dynamic Point Cloud Data in a Relational DBMS

Stella Psomadaki
P5 Presentation

Theo Tijssen
Peter van Oosterom
Roderik Lindenbergh
Ulf Hackauf
External: Fedor Baart

TU Delft

Deltares

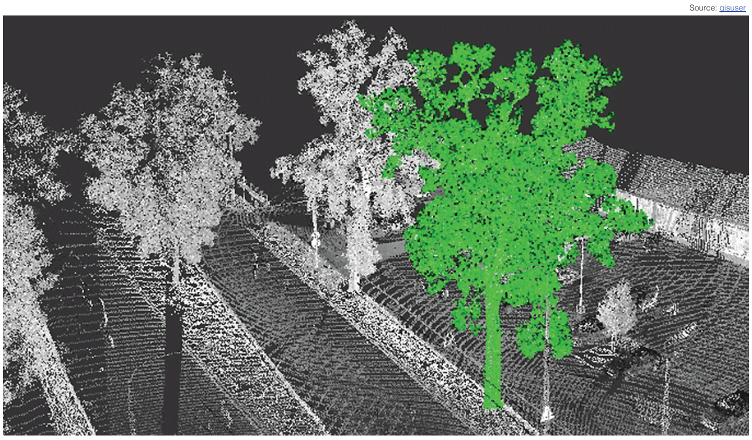
Contents

- Introduction
- Methodology
- Results
- Conclusions & Future work

Introduction

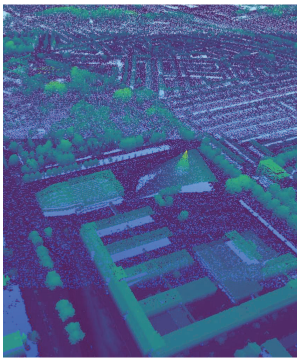
What is a Point cloud?

Source: gisuser



Point clouds

- Rapid growth in point cloud usage
- The management of point clouds is challenging
- Typically managed using files (e.g. LAS, LAZ)
- …But, DBMSs provide point cloud management solutions.



ahn2.pointclouds.nl/

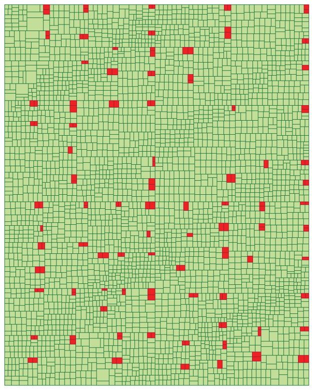
Management of PC in DBMS

Current approaches:

- Oracle *SDO_PC*
- PostgreSQL *pgpointcloud*

Organise points in **blocks**, meaning groups of spatially close points

...or use a normal **flat table**

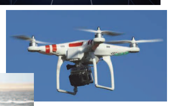


Source: Massive point clouds for eSciences
http://www.gdmc.nl:8080/mpc/

Dynamic point clouds

- Today, developments in point cloud acquisition devices allow repeated scans of the same area
- Dynamic point clouds
  - growing datasets
  - *time is an additional dimension*

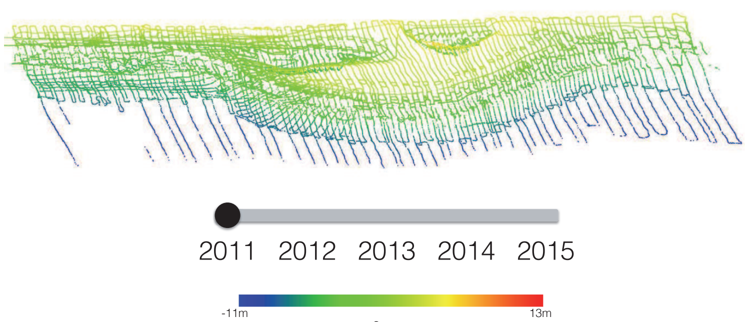
Source: youtube.com



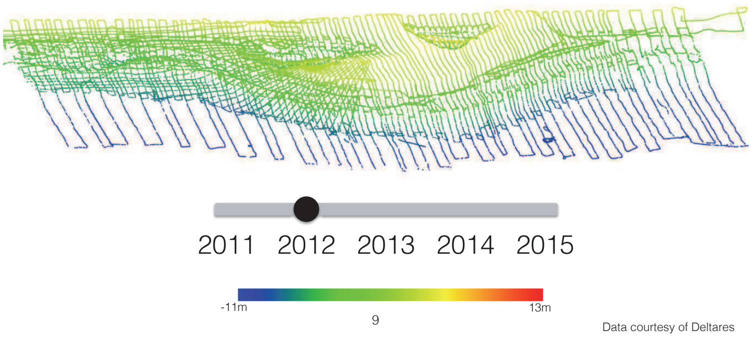
Source: Flickr

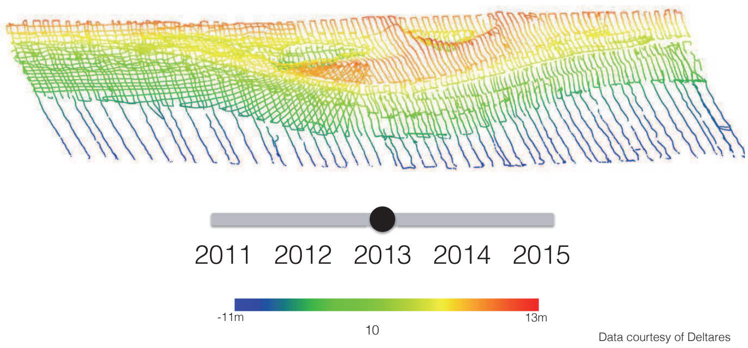
Source: TU Delft

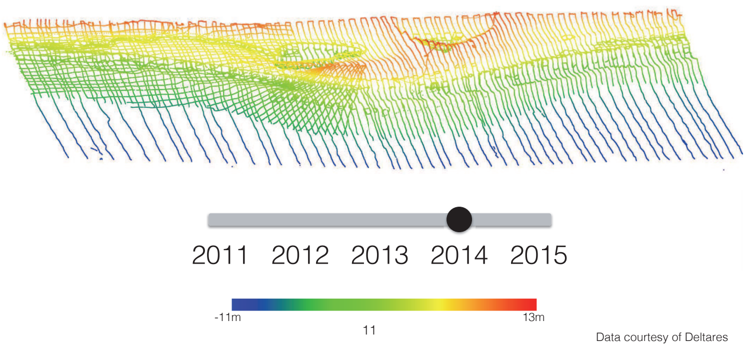
Source: Wikimedia Commons

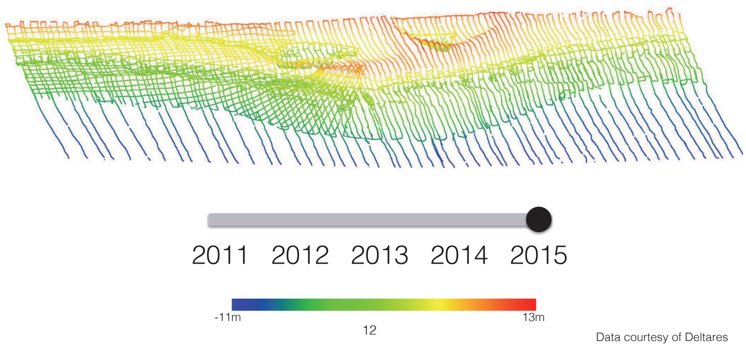


Data courtesy of Deltares


2011 2012 2013 2014 2015
-11m
13m
9
Data courtesy of Deltares


2011 2012 2013 2014 2015
-11m
13m
10
Data courtesy of Deltares


2011 2012 2013 2014 2015
-11m
13m
11
Data courtesy of Deltares

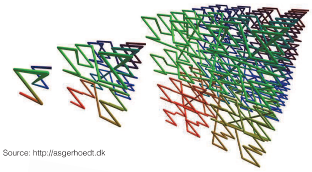

2011 2012 2013 2014 2015
-11m
13m
12
Data courtesy of Deltares

Managing dynamic PC?

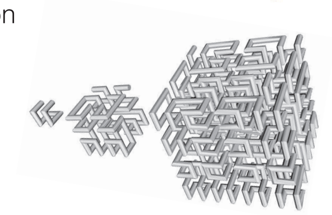
- Blocks
  - ☑ compact storage with better scalability, less overhead, better compression
  - ☐ overlapping blocks, adding new data not trivial
- Flat
  - ☑ flexible, insertions trivial, Use a SFC to improve the organisation (van Oosterom et. al., 2015)
  - ☐ large storage requirements, overhead

Space Filling Curves

- Apply a linear ordering to a multidimensional domain
- Why?
  - Dimensionality reduction
  - Full resolution curve
  - Clustering of points

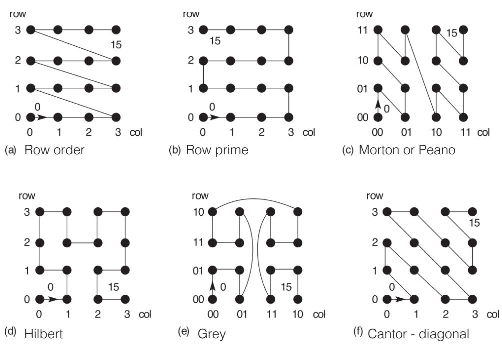


Source: http://asgerhoedt.dk



Sánchez-Cruz H, Bribiesca E;
Study of compression efficiency for three-dimensional discrete curves.
Opt. Eng. 0001;47(7):077206-077206-9.

Space Filling Curves



Source: van Oosterom, Peter. "Spatial access methods." *Geographical information systems* 1 (1999): 385-400.

Space Filling Curves

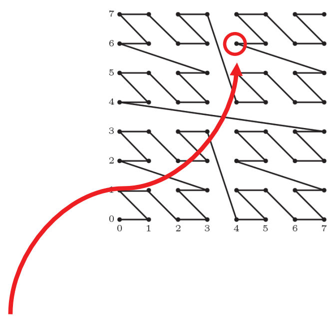
- Morton Curve
- Bitwise interleaving

Example:

x = 4 or 0100 in binary

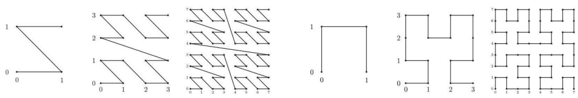
y = 6 or 0110 in binary

morton = 00111000 or 56



Research Question

Is a Space Filling Curve (SFC) approach an appropriate method for integrating the space and time components of point clouds in order to support efficient management and querying (use) in a DBMS?



Methodology: A Space Filling Curve approach

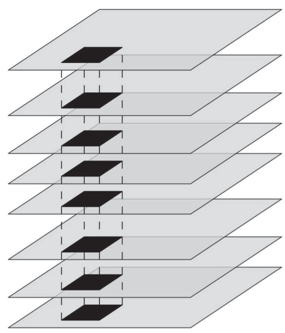
Requirements

Requirements for spatio-temporal data management [Adapted from Gaede and Gunther, 1998]:

- Should support ***operations*** other than just retrieval of the data.
- Should be ***dynamic***: support insertions
- Should be ***scalable***: adapt to growing database.
- Should be ***efficient*** in terms of time (and space): minimise as much as possible the number of disk accesses

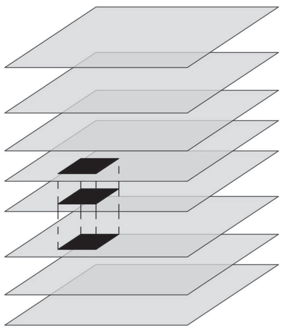
Important queries

- **Space** queries: all points located in a specific area over the complete time range



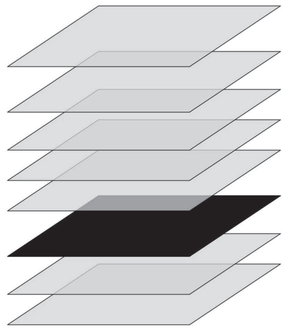
Important queries

- **Space - time** queries: all points located in a specific area during a specific time range



Important queries

- **Time** queries: all points of a specific time moment or range, for the whole spatial domain



A SFC approach

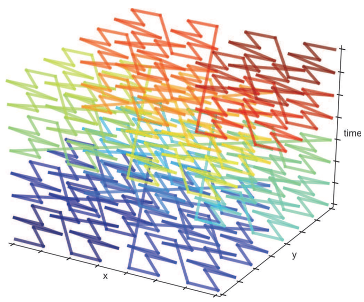
Structuring space and time is not a trivial problem. Contradiction:

- Points close in space and time should be stored (up to a certain extent) in contiguous blocks in disk, for *fast spatio-temporal retrieval*.

- Already organised points should not be reorganised when inserting new data, *for fast loading.*

A SFC approach

Integrated space and time approach: all dimensions have equal part in SFC.



Two treatments of z:

1. as an attribute.

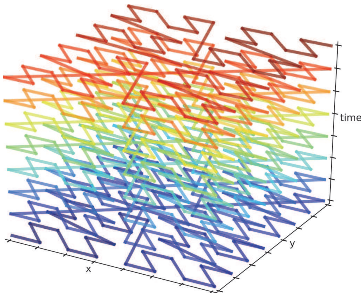
2. as part of the SFC key.

A SFC approach

Non-integrated space and time approach: time dominates over space.

Two treatments of z:

1. as an attribute.
2. as part of the SFC key.



A SFC approach - Loading

Two step approach:

- **Preparation**: Read files and convert to SFC key, according to
  - integration of space and time,
  - treatment of z and
  - scaling of time

  The data are bulk loaded into a normal heap table

- **Loading**: Sort the data based on the key into an Index Organised Table (data stored in the B-Tree index)

A SFC approach - Query

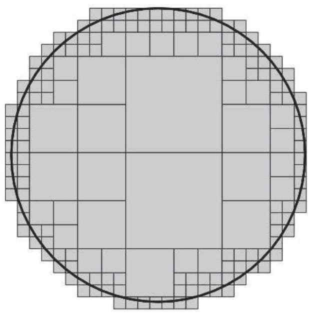
- Translation of the n-D query geometry into a number of continuous runs on the curve.
- Take advantage of the quadrant recursive characteristic of Morton curve: Use a Quadtree/ Octree/ 2^n-tree
- The maximum depth of the tree affects:
  - the number of ranges
  - the approximation of the query geometry

A SFC approach - Query

Multi-step query procedure

- Filter step: approximate query geometry using the 2^n-tree
- Fetch the approximated data and decode back to the original dimensions
- Refinement step: Detect the false hits using a Point in Polygon operation, or time and z refinement.

A SFC approach - Query



Identify Tree Cells

Direct neighbour merging

Reduce the number of ranges without affecting the approximation, by merging neighbouring ranges.

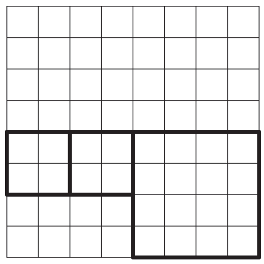


Figure a: Original 3 ranges

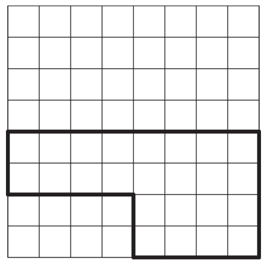
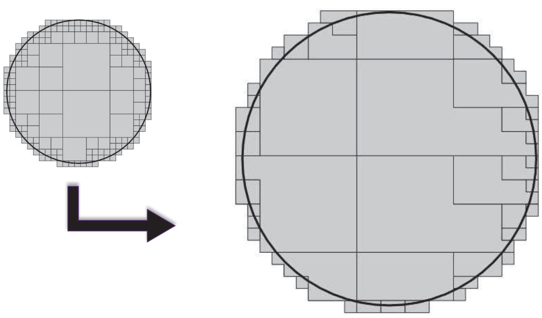


Figure b: Direct neighbour merging (1 range)

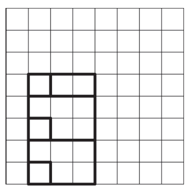
Direct neighbour merging



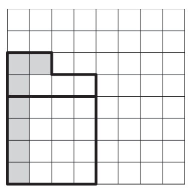
Merge of direct neighbours

Merging to maximum number

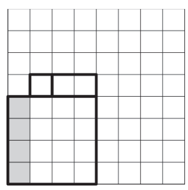
- Impose upper limit to the number of ranges
- Approximation gets slightly worse
- More false hits fetched during the filter step



Original 6 ranges



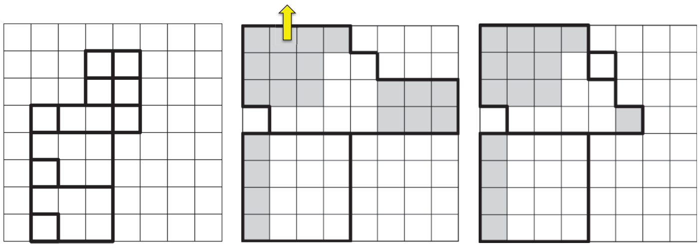
maximum 2 ranges



maximum 3 ranges

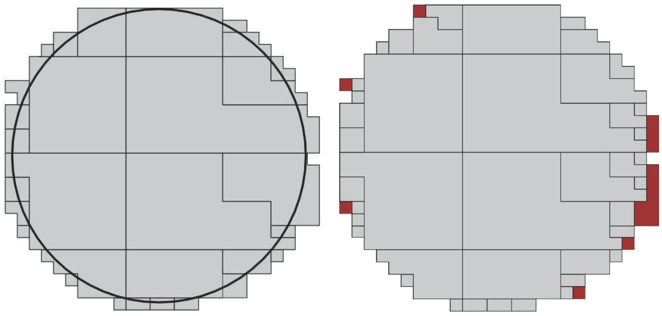
Merging to maximum number

Additional space



Original 11 ranges   maximum 2 ranges   maximum 3 ranges

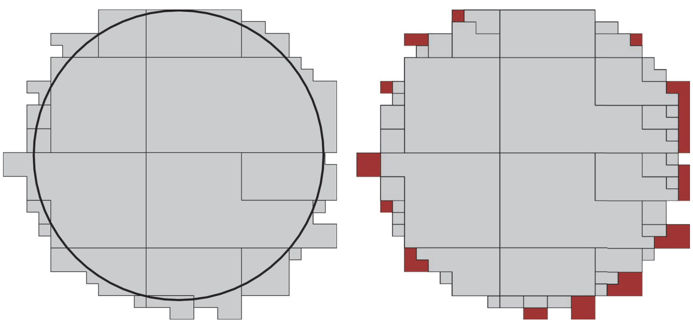
Merging to maximum number



merged   Expansion

Merge to max. number (30)

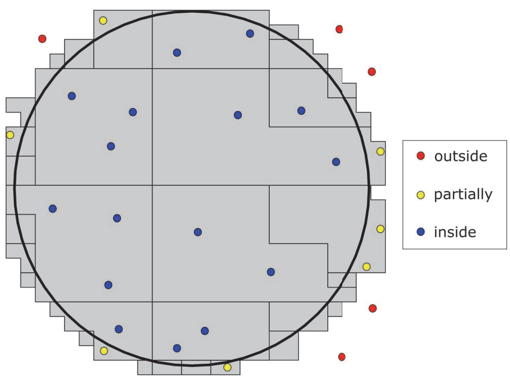
Merging to maximum number



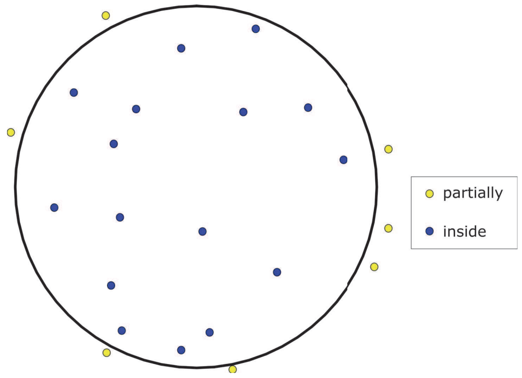
merged   Expansion

Merge to max. number (20)

A SFC approach - Query



A SFC approach - Query



A SFC approach - Query

Multi-step query procedure

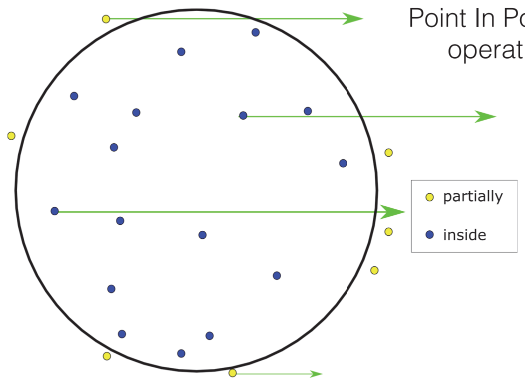
- Filter step: approximate query geometry using the 2^n-tree
- Fetch the approximated data and decode back to the original dimensions
- Refinement step: Detect the false hits using a Point in Polygon operation, or time and z refinement.

A SFC approach - Query

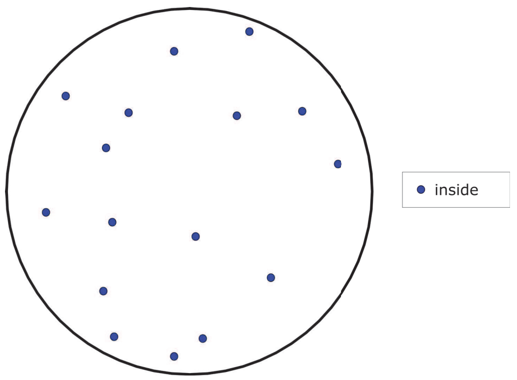
Multi-step query procedure

- Filter step: approximate query geometry using the 2^n-tree
- Fetch the approximated data and decode back to the original dimensions
- Refinement step: Detect the false hits using a Point in Polygon operation, or time and z refinement.

A SFC approach - Query



A SFC approach - Query

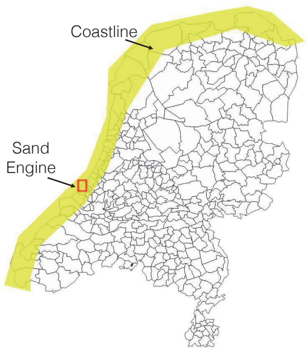


Results

Benchmark design

- Measure performance of storage space, loading time and query response time
- Datasets
  - Sand Engine
  - Coastline of the NL

Dataset	Time resolution	Spatial resolution	Points
Sand Engine	day	mm	100,000 pts/day
Coastline	year	cm	500 million pts/year



Benchmark design

- Benchmark stages

Table 1. The benchmark stages of the Sand Engine dataset

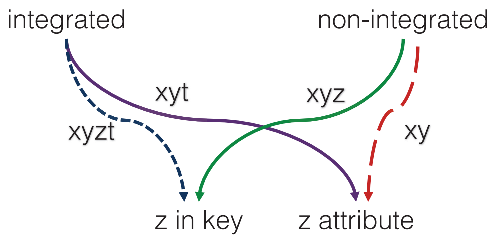
Benchmark	Points	Days	Size (MB)	Description
Small	18 M	230	347	2000 - 2002
Medium	44 M	554	836	2000 - 2006
Large	74 M	931	1414	2000 - 2015

Table 2. The benchmark stages of the Coastline dataset

Benchmark	Points	Years	Size (GB)	Description
Small	500 M	1	9.4	2012
Medium	995 M	2	18.7	2012 - 2013
Large	2020 M	4	37.9	2013 - 2015

Benchmark design

- 4 combinations



Results Loading

Sand Engine

Approach	Time (s)			Size (MB)	Points	
	conversion	Load heap	Load IOT		Heap	IOT
xy - S	105.43	11.79	13.60	471	18,147,709	18,147,709
xy - M	145.14	16.56	49.65	1130	25,561,106	43,708,815
xy - L	167.75	19.72	78.00	1897	30,205,111	73,913,926
xyz - S	352.37	9.91	10.5	368	18,147,709	18,147,709
xyz - M	498.79	14.24	34.07	885	25,561,106	43,708,815
xyz - L	590.00	16.77	61.71	1495	30,205,111	73,913,926
xyt - S	349.68	11.79	13.09	471	18,147,709	18,147,709
xyt - M	492.29	16.56	40.39	1130	25,561,106	43,708,815
xyt - L	594.10	19.72	74.11	1897	30,205,111	73,913,926
xyzt - S	435.48	11.79	10.78	386	18,147,709	18,147,709
xyzt - M	604.27	16.56	33.21	927	25,561,106	43,708,815
xyzt - L	722.08	19.72	57.96	1566	30,205,111	73,913,926

Results Loading

- The **SFC conversion** is the most expensive phase.
- Adding one more dimension in the key decreases the performance of the conversion.

Approach	Time (s)			Size (MB)	Points	
	conversion	Load heap	Load IOT		Heap	IOT
xy - S	105.43	11.79	13.60	471	18,147,709	18,147,709
xy - M	145.14	16.56	49.65	1130	25,561,106	43,708,815
xy - L	167.75	19.72	78.00	1897	30,205,111	73,913,926
xyz - S	352.37	9.91	10.5	368	18,147,709	18,147,709
xyz - M	498.79	14.24	34.07	885	25,561,106	43,708,815
xyz - L	590.00	16.77	61.71	1495	30,205,111	73,913,926
xyt - S	349.68	11.79	13.09	471	18,147,709	18,147,709
xyt - M	492.29	16.56	40.39	1130	25,561,106	43,708,815
xyt - L	594.10	19.72	74.11	1897	30,205,111	73,913,926
xyzt - S	435.48	11.79	10.78	386	18,147,709	18,147,709
xyzt - M	604.27	16.56	33.21	927	25,561,106	43,708,815
xyzt - L	722.08	19.72	57.96	1566	30,205,111	73,913,926

Results Loading

- **Loading into the heap** table is not affected by the benchmark case used.

Approach	Time (s)			Size (MB)	Points	
	conversio	Load heap	Load IOT		Heap	IOT
xy - S	105.43	11.79	13.60	471	18,147,709	18,147,709
xy - M	145.14	16.56	49.65	1130	25,561,106	43,708,815
xy - L	167.75	19.72	78.00	1897	30,205,111	73,913,926
xyz - S	352.37	9.91	10.5	368	18,147,709	18,147,709
xyz - M	498.79	14.24	34.07	885	25,561,106	43,708,815
xyz - L	590.00	16.77	61.71	1495	30,205,111	73,913,926
xyt - S	349.68	11.79	13.09	471	18,147,709	18,147,709
xyt - M	492.29	16.56	40.39	1130	25,561,106	43,708,815
xyt - L	594.10	19.72	74.11	1897	30,205,111	73,913,926
xyzt - S	435.48	11.79	10.78	386	18,147,709	18,147,709
xyzt - M	604.27	16.56	33.21	927	25,561,106	43,708,815
xyzt - L	722.08	19.72	57.96	1566	30,205,111	73,913,926

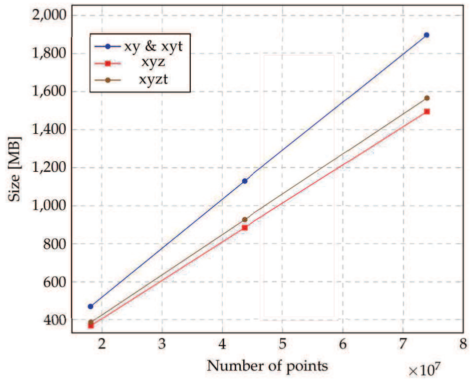
Results Loading

- The **creation of the IOT** is dependent only on the treatment of z used.
- The IOT is created faster when treating z as part of the key.

Approach	Time (s)			Size (MB)	Points	
	conversion	Loac heap	Load IOT		Heap	IOT
xy - S	105.43	11.7	13.60	471	18,147,709	18,147,709
xy - M	145.14	16.5	49.65	1130	25,561,106	43,708,815
xy - L	167.75	19.7	78.00	1897	30,205,111	73,913,926
xyz - S	352.37	9.91	10.5	368	18,147,709	18,147,709
xyz - M	498.79	14.2	34.07	885	25,561,106	43,708,815
xyz - L	590.00	16.7	61.71	1495	30,205,111	73,913,926
xyt - S	349.68	11.7	13.09	471	18,147,709	18,147,709
xyt - M	492.29	16.5	40.39	1130	25,561,106	43,708,815
xyt - L	594.10	19.7	74.11	1897	30,205,111	73,913,926
xyzt - S	435.48	11.7	10.78	386	18,147,709	18,147,709
xyzt - M	604.27	16.5	33.21	927	25,561,106	43,708,815
xyzt - L	722.08	19.7	57.96	1566	30,205,111	73,913,926

Results Loading

- The **storage requirements** are affected only by the treatment of z.
- Treating z as an attribute increases the storage.

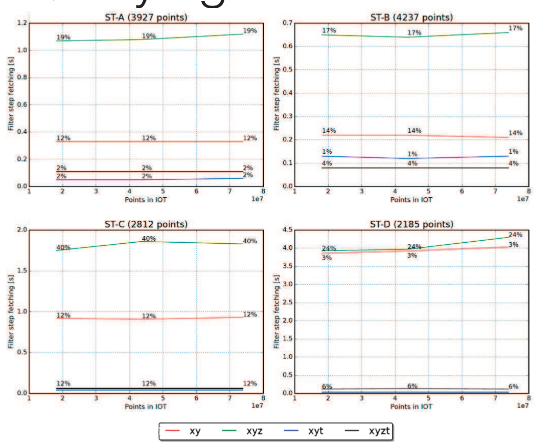


Results Querying

- Test the scalability of the queries
- Focus on the fetching time of the filter step that directly uses the structure. The rest of the steps can be improved in performance and are not analysed.

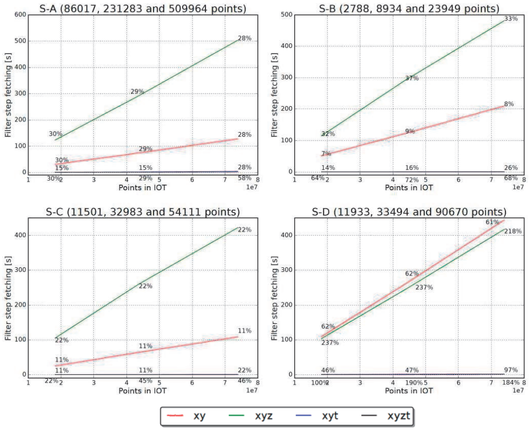
Results Querying

Space – time queries



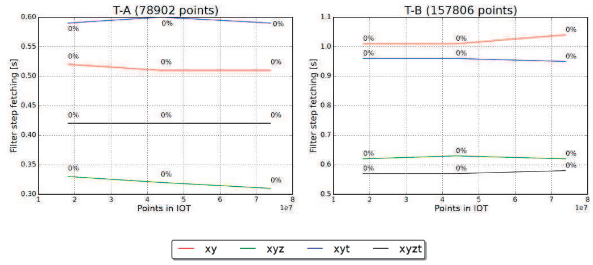
Results Querying

Space only queries



Results Querying

Time only queries



Conclusion & Future work

Conclusions

- Designed and executed a benchmark for dynamic point clouds
- Two integrations of space and time and, two treatments of z
- Integrated approach presented better query response times, compared to non-integrated for the specific use case (both treatments of z possible)
- Key aspect of the implementation: Index Organised Table

Future work

- Native database functionality (encoding, decoding, range generation)
- Investigate a different SFC
- Investigate parallel processing
- Up-scaled benchmark of trillion points
- Investigate the generation of blocks: compression

Thank you for your attention!

References

- Gaede, V. and Gunther,O. (1998). Multidimensional access methods. ACM Computing Surveys (CSUR), 30(2):170–231.
- van Oosterom, P., Martinez-Rubi, O., Ivanova, M., Horhammer, M., Geringer, D., Ravada, S., Tijssen, T., Kodde, M., and Gonc¸alves, R. (2015). Massive point cloud data management: Design, implementation and execution of a point cloud benchmark. Computers & Graphics, 49 :92 –125 .