OTB – Onderzoek voor
de gebouwde omgeving
Faculteit Bouwkunde

Definitief Rapport

Exploratieve Puntenwolken

Edward Verbree en Peter van Oosterom

21 december 2015
GISt Rapport No. 68

TU Delft Delft University of Technology

Challenge the future

Exploratieve Puntenwolken

Dit ‘vliegende startproject’ is uitgevoerd in opdracht van:
de Raamovereenkomst Rijkswaterstaat – TU Delft

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

21 december 2016
GISt Rapport No. 68

21 december 2015

OTB – Onderzoek voor de gebouwde omgeving
Faculteit Bouwkunde, Technische Universiteit Delft
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

© Copyright 2015 by OTB - Research for the Built Environment, Faculty of Architecture and the Built Environment, Delft University of Technology.
No part of this report may be reproduced in any form by print, photo print, microfilm or any other means, without written permission from the copyright holder.

Inhoudsopgave

1 Inleiding 2
1.1 Vraagstelling 2
1.2 Leeswijzer 2
2 Beantwoording onderzoeksvragen 5
2.1 Toegankelijkheid puntenwolken 5
2.2 Inzichtelijkheid puntenwolken 5
2.3 Algemene verwerking en analyse puntenwolken 6
2.4 Exploratieve verwerking en analyse puntenwolken 6
2.4.1 Verkeersveiligheidsanalyses 7
2.4.2 EchoScope 7
2.4.3 Voorbereiding werkzaamheden 8
2.4.4 Inspecties 8
2.4.5 Temporele aspecten 8
2.5 Link MSc Geomatics Synthese Projecten 8
3 Aanbevelingen 10
3.1 Multi Level of Detail (LoD) puntenwolken 10
3.2 Standaardisatie puntenwolken (formats en services) 10
3.3 Ontwikkeling analyses en algoritmen direct op puntenwolken 11
3.4 Verbeteren tools (direct gebruik en interactie) 11
3.5 Spatio-temporele puntenwolken 11
4 Verwijzingen 12
A Appendix Synthesis Project: Pointless 13
B Appendix Synthesis Project: XYZ 23
C Appendix Synthesis Project: OWL 30
D Appendix Studiedag – Visie Rijkswaterstaat 39
E Appendix Studiedag – Insight explorative point clouds 49

1 Inleiding

Binnen de bedrijfsprocessen van Rijkswaterstaat wordt de inwinning, verwerking, visualisatie en analyse van puntenwolkdata steeds belangrijker. Tot op heden wordt deze data vooral gezien als bronmateriaal voor het afleiden van meer hanteerbare producten als rasterrepresentaties en 3D-modellen, waarna het overgrote deel van de puntenwolkdata in principe nog wel beschikbaar is, maar nauwelijks wordt gebruikt. Sinds kort groeit het besef dat de originele puntenwolkdata een zeer grote meerwaarde hebben voor toepassingen waarbij de eindgebruiker door middel van interactieve, exploratieve, visualisatiemethoden en –technieken zélf inzicht en begrip opbouwt van datgene dat door de puntenwolkdata in kaart is gebracht.

De (zeer) grote omvang van puntenwolkdata is tot nu toe een belemmerde factor geweest in deze ontwikkeling. Daarbij komt dat voor de exploratieve verwerking en analyse specialistische software, kennis, vaardigheid en inzicht noodzakelijk is. Door onderzoek en ontwikkeling op het gebied van de directe ontsluiting en visualisatie van *massive point clouds* zijn recent grote stappen gezet in het wegwerken van deze belemmeringen. Het wordt daardoor steeds belangrijker de mogelijkheden van *explorative point clouds* te onderzoeken binnen het brede werkveld van Rijkswaterstaat.

1.1 Vraagstelling

Dit vliegende startproject richtte zich op het onderkennen, onderzoeken en het bieden van oplossingsrichtingen voor het direct en exploratief gebruik van puntenwolkdata. Daartoe waren de volgende onderzoeksvragen leidend:

- op welke wijze is puntenwolkdata toegankelijk te maken binnen de bedrijfsprocessen van Rijkswaterstaat?;
- op welke wijze is puntenwolkdata inzichtelijk en begrijpelijk te maken voor deze grote groep eindgebruikers en ter ondersteuning van welke taken (zoals vegetatiemonitoring, zichtanalyse, overstromingsmodellen, deformatiedetectie, zonlichtanalyse, etc.)?;
- welke algemene exploratieve puntenwolkdataverwerking- en analysetools zijn er reeds beschikbaar binnen de bedrijfsprocessen van Rijkswaterstaat?;
- welke specifieke exploratieve puntenwolkdataverwerking- en analysetools moeten er ontwikkeld worden om de inherente meerwaarde van puntenwolkdata expliciet ten gunste te laten komen binnen de bedrijfsprocessen van Rijkswaterstaat.

1.2 Leeswijzer

Dit document beoogt Rijkswaterstaat een duidelijke visie mee te geven omtrent de meerwaarde van directe en exploratieve toepassingen van puntenwolkdata binnen de bedrijfsprocessen van Rijkswaterstaat.

Gezien de korte doorlooptijd van dit project (september 2015 – december 2015), de aansluiting van dit project op het *Massive Point Clouds for eSciences* project (www.pointclouds.nl), en het door Rijkswaterstaat (Dick ten Napel) mede begeleide MSc Geomatics Synthesis Project *Direct Computing with Explorative Point Clouds* (september 2015 - november 2015) worden resultaten en inzichten van bovengenoemde andere projecten binnen dit verslag (her)gebruikt ter illustratie van de recente ontwikkelingen en toekomstige toepassingsmogelijkheden binnen Rijkswaterstaat. Een deel van de informatieoverdracht richting Rijkswaterstaat heeft dan ook al plaatsgevonden door feedback op de door de studenten opgeleverde rapportages en presentaties tijdens het afsluitende Symposium van het Synthesis Project op 6 november 2015.

Een andere belangrijke ontwikkeling is de Open Geospatial Consortium Point Cloud Domain Working Group (OGC Point Cloud DWG). De doelstelling (problem statement) van deze Point Cloud DWG is als volgt omschreven: *"OGC has a number of Working Groups to discuss issues relating to services, domain communities, and various data formats. However, point cloud data has often been overlooked or treated in the same buckets as images or terrain. Today the use of point clouds is growing at a rapid rate and can be found in a variety of domains including utilities, mining, 3D modeling, etc. Point cloud data is currently stored in many formats, some now de facto standards, defined for many domains such as Multi-dimensional Scientific data, LiDAR data, Elevation data, Seismic data, Bathymetric data, Meteorological data, and Fixed/Mobile consumer sensors (IoT). With so many uses of point cloud data but little standardization, a variety of different formats exist. OGC membership has registered a concern that without development of Best Practices or consensus standards, divergence will continue in the community and interoperability will be inhibited."*
Zie: http://www.opengeospatial.org/projects/groups/pointclouddwg.

Edward Verbree heeft in het kader van dit project op 15 september 2015 tijdens de OGC Technical Meeting in Nottingham een voordracht gehouden en Peter van Oosterom is sindsdien co-chair van deze OGC Point Cloud DWG. Het Ministerie van Infrastructuur en Milieu is *Associate OGC Member* van het OGC. Rijkswaterstaat kan dus de bijeenkomsten van deze OGC Point Cloud DWG bijwonen en daarmee mede de inhoud van de agenda bepalen.

Tot slot was er op 8 december 2015 de studiedag: *Management of Massive Point Cloud Data: wet and dry (2)*, georganiseerd door het Nederlands Centrum voor Geodesie en Geo-informatica (NCG), de Oracle Gebruikersgroep Holland (OGh), de TU Delft, en het Netherlands eScience Center;
Zie: http://www.gdmc.nl:8080/mpc/documents/mpc-seminar-8-dec15
Tijdens deze druk bezochte (>100 deelnemers) studiedag was er wederom veel aandacht voor inwinning, databeheer, en toepassingen (zowel nat als droog) van puntenwolken, met o.a. een presentatie van Edward Verbree *Insight through explorative point clouds*. De geplande Rijkswaterstaat voordracht van Dick ten Napel over *Wet and dry point cloud acquisition and applications within RWS* kon helaas door ziekte niet geheel gepresenteerd worden, maar het voorbeeld van direct gebruik van een puntenwolk voor de inspectie van een scheepswrak (slide 9 in de presentatie) is door Edward Verbree wel expliciet besproken en getoond aan het publiek. De inhoud van de gehele Rijkswaterstaat presentatie wordt gebruikt in dit document en is ook opgenomen in de appendices.

Dit rapport volgt de structuur van de onderzoeksvragen, waarop direct in hoofdstuk 2 de onderzoeksvragen worden beantwoord Paragraaf 2.1 gaat in op het toegankelijk maken van puntenwolkdata binnen de bedrijfsprocessen van Rijkswaterstaat. Paragraaf 2.2 richt zich op welke wijze puntenwolkdata inzichtelijk en begrijpelijk te maken zijn voor deze grote groep eindgebruikers en ter ondersteuning van welke taken (zoals vegetatiemonitoring, zichtanalyse, overstromings-modellen, deformatiedetectie, zonlichtanalyse, etc.) Paragraaf 2.3 behandelt welke algemene

exploratieve puntenwolkdataverwerking en -analysetools er reeds beschikbaar zijn binnen de bedrijfsprocessen van Rijkswaterstaat. Tot slot beantwoordt paragraaf 2.4 welke specifieke exploratieve puntwolkdataverwerking en -analysetools er ontwikkeld moeten worden om de inherente meerwaarde van puntenwolkdata expliciet ten gunste te laten komen binnen de bedrijfsprocessen van Rijkswaterstaat.

Hoofdstuk 3 biedt met verdere aanbevelingen een opzet voor mogelijk vervolgonderzoek.

In appendix A zijn de Engelstalige samenvattingen (*extended summaries*) van de rapportages van het MSc Geomatics Synthese Project opgenomen. De volledige rapportages zijn op aanvraag beschikbaar. In appendix B staan de slides van de presentatie van Dick ten Napel van de *Management of Massive Point Cloud Data: wet and dry (2)* studiedag van 8 december 2015, en in appendix C de slides van Edward Verbree van diezelfde studiedag.

2 Beantwoording onderzoeksvragen

Dit beknopte hoofdstuk met conclusies en aanbevelingen beoogt Rijkswaterstaat een visie mee te geven omtrent de meerwaarde van directe en exploratieve toepassingen van puntenwolkdata binnen de bedrijfsprocessen van Rijkswaterstaat.

2.1 Toegankelijkheid puntenwolken

Rijkswaterstaat moet zich bewust zijn van het feit dat puntenwolkdata het eindproduct is van het inwinningsproces en tegelijkertijd het startpunt is van verdere bewerking, analyses en visualisaties. De beschikbare puntenwolkdata worden in het huidige werkproces na extractie van benodigde metingen of berekeningen van afgeleide producten dusdanig gearchiveerd (of weggegooid) dat verder gebruik niet eenvoudig mogelijk is.

Die afgeleide producten (bijvoorbeeld hoogte/diepte-grids, XY-dwarsdoorsnedes (profielen), Z-dwarsdoorsnedes (hoogte/diepte contouren)) worden veelal op basis van een tussenbewerking (interpolatie) op de puntenwolkdata verkregen waarna de link naar de originele puntenwolkdata verdwijnt. Hiermee verdwijnt ook een deel van de informatie, die mogelijk toch nuttig kan zijn voor verschillende toepassing. Voorwaarde is dan wel dat de data eenvoudig toegankelijk zijn. Echter ook het hergebruik van de puntenwolkdata in kader van andere toepassingen wordt lastig.

Door de link vanaf de afgeleide producten naar de originele puntenwolkdata beschikbaar te houden vanuit een database-omgeving of een goed ingerichte puntenwolkdataserver (op korte termijn een LAS/LAZ-file-server, mogelijk op de iets langere termijn een op OGC standaarden gebaseerde puntenwolk web-service; zie OGC DWG PC activiteiten) wordt het voor de eindgebruiker eenvoudiger de kwaliteit van deze producten goed te kunnen beoordelen. Het is dus noodzakelijk om niet alleen het resultaat van de op deze puntenwolkdata uitgevoerde bewerkingen beschikbaar te stellen, maar ook inzicht te bieden op de puntenwolkdata zelf.

2.2 Inzichtelijkheid puntenwolken

De beschikbare puntenwolkdata kan direct ontsloten worden vanuit de database-omgeving of LAS/LAZ-file-dataserver. Het soort bevragingen komt veelal neer op een relatief simpele *bounding box query* voor het afbakenen van het studiegebied met eventueel daarna een verdere *points-in-polygon* selectie. Daarnaast is ook de *perspective view query* (veel detail dichtbij, verder weg minder detail) van belang voor interactieve exploratie van puntenwolkdata. De geselecteerde punten zelf zullen dan vervolgens wél aan de eindgebruiker beschikbaar moeten worden gesteld, zodat deze dan op een directe en interactieve wijze gevisualiseerd en bewerkt kunnen worden. Eenvoudige en gebruikersvriendelijke interactie is hierbij van groot belang; zoeken en navigeren zowel via (overzicht/puntenwolk)kaart als via het opgeven van postcode of locatie (plaatsnaam, etc.).

Rijkswaterstaat heeft, juist om puntwolkdata aan een grote groep eindgebruikers beschikbaar te stellen, een grote voorkeur voor een web-gebaseerde (Web-GL) omgeving.

De bewerkingen zullen erop gericht moeten zijn om de meerwaarde van puntenwolkdata te verkennen door het combineren van meerdere puntenwolkdatasets (à la map-algebra) en het direct vanuit deze puntenwolkdata afleiden van dwarsprofielen, hoogte/diepte-contouren, zichtlijnen, hellingshoek, hellingsrichting, etc., etc. Belangrijk gegeven daarbij is dat link naar de onderliggende puntenwolkdata beschikbaar blijft.

2.3 Algemene verwerking en analyse puntenwolken

De hiervoor geschetste werkwijze maakt duidelijk dat op dit moment er vrijwel geen algemene exploratieve puntenwolkdataverwerking en -analysetools beschikbaar zijn. Alleen daar waar er nu aanleiding is om de originele puntenwolkdata beter te bekijken, te onderzoeken en informatie af te leiden (exploratieve data-analyse) worden beschikbare tools als ArcGIS (Esri) en Qinsy (QPS) ingezet.

Voordelen van direct gebruik zijn:
- geen informatieverlies;
- besparing van (doorloop)tijd omdat conversie van omvangrijke dataset bewerkelijk is;
- sommige operaties werken beter direct op puntenwolken.

Op dit gebied valt er dus nog veel winst te halen door het realiseren van direct op puntenwolkdata gebaseerde analyses t.b.v. taken als waterbodemdieptemonitoring, vegetatiemonitoring, zichtanalyse, overstromingsmodellen, deformatiedetectie, zonlichtanalyse, etc., etc. Geadviseerd wordt om voor één van de analysetechnieken verder te onderzoeken of deze kan worden omgezet van de huidige raster/vectorbasis naar een puntenwolkbasis. AHN is een belangrijke dataset en watermanagement een belangrijke toepassing. AHN is dus een eerste kandidaat voor het omzetten van een grid-gebaseerde stromingsanalyse (simulatie) naar een puntenwolk- gebaseerde analyse. Een andere optie zou kunnen zijn het realiseren van deformatieanalyse direct op tijdreeksen van puntenwolk data.

2.4 Exploratieve verwerking en analyse puntenwolken

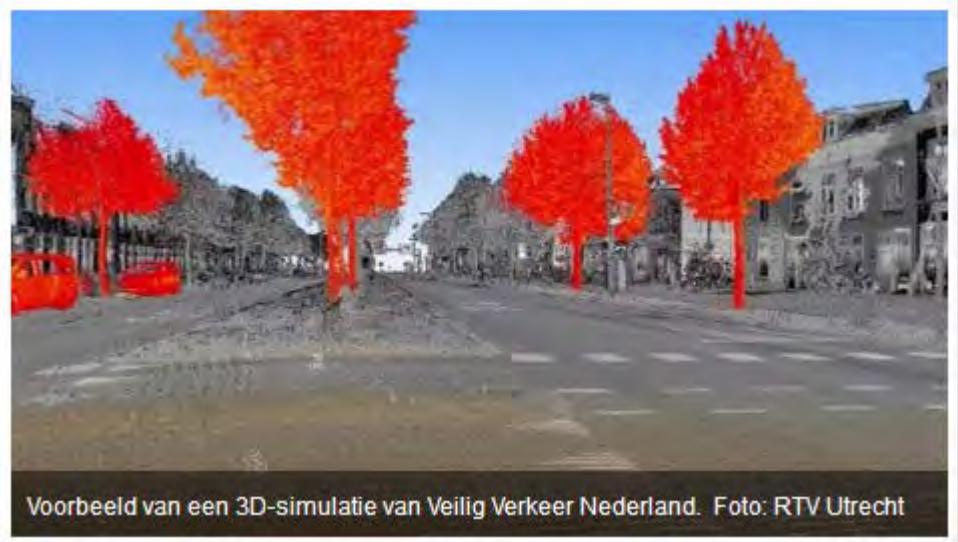
Alle puntenwolkdata, die als eindproduct van een inwinningsproces kan worden benut (denk aan AHN2-puntenwolk), dient - zoals al eerder aangeven - zoveel mogelijk direct bevraagd, gevisualiseerd, bewerkt en geanalyseerd te worden, waarbij de eindgebruiker ‘in control’ is. Tools voor interactieve visualisatie en visuele interactie moeten ontwikkeld, dan wel opgeschaald worden naar toepassingen voor zeer grote puntenwolkdata. Een treffend voorbeeld is de AHN2 pointcloud viewer (zie: www.pointclouds.nl), die het mogelijk maakt deze grote dataset zelfstandig te exploreren, waardoor o.a. fouten in de reeds uitgeleverde data opgespoord konden worden. Binnen deze viewer is al een aantal tools voor bevraging en meting beschikbaar.

Maar – zoals ook al eerder aangegeven – tools voor het direct afleiden van dwarsprofielen, hoogte/diepte-contouren, zichtlijnen, hellingshoek, hellingsrichting, etc., etc. dienen ook ontwikkeld te worden. Daarbij is het niet alleen van belang dat deze tools met zeer grote datasets kunnen omgaan, maar juist en vooral dat de link (inzicht) naar de originele puntenwolkdata beschikbaar blijft om cartografische exploratietools uit te kunnen nutten.

Gedurende dit project zijn er naast de initieel voorziene taken (gebaseerd op vegetatiemonitoring, zichtanalyse, overstromingsmodellen, deformatiedetectie, zonlichtanalyse), is er ook nog aandacht besteed aan de nodige andere taken. Dit was soms binnen het project (zoals het Synthese project) en soms buiten het project. In willekeurige volgorde volgt nu een opsomming:

2.4.1 Verkeersveiligheidsanalyses

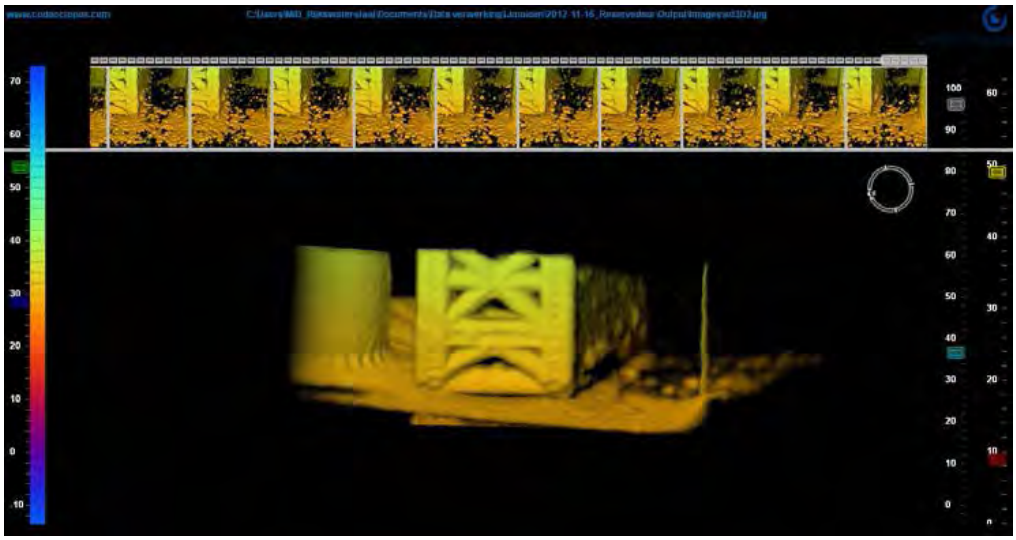
Verkeersveiligheidsanalyses vanuit verschillende perspectieven (kinderen, volwassenen, voetgangers, fietsers, automobilisten, buschauffeurs of vrachtwagenchauffeurs), identificeren van mogelijke zichtobstakels. In de analyse van verkeersveiligheid worden de punten die voor verkeersveiligheid horen bij zichtobstakels (bomen, geparkeerde auto's, etc.) rood gemaakt; zie project VVN Utrecht, bericht
http://www.rtvutrecht.nl/nieuws/1406903/3dbeelden-van-gevaarlijke-kruispunten-in-utrecht.html
(en dan filmpje kijken bij ong. 2:30 min).



Figuur 1: 3D-simulatie van Veilig Verkeer Nederland.

2.4.2 EchoScope

Ook bij andere inspectietaken kunnen puntenwolken worden geproduceerd en ingezet; zowel onder water met b.v. de echoscope (inspecties van sluizen) als op land (b.v. de inspectie van asfalt, zie: https://www.tno.nl/media/5498/automated_ravelling_inspection_and_maintenance_planning_on_porous_asphalt_in_the_netherlands.pdf



Figuur 2: EchoScope voorbeeld uit rapport A.J.B. Oosterveld, 2013.

2.4.3 Voorbereiding werkzaamheden

Voorbereidingsaanvraag offerte en verkennen huidige situatie via puntenwolk (vanaf de werkplek, niet in het veld); bijvoorbeeld in het kader van een voorbereidingsaanvraag van offertes en verkennen van werkzaamheden in het veld of bij het maken van kostenramingen bij een project: een 3D-weergave van een verkeerssituatie waarbij verkeersstromen elkaar ongelijkvloers kruisen om een goede inschatting te kunnen maken van de kosten bij een ingrijpende aanpassing van de infrastructuur.

2.4.4 Inspecties

Voor verschillende Rijkswaterstaat processen kan via de inzet van drones data worden ingewonnen, en de vaak resulterende stereobeelden kunnen worden gebruikt voor produceren van puntenwolken; zie de Brochure "Drones in het publieke domein" (2015) met als voorbeelden o.a. Crisisbeheersing (Oliebestrijding op de Rijn bij Tolkamer, Monitoring overstromingsrisico's Noordwaard), Verkeersmanagement (Verkeersmonitoring Knooppunt Emmeloord, Verkeerskundig onderzoek gemeente Zeist, Sporenonderzoek na ernstige autoongevallen), Beheer en onderhoud (Monitoring Zandmotor, Inspectie brug Babberich, Inspectie Rietkraag, Inspectie Zeelandbrug, Inspectie Windturbines), etc.

2.4.5 Temporele aspecten

Een andere nuttige toepassing van exploratieve puntenwolken is op basis van puntenwolken op meerdere momenten in de tijd, de verschillen opzoeken en deze interactief presenteren (en visualiseren en eventueel ook volumes, afstanden of andere maten berekenen).

2.5 Link MSc Geomatics Synthese Projecten

Binnen de samenwerkingsovereenkomst Rijkswaterstaat – TU Delft wordt er gestreefd naar een doorvertaling van projecten binnen het onderwijsprogramma. De onderzoeksopgave van dit vliegende start project 'Exploratieve Puntenwolken' is dan ook het thema geweest van het MSc Geomatics Synthese Project, *"A culminating group project at the end of the first year only for Geomatics students. The project allows students to combine the knowledge from the core programme and apply it to a real-world project while gaining hands-on experience in project management. The objective of this course is to undertake independent group work through a small but real-world research project in collaboration between companies and the university itself. The students are able to practise skills in acquisition, visualization, processing, analysis and interpretation of geospatial data, as well as project management - making this course an invaluable experience."*

In de periode september-november 2015 heeft een 15-tal studenten (in groepjes van 5 studenten) een drietal (deel)projecten uitgevoerd:

- voor het routeren van drones door de vrije ruimte in de nabijheid van kunstwerken, gebouwen of andere obstakels heeft het Synthese Project "Pointless" (Appendix A) onderzocht of deze routeringsberekening direct op data van het puntenwolkmodel kan gebeuren. De inzet van drones is weer relevant voor diversen (inspectie)taken zoals hierboven reeds aangegeven.
- verrijken van een 'pure xyz' puntenwolk met intelligente classificatie van punten op basis van type object waartoe punten waarschijnlijk behoren, zonder objecten zelf te reconstrueren (weg, bebouwing, grasland, bos, water, etc.); zie Synthese Project XYX (appendix B). Hierdoor wordt de puntenwolk nog rijker en voor meer toepassingen potentieel nuttig.

- puntenwolken met RGB waarden zijn mogelijk ook voor meer exploratieve toepassingen nog beter te gebruiken door de betere herkenbaarheid via RGB kleuring van de puntenwolk tijdens de interactie. Vooral puntenwolkdata verkregen via dense point matching vanuit overlappende images zijn mogelijk zeer waardevol (naast de laser, of MBES puntenwolken); zie Synthese Project OWL (appendix C) en de recente MSc GIMA theses van resp. Sanne Douma en Tjako Smit.

3 Aanbevelingen

In het vorige hoofdstuk zijn de onderzoeksvragen beantwoord en is – met name in paragraaf 2.4 – aangegeven welke specifieke exploratieve verwerking en analysetools er ontwikkeld moeten worden om de inherente meerwaarde van puntenwolkdata expliciet ten gunste te laten komen binnen de bedrijfsprocessen van Rijkswaterstaat.

Daarnaast verdient het aanbeveling verder onderzoek te laten uitvoeren naar de onderstaande punten:

3.1 Multi Level of Detail (LoD) puntenwolken

De opslag, bevraging en presentatie van zeer omvangrijke puntenwolkdata is en blijft tegen de limieten van opslagcapaciteit, rekenkracht en visualisatiemogelijkheden aanlopen ondanks de inzet van slimme opslagstructuren, geoptimaliseerde databases en het gebruik van hoogwaardige grafische kaarten. Echter, niet ieder punt is even belangrijk. Door het slim toepassen van Level of Detail technieken worden kenmerkende punten eerder opgevraagd, eerder betrokken bij berekeningen en eerder getoond. Het in- en opdelen van ruimtelijke informatie in 'mate van belangrijkheid' of 'Level of Detail' is een beproefd concept, maar is nog niet uitontwikkeld voor puntenwolkdata. Het voornaamste vraagpunt daarbij is: "*All points are equal, but some points are more equal than others"* (vrij naar: Animal Farm, 1945). Hoe onderscheidt een punt zich van de omgeving? Welke punten dienen eerder in beeld te komen? Hoe om te gaan met *perspective view queries* veel detail dichtbij, dus meer punten in beeld; verder weg minder detail, dus minder (of geen punten) in beeld? *Smart mapping*: welke punten dienen visueel benadrukt te worden?, etc., etc.

3.2 Standaardisatie puntenwolken (formats en services)

Deze aanbeveling is min of meer in lijn met de *key activities* van de OGC Point Cloud 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."*

3.3 Ontwikkeling analyses en algoritmen direct op puntenwolken

Bewerkingen op, en onderling tussen, puntenwolkdata worden nu nog veelal uitgevoerd door tussenliggende datastructuren als Grids en TINs. Gevolg daarvan is dat er óf informatie verdwijnt (door interpolatie) óf de onderlinge relatie tussen de punten puur geometrisch wordt bepaald (door Delaunay criterium) en niet door de aard en het soort van de puntenwolkdata. Directe berekening op de puntenwolkdata kan dat soort problemen voorkomen; in hoeverre en welk voordeel er kan worden behaald dient verder onderzocht te worden.

3.4 Verbeteren tools (direct gebruik en interactie)

Puntenwolkdata heeft het in zich om een een-op-een (kaart)beeld van de werkelijkheid te kunnen tonen. Dat beeld wordt gevormd door een combinatie van interactieve visualisatie (de gebruiker bepaald wát en hóe er iets in kaart wordt gebracht), maar ook door visuele interactie (het communicatieproces tussen mens en computer via grafische voorstellingen op het scherm). Deze specifieke interactiecyclus dient voor puntenwolkdata nog verder ontwikkelt en bestudeert te worden.

3.5 Spatio-temporele puntenwolken

Puntenwolkdata is veelal oppervlaktedata; verkregen vanuit een scan of beeldopname. Een compleet andere bron van (dynamische) puntenwolkdata zijn bewegende objecten van alles waarin en waarop de mens zich voortbeweegt (auto's, schepen, vliegtuigen, etc., etc.), en de verplaatsingen van de mens zelf, en andere bewegende objecten. Al deze vervoersstromen bieden óók veel inzicht; in hoeverre en welk voordeel er behaald kan worden dient verder onderzocht te worden.

4 Verwijzingen

1. A.J.B. Oosterveld (2013). Inzet EchoScope voor Waterdistrict Noord-Holland. Verslag van opnames in omgeving IJmuiden en Schellingwoude. Opnameperiode 6 t/m 16 november 2012. Rapport Ministerie van Infrastructuur en Milieu, Rijkswaterstaat, Meet- en Informatiedienst Noord-Holland, Februari 2013.
2. Aad van den Burg en Cees van der Pligt (2015). Drones in het publieke domein, Een quickscan, Rapport Ministerie van Infrastructuur en Milieu, Rijkswaterstaat, augustus 2015.
3. Sanne Douma (2015). Dike cross section measurements based on Semi-Global Matched stereo photo DEMs versus LiDAR DEMs. MSc GIMA thesis, december 2015. http://www.msc-gima.nl/thesis-archive/
4. Tjako Smit (2015). Automatic dike body extraction from point clouds from high-resolution stereo photos. MSc GIMA thesis, december 2015. http://www.msc-gima.nl/thesis-archive/

A Appendix Synthesis Project: Pointless

2 Executive Summary

This executive summary describes the global content of Project Pointless, and contains the most important technical details of the project. It is written as an informative summary of the full report, and can be read as a separate document.

2.1 Background

Point clouds are (often large) collections of points registered in three dimensional space, which can be stored in combination with additional properties such as colour and signal strength. Together, they can provide a realistic view of the world as it was measured by laser scanning technologies or derived using photogrammetry. Point clouds are a collection of (x,y,z) coordinates and properties, which have the potential to provide a wealth of information, and can be used to accurately describe features, for example the point cloud of the Project Pointless team in figure 1. Although the individual points merely represent (x,y,z) and colour (r,g,b) values, the combination of these points provide an accurate representation of the five team members. The trained human eye can recognize their facial expressions, posture, type of clothing, and can even recognize that the team member in the middle is injured since he has a cast around his ankle. For applications involving point clouds, often intermediate geometric models such as 3D BIM models are generated, and the original dataset is deleted or archived (van der Spek and Verbree, 2015). This considerably reduces the amount of data, but also inherently infers a loss of information. Because these models are a simplification of the original point cloud, the full richness and details of captured data can go lost (Verbree and van Oosterom, 2014). However, if operations are carried out using the 'raw' data directly, this has the potential to enrich the original point cloud dataset.

A more traditional implementation for using point clouds to create a 3D representation of a real world object, is to take a 2D representation (a polygon for example) and using the point cloud to extrude this polygon to level of detail (LOD) 1, 2 or 3. However, if this method is performed and the original point cloud is afterwards not being used anymore, then this creates a model with a lot less detail than was originally captured. Figure 2 shows what the 4 different levels of detail mean: LOD1 is a basic block model of a building and with LOD2 there is a more accurate roof shape. LOD3 includes more accurate facades and LOD4 models the interior of a building as well. Figure 3 shows LOD1, LOD2 and LOD3 models compared to the original point cloud. Even the LOD4 model, which also models interior space, does not have any information about the indoor world. It merely allows the user to go inside the extruded polygon. The original point cloud with all its detail does have information about the inside world, and applications like the ones presented in this report are the first steps into the full exploitation of all the richness of explorative point clouds. The usage of explorative point clouds holds major advantages such as no need for an intermediate geometric model. This saves storage space, time and most importantly retains all details included in the point cloud. Furthermore it is also easier to update and can be reused.

Point clouds can be classified into different categories with respect to their measurement environment; aerial point clouds are obtained using aerial vehicles such as airplanes and drones, terrestrial

Figure 1: Point cloud of team Pointless (from left to right: Ivo de Liefde, Florian Fichtner, Erik Heeres, Olivier Rodenberg and Tom Broersen)

Figure 2: Levels of detail in GIS
image courtesy to Biljecki (2015)

Figure 3: Levels of detail of Bouwpub compared to point cloud

point clouds are obtained using ground equipment such as cars or total stations, while indoor point clouds focus specifically on interior spaces. Each of these types of point clouds serve their own range of applications. For indoor point clouds these applications range from automatic generation of indoor models to object recognition and path finding. The focus of applications involving indoor point clouds is almost always on identification of the boundaries of space, or of the objects inside it, instead of on the space itself. However, classification of the empty space can serve a multitude of purposes. One very obvious application would be (empty) volume estimation. A company might want to estimate how much storage space is left in a silo or warehouse, for which estimation of the volume of empty space from a point cloud can be useful. Also, the empty space can be used for (walkable) path finding and creating realistic point cloud walkthrough applications, as it can restrict users to walk only through the empty space in the point cloud. Especially for collision avoidance for autonomous robotics finding and structuring the empty space can be useful (Payeur, 2006). Space fitting queries provide another potential use case, these can be used to check whether large objects fit through the empty space. Rijkswaterstaat currently uses a single maximum height value to indicate whether a vehicle fits under a bridge or through a tunnel. For many cases this maximum height value will be sufficient, but when the shape of the vehicle or the shape of the bridge or tunnel are more complex, a single height value will not be an effective solution. A three-dimensional model of the empty space can then be of added value to calculate directly whether the passing object can fit through. This can also be useful in maritime applications to verify whether a ship fits through a specific waterway, or in indoor applications for calculating a possible route for a drone to fly through a building. Despite the many potential uses, no earlier research to identify empty space in point clouds is available in present day scientific literature.

Octrees are based on the recursive and non-uniform subdivision of space and are a common approach to structure and segment 3D point clouds. Earlier research, like Wang and Tseng (2011), Zhou et al. (2011) or Schön et al. (2013), used an octree structure to segment point clouds, but also applications such as Potree and the Point Cloud Library, make use of an octree structure. Octree structures provide an efficient way of indexing the point cloud, and are handy for neighbourhood operations (Burrough, 1986). Furthermore, they can be used for the partitioning of space and result in a hierarchical structure. Project Pointless hypothesizes that by using the octree structure to segment and index the point cloud, also the empty space can be derived in an efficient way.

2.2 Purpose statement

Many potential use cases of empty space classification were identified in section 1.1. Project Pointless tries to comply to these use cases by taking a novel approach for processing point clouds; instead of identifying boundaries of objects, it will classify the empty, pointless, space in any interior point cloud by using the efficient octree structure. This will be done directly from the raw point cloud data, without creating intermediate models of reality, thus retaining all the details present in the original point cloud. A smart workflow will be designed which efficiently identifies and visualises the empty space directly from the point cloud source. This workflow should be universally reusable as a basis for other projects involving (interior) point clouds. The computation adds a structure to the raw point cloud dataset which can be understood and interpreted by computers. To showcase the potential of empty space identification, an indoor routing algorithm will be used to compute the shortest path for a drone through a building on the basis of the identified empty space. The project objective can be described in one sentence:

"Creating a smart and universally reusable workflow which efficiently identifies and visualises the empty, pointless, space directly from any interior point cloud source using an octree approach, and computes the shortest path through an empty space."

2.3 Methods

Following from the purpose statement in section 1.2, this section will describe the methods used in this project to generate the octree, and accordingly identify and visualise the empty space in interior point clouds. This section will also explain the approach taken to implement the use case, using an indoor routing algorithm to compute the shortest path through a building on the basis of the identified empty space. The descriptions in this section will be brief. For an extensive documentation, the reader is referred to chapter 4: concepts of the Final Review report.

2.3.1 Octree generation and empty space finding

This project makes use of an octree structure to index and segment the original point cloud. An octree is a tree structure where the volume is recursively subdivided "into eight congruent disjoint cubes (called octants) until blocks of a uniform colour are obtained, or a predetermined level of decomposition is reached" (Samet, 1988). This means that with every level of the tree the entire space is split into 8 blocks. The space continues to recursively subdivide into 8 blocks until either the maximum amount of levels is reached or a block is completely empty.

The octree implemented by Project Pointless is a so-called linear octree, which makes use of materialised paths to refer to the nodes in the tree. Such materialised paths are built by taking a point and traversing the tree from root node to leaf node, thereby iteratively adding to a string the name of the node where the point is located in at that level of the tree. This means that every level in

the octree will be represented by one single digit in the materialised path. After octree generation, every point in the point cloud has one materialised path assigned to it, referring to the leaf node it is located in. The cells in the octree will be enumerated in z-order, with values of the octants ranging from 0 to 7. To obtain the maximum resolution for the empty cells in the octree, the non-empty nodes (thus nodes containing points) are always forced to split up to the maximum user-specified tree level. There can thus be no non-empty leaf nodes in the tree below this maximum level.

To generate the materialised paths for every point in the point cloud, a technique called binary masking is used. Since every cell in the octree splits into eight smaller octants, it is possible to encode with only 3 bits in which octant a point is located. Along every dimension of a cell, the point can be at one of the two sides of the split line, which can be indicated by either a 1 or a 0 (true / false). In three-dimensional space it is thus possible to specify for a certain point in which octant it is located by a combination of three bits; x[0 or 1], y[0 or 1], z[0 or 1]. To accordingly generate the materialised path using this method, first a conversion of the coordinates of the point into binary numbers for x,y, and z takes place. Then every digit in these binary numbers will be iteratively compared to the digits in a binary mask, the form of which depends on the current level in the octree. For level 1 (and binary x, y, z coordinates of [101,011,001]), the binary mask will be 100 (so only the first digit, corresponding to the first level in the tree, is evaluated). Iteratively comparing the digits in the (x,y,z) coordinates to the binary mask using a Boolean AND operator, will result in the following outcomes: True for x, False, for y, and False for z. Then, if x is true, a 1 is added to a temporary variable, if y is true, a 2 is added, and if z is true, a 4 is added. The sum of these numbers is the number that represents the node in which the point is located in at the current level in the tree. This procedure is repeated for every level in the tree, every time concatenating the node numbers to form the materialised path. For the above procedure to work, the point cloud first needs to be translated and scaled. The translation is necessary to define the lower left corner of the front of the bounding box as the origin of the point cloud. The scaling is needed so that the method can be used for any type and size of point cloud, and so that every cell has coordinates represented by integers between 0 and $2n$. Then by removing the decimals from every point's coordinates, the points can be snapped to the particular cell it is located in.

The above procedure results in a list of materialised paths, referring to all the non-empty leaf nodes where the points in the point cloud are located in. This procedure does not yet explicitly store the empty nodes. To find these, a list is generated containing all non-empty nodes. This list is compared to a list of all potential nodes, and if any of these potential nodes are not in the list of non-empty nodes, then this means the node is empty. The result will be a list of empty nodes, which can occur at any level in the octree, and can have different sizes depending on their level in the tree. The geometry of these nodes then has to be determined. The size of the nodes in local coordinates is equal to $2n$, where n is equal to the difference between the current level and the maximum number of levels in the tree. The origin point of the nodes can be found by using the same binary masking technique to convert from materialised paths back to decimal coordinates. All points and the octree structure with the empty and nonempty nodes are stored in a DBMS for efficient access and retrieval.

2.3.2 Visualisation

After the points and octree structure have been stored in the database, it is possible to use the data in further applications. However, it is useful to have a tool that allows visual inspection of the data to verify whether the generation of the octree structure and the empty nodes was successful. To do this a WebGL viewer was implemented that can retrieve data from the octree in the database and visualise it. The use of WebGL requires no additional plugins to be installed.

The visualisation of the empty space is done using a combination of three.js and FLASK. Three.js is an open source JavaScript library for visualising three dimensional scenes in the browser. Its goal is to make WebGL easy work with. Since Three.js runs in the browser a website had to be created which runs Three.js and connects it to the database with the empty space and the point cloud. FLASK has been selected to deal with the backend of this website. FLASK "is a micro framework for Python" (Flask, 2014). This allows to use the backend to call the octree generation script which was also written in Python. After the script has been called FLASK connects to the Postgres database and loads the data into a Three.js scene.

2.3.3 Use case: 3D route calculation

The A* algorithm was used to compute the shortest path between two points in the point cloud. A path is here defined as the route between the two nodes the points are located in. To find the shortest route, recursive exploration of adjacent nodes is required (Noto and Sato, 2000). These so-called 'neighbours' of every node are found by using a method developed by Vörös (2000). This method is able to find all inner and outer equal-, smaller-, and larger-sized neighbours for every node by utilizing the hierarchical structure of the octree.

2.4 Support for the conclusions

The project successfully created a Python script, which takes a point cloud from any .las file, and efficiently generates a linear octree using binary masking. All points in the original point cloud, with the materialised path of the nonempty leaf nodes they are located in, were stored in a PostgreSQL database. From these materialised paths of the nonempty leaf nodes, the empty nodes and their geometry can be quickly extracted. The current octree implementation was written such that an octree of 8 levels deep is generated, which equates to +/- 10cm resolution for the dataset used here. The speed of the octree generation script was evaluated by comparing between generation of octrees of 6, 7, and 8 levels deep. This comparison showed that doubling the octree resolution from 6 to 7 levels deep, causes only a small increase in total processing time (+/- 10%), and the increase from 7 to 8 levels causes an increase of only 2%. Moreover, almost all of this increase in processing time can be accounted to the increased time needed to write the results to the database. The generation of the octree itself, and derivation of the empty space from this octree, only takes 1% more time when increasing the number of levels in the octree from 6 to 8. This indicates that the octree generation

Figure 4: visualisation of empty space in a slice of a point cloud

script scales very well when increasing resolution. Moreover, 65% to 74% (depending on the depth of the octree) of the total processing time is spent on writing points to a file, thus indicating that the script can become even faster if a method is found to avoid this step. The influence of a change in point cloud size was also tested. The script was tested with point cloud sizes of 2.3 million, 4.6 million, and 9.2 million points. This comparison shows that the increase in the amount of points in the point cloud has an impact on performance, the time spent on writing points to a file more than doubles when doubling the amount of points. Also, the relative amount of time spent on this procedure increasing from 65% for 2.3 million points, to 74% for 9.2 million points. Writing points to a text file is by far the least scalable component of the script. However, here again the generation of the octree and derivation of the empty space shows to scale well, with the amount of processing time increasing 3.6 times when including 4 times more points. It is presumably the combination of fast snapping of points to the leaf nodes, binary masking, and membership testing using sets in Python, which makes the octree generation and empty space finding so efficient.

Visualisation of the points and empty space using the currently implemented combination of WebGL, FLASK, and three.js, worked for relatively small point clouds (e.g. 2 million points). A slice of the data (empty nodes and points) is visualised in figure 4. It shows that the empty space in between the points is successfully classified into the empty cells, and the cells become smaller when getting nearer to the points. The current implementation of the viewer is however not sufficient for larger point clouds, which can cause crashes. This could be solved in the future by using Potree, or an adaptation of Potree for visualisation purposes.

The route finding algorithm (Figure 5) has been timed while computing routes in octrees with different amount of levels. These numbers show that there is still room for improvement of the performance: for six levels the algorithm takes less than 20 seconds to find a path, for seven levels it takes two minutes. However for eight levels it suddenly takes almost half an hour to compute a route. This is because the number of neighbours that should be checked in the A* algorithm increases exponentially with the addition of an extra level.

Figure 5: visualisation of a route (gray cubes) through the empty space in a point cloud

2.5 Conclusions

Project Pointless aimed to create a smart and universally reusable workflow which efficiently identifies the empty space in a point cloud, store this in an efficient way, visualise it and use this octree structure in an example application. The empty space as well as the points were structured in an octree. This allowed the team to build a successful application of a path finding algorithm using only the empty space of a point cloud. The performance and scalability have been statistically examined. The results indicate that the octree implementation is efficient and scalable. It is presumably the combination of fast snapping of points to the leaf nodes, binary masking, and membership testing using sets in Python, which makes the octree generation and empty space finding so efficient.
The method of Project Pointless does not use intermediate geometric model and therefore is explorative. The raw point cloud data is merely indexed and stored in a database management system. The procedure is fast and automated, enriches point clouds with a structure that can be understood by computers, while keeping all advantages of the raw data set.

From a performance viewpoint, the octree generation and empty space finding can be considered successful, and possible further optimisations have already been identified. Furthermore, the script is flexible since it is able to use any .las file containing an interior point cloud as input, due to the translation and scaling operations carried out on the points before further processing. The script is also free for anyone to use and completely open source. The implementation is also reliable since the script checks all input parameters on start-up to prevent catching invalid input, it makes sure that the input point cloud is in a valid file format, and whether a database connection can be established. Furthermore, the .las file containing the point cloud is first copied to ensure the original stays intact and can still be worked with during processing. Finally, before writing results to a database, they are first written to a .csv, which minimizes the connection to the DBMS and thus failure when this connection is lost.

The current implementation still knows some limitations. The performance and scalability of the implementation is limited by the writing of points to a file, and subsequently copying this file into the DBMS. Also the octree generation script is not yet scalable to multiple users since it can locally run only once at a time. Finally, visualisation of the points and empty space using the currently implemented combination of WebGL, FLASK, and three.js, is not sufficient for larger point clouds, which can cause crashes. This could be solved in the future by using Potree, or an adaptation of Potree for visualisation purposes.

The use case of 3D route calculation was found to work well and does not require any intermediate geometric models or graph network to find a route in 3D. All calculations are performed directly using the original point cloud source and generated octree structure. The current implementation of the 3D path finding algorithm is already able to successfully find a route through the empty space, but knows some limitations in its performance. Neighbours are currently checked on the fly, which means that they have to be recomputed every time the script is executed. Also the script does not scale well. However, these are limitations that can be overcome, and the use case effectively showed that the octree structure and empty space generated here has great potential for real-world use cases, and allows to quickly develop useful applications on top of it. The idea of finding and structuring the empty space and making direct use of an explorative point cloud was successfully implemented and also proved to be fruitful for future applications.

The project successfully fulfilled its goals set out at the start. The media campaign utilised by Project Pointless attracted significant attention, which shows that the topic and idea of Project Pointless meets the state of the art and that people are interested in the result as well as its process.

B Appendix Synthesis Project: XYZ

Chapter 1: Executive Summary

1. Introduction

In today's technologically driven era, the terms position and location are highly affecting our lives. In addition to that, the introduction of virtual earths and street view applications have well shifted our perception of geo-information from 2D towards 3D. This shift was also affected by new technologies like LiDAR that automated the generation of 3D city models. LiDAR technologies within a decade matured leading to the use of point clouds into everyday applications. But despite the immense amount of information hidden inside a point cloud, its information in today's city modelling paradigm is often not exploited to its best potential. In the majority of today's projects, point clouds get thrown away after 3d models are created. These point clouds are for a lot of users too big to handle, and therefore not easily replaced by traditional data sources like surveying. However, it is foreseen that with the increasing power of mobile devices like smartphones or tablets and their browsers, those days of insufficient computing power are coming to an end. It is therefore relevant to explore ways to utilise the point cloud data directly.

Cyclomedia, a specialist in capturing, processing and visualising the urban environment by using cyclorama images, is the project client of this research. For Cyclomedia, the road is the central object for its data collection and, therefore, investigating the ways to extract road information from point clouds is central for this research. In addition to that, the road structure in a city can be very useful in extracting other features, like: buildings, traffic signs, vegetation or cars. Besides road extraction, the aim is to label point clouds with their function or composite material. These labels can help in order to create certainty about surface conditions, such as the surface infiltration capacity (SIC). A SIC map of an urban area could be very useful input for evidence-based urban design of water-sensitive/ low-impact neighbourhoods and water management schemes for reducing flood vulnerability.

In this research, the aim is to create a workflow that results in a dataset that will help urban planners, urbanists and road- and city maintenance organisations in doing their jobs better and more efficient. In other words, the aim is at enriching the data with class labels, giving it added value and new user possibilities. The central question in this research is therefore: *How can a point cloud be semantically enriched by providing a labelling process using geometrical properties and colour values?*

The executive summary is organised as following. Section 2 describes the Data collection equipment and process. Section 3 describes the pre-processing workflow that together with the workflow presented in Section 4 was proven to give the best results for street level extraction. The summary ends with section 5 where general conclusions and recommendations for this project are given.

2. Data collection

The data collection for this project is performed by Cyclomedia's mobile mapper system. The system used was their newly developed prototype system with LiDAR, five cameras and a combination of GNSS receiver and INS system to establish the position of the car. The LiDAR scanner used was the Velodyne HDL 32 that has 32 channels and catches 700,000 points per second with ± 2 cm accuracy. The horizontal field of view of the scanner is 360°. The five cameras, on the other hand, together produce 360° panorama images. These images are used in the process of stereo-imaging to create the point cloud, and to provide the LiDAR point cloud with colour. In this project, only the stereo imaging point cloud is used which contains points with a three-dimensional position calculated through the observation of a point by multiple images.

3. Pre-processing

3.1. Non-ground filtering

Point clouds contain rich information about the scene that they were obtained from. However, not all information is relevant for all applications. In the case examined in this project, the focus area lies on the ground surface. This means that the non-ground points (buildings, trees, cars) should be removed from the dataset. After investigating several algorithms, the non-ground filtering process offered in LAStools was proven to produce the best results.

3.2. Relevant classes

Designing a workflow that would extract the road information and its surroundings first requires the definition of the application relevant classes. In this case road extraction and materials focused on street level are examined. The most relevant subcategories are: *Grass, Road, Cycle path* and *Footpath*. Those categories represent how humans interpret the street level. For more specific applications, (e.g. the surface infiltration capacity), where the composition of materials is needed, a specialist would benefit from a classified point cloud with the classes *asphalt*, *grass* and *tiles*.

3.3. Point cloud indexing

Since point clouds from their nature are big data, quick access to the points can be incorporated by applying a spatial index. After comparing octrees and kD-trees, the kD-tree data structure turned out to provide the most balanced structure and the quickest access to the neighbouring points of each point. The reason for this is that the kD-tree makes use of a 1D distance comparison for indexing whereas in the case of an octree 3D distance calculations are needed.

3.4. Geometrical properties of the neighbourhood

The nearest neighbours are used to calculate 3D features for every point, because geometrical information of one specific point does not provide much information on its own. However, if a neighbourhood of the point is analysed, much can be understood with the geometrical patterns.

These geometrical properties of a point's neighbourhood can be calculated with the principal component analysis (PCA). PCA can be used to estimate the eigenvalues which can be used to infer different measures, like linearity and planarity, that describe the regions' properties. The only measures that are eventually used in our algorithm are the normal and the curvature. Other researched geometrical properties of a point's neighbourhood, like planarity, scattering or the sum of the eigenvalues, are not sufficient to make a distinction between already horizontal planes.

3.5. Colour spaces transformation

Applications for point clouds have been greatly extended since the introduction of colour in them. This is because humans can correlate colours to semantics and thus infer a great amount of relevant information. The colour information of the point cloud is therefore used to cluster points together that possibly belong to the same class. However, the RGB colour space that is easy interpreted by a human, possesses a lot of difficulties in computer vision since colour differences are not uniform. Therefore, more uniform colour spaces namely, HSV, HSI and CIE were investigated. Those spaces transform the highly correlated RGB values to less correlated components. The results of those colour transformations were tested for their applicability in distinguishing between the different classes in later stages.

4. Point cloud labelling

Labelling a point cloud in this research means getting those parts out of the point cloud necessary for assessing a certain task. Training data is used to make it possible to label, classify, parts of a point cloud. This training data, samples, are manually selected clusters of points with a known class. In choosing the right samples, it is important to pick various regions and clusters of points that show inconsistencies in colour, compared to the overall class. Experience showed that these variations in training data increased the classification results.
In order to identify the method that produces the best outcome, gets the necessary parts out of the point cloud most accurate, two methodologies were investigated: a point based and a region based classification method.

4.1. Point based classification

The point based method investigates whether it is possible to label a point cloud by only utilising the colour properties of each individual point. Point based labelling is only possible if samples for each class are available. From these samples, the range of the colour value in HSV or CIELab is determined, by calculating the

minimum and the maximum value for each class. Next, it is checked if the colour properties for every individual point are either between the minimum and maximum of each component per class or between the mean minus or plus 2 sigmas. Machine learning algorithms were also tested on point properties. Although some promising results came from these classifications, point based methods were proven to be error prone since a lot of colour inconsistencies are present in the point cloud. CIELab, in general, gave the best distinction but the accuracy assessment for point based methods proved to be not sufficient.

4.2. Region based classification

Region based classification schemes take the neighbourhood properties of the points into consideration. These properties of the individual point are used to determine if neighbouring points hold the same properties, and belong to the same region or class, or not. The neighbourhood properties that are used are curvature, which gives an indication of the smoothness of the underlying surface, the normal, which is the vector that is perpendicular to the points region and the colour and colour distance.

To grow a region, a seed point is chosen, and all neighbouring points are tested if they hold similar values on certain properties, if they do, and therefore belong to the same class, the points are added to the region. This process is then repeated for all new points in that region, until no newer points are found that belong to that region.

In the applied region growing algorithm, a low curvature threshold is used, because the majority of the classes are found inside the same plane, therefore raising the curvature threshold leads to under-segmentation. Under-segmentation means that two different classes are labelled as one. While, when a surface is divided into really small surfaces, you can call this over-segmentation. Under-segmentation happened with curvature thresholds over 0.001. The decision was made that over-segmentation is more acceptable than under-segmentation. Intentionally causing over-segmentation is, at least in theory, much simpler to correct than under-segmentation, since human intervention cannot be avoided in the second case.

4.2.1. Unsupervised classification

The previously caused over-segmentation of the region growing algorithm is tackled by a merging algorithm that is based on colour similarity. The main idea is that the small regions (with points less than 5000) that are formed in the region growing algorithm, are merged with their neighbouring regions that have the most similar colour. The colour similarity is compared by calculating the euclidean distance of the components of the CIELab space, excluding the L component that represents lightness. The most similar region is the one with the lowest value. The result shows more relevant regions. However, there are still cases of over-segmentation and even under-segmentation, where two classes are labelled as one.

4.2.2. Supervised classification

The supervised approach to classifying a point cloud requires a set of training sample data that have labels associated with them. This sample data comes from a scene and is selected with prior knowledge about the scene. This means that, for the area of interest of this project, samples are collected that belong to the road, cycle path, tiles and grass.

The different classification methods that are tested can roughly be subdivided in two categories: First, Gaussian processes, which assume that the class properties are Gaussian distributed, and a class gets assigned to a point or a region with the highest probability. Second, machine learning algorithms, that are implemented by making use of the python module sklearn, that allows classification through simple vectorised training samples and data. From these training samples, a decision space is created and used to classify the regions or points. The output of the Gaussian process is most satisfactory, with a mapping accuracy of 92% for road, 77% for cyclepath, 80% for tiles and 99% accuracy for grass. The sklearn module didn't give satisfactory results, as a lot of points, that don't belong to one of the predefined classes, are assigned a class. This is caused by the functioning of the algorithm, that is written in such a way that each point or every region gets assigned to the class with the highest score, even if it is very unlikely that this class should be assigned.

4.2.3. Smoothing

To tackle the errors of omission and commision, a smoothing algorithm is developed. This method is based on image smoothing filters that remove noise. The algorithm traverses all points in the classified point cloud. For every point it checks its k nearest neighbours and calculates what is the most frequent class among these neighbours. Next, the point gets reclassified if the original classification differs from the outcome of the smoothing algorithm. This process offers both a visual and an accuracy advantage to the final result.

5. Conclusions & recommendations

5.1. Conclusions

From the outcomes of this research, it becomes clear that a region-based supervised classification using training samples is the best way to distinguish between the contextually relevant classes. This is because the colour information of the point cloud is often times less accurate than the positional information. The colours of our point cloud show some differences in relation to reality and therefore unsupervised schemes present over-segmented results. Furthermore, over-segmentation with unsupervised classification can be the result of shadow and water on surfaces, which then appear darker, worn out colours of the

road/cycle path markings, different materials of tiles and other noise in general. To work around those inaccuracies, training samples can be provided. This way a supervised classification captures all appearances of a class whereas an unsupervised classification would create different regions. However, during the project, the researchers realised that one training sample per class is not sufficient. Therefore, the user has to provide the best combination of training samples per scene that will not be excessive in number and amount, but representative enough of the colours present. Finally, extending the method to other areas can be succeeded, but only if the provided training samples are representative.

5.2. Recommendations

The following recommendations are provided as future work for the project. Recommendations one and two are based on difficulties and inefficiencies in the manually picking of training samples. The third recommendation is a solution for the difficulties we faced in the region growing on colour values.

1. Explore the possibilities to match the regions with land use maps.
2. Provide the positional information of the mapping vehicle/device with (each part of) the point cloud.
3. Radiometrically enhance the cyclorama images before the point cloud creation.

Future work can be found in relating the point clouds, in particular the semantically labelled parts, to over simplified models of reality mostly utilised today by scientists.

C Appendix Synthesis Project: OWL

1. Executive Summary

This report is the result of the synthesis project, fulfilled by the project group Owl. The synthesis project is part of the second year of the master Geomatics from the TU Delft and has the topic: 'Exploring Photogrammmetric point clouds'. The main objective of the project is to explore the operations that can be carried out with the 'raw' dataset and that would directly enrich the use of point cloud data. 3 different project groups focused on a different topic related to point clouds. Every group got assigned to a supervisor and a representor of a company with related business activities to the topic.

Project group Owl consist of 5 students with all a different background, such as computer science, urbanism, industrial design and archaeology. The knowledge acquired in the first year of the Geomatics for the Built Environment master will be used in orde to come to a high-level product. The project group chose the topic point clouds acquired by drones. In order to get insight in this topic and to get experienced assistance during the project, Owl was supported by Pieter Franken; The founder and managing director of Skeye. A
Dutch company, which is specialized in aerial survey and inspection, mainly done by drones using photogrammetry. During the acquaintance with Skeye at the start of the project, the company brings up an issue, which they often deal with. Clients hire Skeye in some cases to obtain a point cloud for surface inspections. Clients expect to see certain details, such as cracks in this point cloud, but unfortunately they are not or barely visible. This is a pity, since they are visible on the original taken photos. It would therefore be interesting to find a proper, understandable and usable way to link the original photos to the resulting point cloud. This issue has been taken by the Owl project group as a challenge and their main research question will be:

'How can the understanding of point cloud data and the related imagery be improved for the inexperienced user?'

When Skeye creates a dense point cloud and highly detailed photos for visual inspections many images are inseparable. Team Owl wants the same quality of dataset to simulate the problem.

According to the project, the primary goal was to use a part of BK City to gather the point cloud and after a consultation with Pieter Franken Owl decided to measure the Bouwpub, as it is an independent building with a manageable size. The point cloud will be acquired via the photogrammetry method, as Skeye is used to it and laser-scan equipment is generally speaking too heavy and too expensive to attach underneath a drone. The project has a time frame of 2 months and involves several stakeholders. In order to achieve the final goal of the project and to fulfill all the requirements of the different stakeholders a certain
organization structure and project management is necessary in order to keep the progress of the project on track.

At first a Rich Picture is created, which is an applicable visual tool to clarify the steps of the project and the involved stakeholders. The resulting Rich Picture scheme is the base structure for the project.
Due to this scheme, Owl has defined 5 different stakeholders; Owl, TU Delft, Skeye, end users and the legislation. For these separate stakeholders the requirements are defined in order to fulfill their expectations. These requirements are analyzed by the MoSCoW analysis. A simple analysis method to define the priorities in a project program. The method divides the requirements in 4 different categories; 'Must have', 'should have', 'could have' and 'won't'. From this analysis the key requirements of the project can be derived. Key requirements are unavoidable in order to achieve the final goals of the project. These are indicated by the categories 'must have' and 'should have'. Owl defined the following key requirements; educational goals, project management, data acquisition, data processing, quality control, safety, application, research, deliverables and innovation. Besides requirements, the project is subject to certain constraints as well. These constraints are defined by Owl during brainstorming sessions. Each of these constraints has a different impact on the project.

The main constraint of the project was the lack of a fly-permit for the drone in the vicinity of BK City, due to legislation restrictions and the processing and management of the huge amount of data from a high quality point cloud. 3 other constraints with less impact were the calculation power of computers, permits for the data acquisition and usable software toolkits.

A next step in the project management is creating a Risk Map on the progress of the project. This map indicates by the likelihood of appearance and its impact, to which extent a certain constraint appears as a risk. According to this analysis, the absence of a drone flying permit, is the highest risk for this project, as it turned out later. After these analyses a planning is made for the entire project by a Gannt Chart, indicating 4 main phases in the project; Preparation, data acquisition, data processing and application development. The preparation phase contain several different elements; Literature review, in which we conducted research to comparable projects, in order to get fruitful ideas to implement in our own project. Literature about projects of Skeye (equipment and acquisition considerations) and the Via Appia project(software considerations) project were of great support for the Owl project.

Besides, the literature review contained research to the possibilities and impossibilities of using a drone for our project due to legislation issues. It turned out that it was impossible to use a drone above built environment such as the BK City area and furthermore it is located in a fly-restriction zone of Rotterdam The Hague Airport. This situation forced Owl to create an alternative data acquisition plan, which is derived after brainstorm sessions and consultations with Pieter Franken. After several ideas passed in review, like holding the drone in our hands on a ladder, The final plan was to capture the photos manually while standing on a cherry picker. For this, fixed settings of the camera needed to be calculated, such as camera positions. A proposal of the data acquisition project has been drafted and discussed with Skeye. Additionaly, permission was required by the facility management of BK City to execute the project, since several parking lots needed to be kept empty during the data acquisition.

After handing in the data acquisition plan and some consultations Owl received permission and could start the data acquisition. This operation required much equipment, which were borrowed or hired by different parties and tested in advanced in order to execute the data acquisition as smooth as possible. For the photogrammetry it is important to be aware of the main factors of a camera; photo resolution, camera calibration, angles between photos, orientation quality, photo redundancy and ground control points.

Skeye provided us a digital single-lens reflex camera, which is very suitable for photogrammetry due to its many setting options. It is the most commonly used camera by Skeye for their drones surveys. Taking photos for photogrammetry, 80% vertical overlap and 60% horizontal overlap have been taken into account in order to get a sufficient quality of the point cloud. By taking this into account, together with certain camera settings, all the camera positions can be calculated. Owl uses a fixed distance to the façade of 5 meters on the advice of Pieter Franken. This distance delivers the appropriate level of detail while not being redundant with this camera.

By computing the real focal length (52,02 mm), the distance covered by one image in the width (3,46m) and height (2,31 m) direction and the ground sample distance (16 pixels/cm), it turned out that 1136 images needed to be taken from 56 different locations and on 16 different heights. These different positions were drawn with chalks on the floor indicating where to ride the cherry picker. The camera positions runs in lines along all separate facades, including oblique pictures. At every location photos were taken from the 16 different heights; from ground level to 10 meters. Using the cherry picker, hired from a commercial party, these height could be achieved. The data acquisition occurred in teams of three people, of which one member took the actual pictures and checks if its settings are sufficient, one member records the orientation and position of the camera and a third member operated the cherry picker and regulating the heights with a distance laser, which was provided by CGI company.

As inertial measurement unit, Owl created an app. This app was downloaded on a smartphone. While holding it next to camera it records per photo the pitch, roll and yaw and its x and y coordinate via GPS. These records where automatically sent to an email address. Since these parameters are just an estimation, they have a lot of deviation. To determine the z-coordinate, Owl used the fixed heights, which could be related to the AHN2 afterwards.

On every façade 8 ground control points were placed, existing of plastic black-and-white grids. Normally, this number of ground control points is abundant, but later it offers more opportunities for control checks of the data. The ground control points represent accurate positions on which the rest of the façade can be referred to. They are measured by a Total Station, which provides much more accuracy than GPS on a smartphone. The Total Station measured the ground control points in a local system, which further exists of a central measure point, a backsight, 4 fixed measure points with each a proper view on a different façade. This local system is in a later phase related to the RD coordinate system with the help of a dGPS. In addition the point cloud can be related to the RD coordinate system too. Both the dGPS and the Total Station were borrowed from the Faculty of Civil Engineering and the local system is georeferenced via the spatial adjustment tool in ArcGIS. Further equipment, like ladders or safety equipment was provided from the facility management of BK City.

Unfortunately Owl didn't manage to accomplish all the goals of the acquisition plan. The original plan was to manage the data acquisition in one day, but in the end we spent 7 days on photographing the bouwpub. Besides due to unlevelled surface, we were not able to reach every camera position. We managed to have a full coverage on these spots, but in the end it results in a less dense point cloud on these parts.

In the end 1059 photos were captured in RAW and JPEG format. The data set contained more photos than originally was planned. This occured because of several reasons; While capturing the photos, we realized that the four corners of the bouwpub needed to be photographed as well. Further extra camera positions on each side of all facades had to be added in order to create sufficient overlap at these spots. At last, due to obstacles, we were only able to photograph one façade from 4 meter distance, resulting in smaller and more pictures. Other issues that appear were water and window reflection and autofocal matters of the camera, resulting in less dense spots in the point cloud. After the data acquisition the processing phase started. The processing part is the connective part between the acquisition and application part. The goal of the processing part is to create a dense point cloud. The input used for the processing are: images, camera orientations and GCP coordinates.

In figure 60 the processing work flow is shown. Processing was the one of the most time consuming parts of the project since the dataset was very big and hard to process it. Moreover, the processing power and memory needed didn't permit to use our PCs but a powerful computer located in Amsterdam. For this reason, Team Viewer was used in order to access it. With Team Viewer you can stream the desktop of another computer, while you have input from your own computer. Team viewer slowed down further processing due to bad internet connection. The processing started with data preparation. The data preparation includes the image type transformation from raw to jpeg; the organization of all cameras' information into csv files. During data preparation the data were converted from local coordinate system to Amersfoort RD New (28992) system. The conversion was done with the use of FME software. Moreover the exact location of the markers was defined using a DGPS.

The next steps of processing were done using Agisoft Photoscan since this software fulfills the projects expectations properly compared to other software (i.e. Pix4D). The steps followed in Photoscan are as follows. First of all, we created 4 chunks, one for

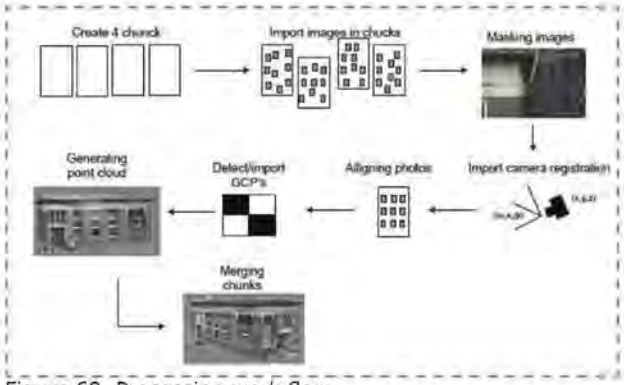


Figure 60, Processing work flow

For each chunk the reference settings had to be setted while the camera information was loaded from the csv files which were created in data preparation step. The next step is to apply masks manually for each image; the photo alignment follows. Furthermore, we detected the markers by using the appropriate command in Photoscan and we assigned the coordinates information to the markers. After chunk alignment, the chunks were merged and cleaned in order to remove noise. Finally, we exported the point cloud and the cameras in order to use them at the application that was build by the Owl team and we sub-sampled it for the application's requirements.

The point cloud was created from 1.132 images, while 28 GCPs were used for the georeferencing. The point-cloud created by the above procedure consists out of around 450.000.000 points. To minimize noise we removed certain regions and points that they did not belong to Bouwpub. As a result, the resulted point cloud appears to be clear. The point cloud that is going to be used for the application represents wall A and D and initially was made out of 291.810.899 points. However, to keep the application working smoothly, this dataset was downsampled. The size of the data set is an important aspect of the performance of the application.

The main goal for the project's application is to create an efficient and effective relation between points and pictures. Therefore several approaches have been considered. First of all, the way of creating the application, Owl could use the existing software, which offers the required functionality. Nevertheless this software is not existing. Secondly Owl could design a new software program from scratch, which demands a lot of programming, which is highly depend on time and skills. Owl chose therefore to use best of both worlds, by developing an extension of an existing software program.

The minimal requirements for such software are the ability to work with point clouds, support the related file formats, must have a proper rendering 3D viewer and allows extensions by plug-ins, which are preferably compatible with python, since this is a programming language of which all members are acquainted with.

Owl considered several software packages for the application; VTK, Paraview, CloudCompare, Meshlab and M Lidar viewer. The advantage of Paraview is that it supports python, while Meshlab and CloudCompare only supports C++. Since python is the only programming language of which all the group members are familiar with, this is the most feasible option. Paraview contains a practical visualisation and interaction functionality, which is for primary use very handy. Nevertheless a drawback of this program, is the fact that it allows to less control over the point cloud for the application, and especially the rendering speed decreased enormously. Therefore the project group decided only to use VTK, which is a toolkit of Paraview. Further Owl surveyed and tested what the best viewer is for our application; Several viewers have been tested, like the viewers of CloudCompare, Meshlab and M Lidar viewer. However, since these viewers do not have sufficient rendering performance Owl developed their own viewer supported by the Visualization ToolKit (VTK) with added functionality. After the software considerations, Owl analyzed the possibilities and impossibilities on data storage and relation finding.

The application has to use the data in order to provide the functionality it is intended to provide. This data comprises information about the camera, the photos and the point cloud. The data has to be stored in an efficient way so that it can be quickly accessed by the application.

We considered several approaches for data storage and looked at a file-based, database-based or hybrid solution. A hybrid solution appears to be the best approach. In this approach only the relations between photos and points are stored in the database and the other information is stored in files. With this approach there is no redundancy, the data can be quickly loaded and the data does not have to be transformed into another format before it can be rendered.

Step 1 of the application is finding the relations between the photos and points in the point cloud. Unfortunately, The relations between the photos and the points in the point cloud can not be exported from PhotoScan. Therefore it requires some 3D data processing to find these relations.

Relation finding is currently done by intersecting all points with the field of view (FoV) of a photo. According to this method, a point is related to a photo if it lies within the field of view of that photo.
This field of view is a pyramid shape that can be constructed using the internal and external camera parameters and optionally a field of view depth to limit the search range. Although this method yields a quick result, it is not entirely correct because not all points within the field of view are also in line of sight of the photo. Therefore a subsequent filtering step has to be conducted to filter out all points that are not within the line of sight of the photo.

We considered and partially implemented several methods to include line-of-sight analysis in the relation finding algorithm, but they are not included in the current application because they did not yield the desired result. The first method we tried was converting all points to small planes and then drawing a line from each point to the photo position to see if it intersect with a plane. If it does, it is not in line of sight from the photo position.

This method caused numerous crashes due to the large number of planes that were created. This even occurred when the point cloud was thinned. We chose an alternative method. The second method we tried was creating depth maps and then using these depth maps to find points which are not in line of sight. This partially worked because the depth map only contains the minimum values of the distances. However, the depth map also contained gaps due to the fact that the point cloud was thinned. As a consequence points farther away from the photo position were also included in the depth map. This caused the filtering process to produce no real good results and more implementation was needed to overcome this problem. These implementations did not make it in time to be included in the application.

The third method we tried was converting the point cloud to a volumetric model (voxels) and then drawing a line from each point to the photo position to see if it intersect with a voxel. If it does, it is not in line of sight from the photo position. We partially implemented this method, but the implementation did not yield the desired result because too many points where filtered out. This was due to several characteristics of the volumetric model and how the implementation worked.

The final application consisted of several python scripts. It starts with two exported text files from PhotoScan, of which one contain the point cloud (.xyz) and the other the camera position file (.txt). In the point cloud file the XYZ coordinates and the RGB values are stored. In the camera position the exact position and orientation of each camera are stored. The position is represented by its XYZ coordinates and the orientation is stored by the Omega, Phi and Kappa values, which represent the yaw, pitch and roll parameters. Further the text file starts with the file name of the related photo. At first the point cloud file will be processed by the xyz2vtp script, translating the points, represented by coordinates, into polygonal data, like vertices in a VTP file. This step enables to render the point with the help of the VTK. As input file it takes the pointcloud.xyz file, reads it, convert it and write it into a output file.

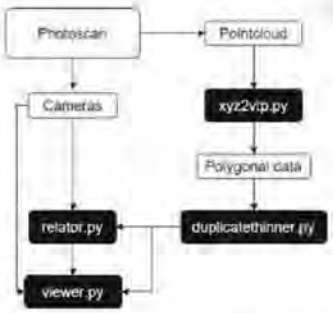


Figure 61, Application flow diagram

The next step in the application is the duplicatethinner script. This script is necessary in order to accomplish proper rendering performance with the highest possible point. The advantage of using the duplicatethinner script is that the data of a high-quality point cloud derived by PhotoScan can be maintained for other applications, while the data uses the pointcloud for rendering. The script takes a certain tolerance and the polygonal data file as input and creates a new point cloud with less points as output file. This tolerance value indicates the minimum distance between two points. the polygonal data file gets thinned by this tolerance via the BuildMapForDuplicatePoints function in VTK. The rendering performance of the output can easily be checked by the simpleviewer script. Ideally we would have an point cloud which is as dense as possible without rendering issues. Nevertheless this rendering performance and its related tolerance vary per point cloud depending on their size. This script enables to test the tolerance for separate point clouds, which makes this application useful for more point clouds. The next step is to relate the photos to the point cloud data set. The script relator.py takes the thinned point cloud file and the cameras text file as input and finds the relation to each other. These relations are stored in a Postgres database. The final part of the application is the viewer script, which is the bundle of all different data sets. It takes the cameras text file, the thinned point cloud and the relation database as input. Running this script will open a user interface in a new window, rendering the thinned point cloud. In this window the user is able to zoom in and out and navigate along the point cloud.

When the user clicks on a point on the point cloud, several blue dots will pop up at the exterior side of the point cloud, representing the position of the cameras in which the point is captured. By clicking on the blue dots, the field of view appears and the related photo pops up in a new window. In this window a dot indicates where the related point in the point cloud is located. In this render window it is allowed to pan and zoom in the photo, which is useful for fast detailed visual inspection and the relation between photos and points is intuitively shown to the user.

After the development, several tests have been executed to check the quality of the application. The application has been tested on 8 requirements. These requirements were created before and during development of the application and are based on the user experience with the application and on technical aspects of the application. These tests reveal that the application cannot handle point clouds with over 4 million points. Next to that it reveals that the relation finding between photos and points has to be improved. However, the application does provide a good and intuitive interface for the user and it clearly shows the relation between photos and points. It does give the user good insights and thereby fulfills its goals.

After conducting the project, Owl have some considerations for future work; For instance, it would be nice to export the relation between the points and photos straight from PhotoScan. At the end Owl achieve to find an user interface, which enables easy intuitive interaction between a point and its original picture. Although Owl managed to find filtering algorithms, which create relations between the points and the original pictures, the final solution has not found. The application is user friendly for customers, but the starting up button of the applications can be further improved. At last it is a pity that we couldn't make use of drones and were not able to discover the benefits of this technology compared to other point clouds.

2. Introduction

'Explorative point clouds' is the subject of this project. Shortly said: "A point cloud is a set of data points in a three dimensional coordinate system and often intended to represent the external surface of an object" (Wikipedia 2015). These point clouds are a valuable tool for all kind of spatial analyses and they are mostly used to translate into objects with less data. This is due to the fact point clouds are famously large in size, too large to handle you could say. Here lies the challenge for the teams of the synthesis project. How to enrich the use of raw point cloud data. To enhance our professionalism, team building, and the communication with other parties we named our team Owl, because an owl can see in the dark with a razor sharp 360 degrees vision.

All three teams of the Synthesis project were assigned to a company that represents a different way of point cloud acquisition approach and this company acts as a stakeholder, use-case and expert in this field. Owl was assigned to Skeye BV. Skeye is specialised in carrying out services using unmanned aircrafts. They use the knowledge of geodesy, surveying and aerial photography to supply companies with aerial surveys and inspections. Skeye carries out projects worldwide and makes extensive use of point clouds. They use cameras instead of laser scanners and are specialised in creating high-quality point clouds from imagery.

During the Kick-Off of the GSP an often occurring problem was revealed by Pieter Franken the managing director and founder of Skeye. He told us that even though point clouds offer great spatial analysis opportunities, a high quality image was preferred by their clients for close up inspections. For this reason Skeye supplies both the close up imagery and the point cloud for the client. However the client is often inexperienced with handling point clouds, which is the difficulty; the amount of pictures and the high level of detail on the image makes it confusing and unclear for the client to easily put the images in perspective to the whole structure. This experience and understanding of the use of point clouds and the related imagery offers great space of improvement.

We gratefully accept this challenge to help both Skeye and their clients by researching this issue. Our mission is to deliver a way/ framework for visualizing the acquired point clouds in a way that can satisfy Skeye' customers. In other words, Owl's mission is to bridge the gap between customer's expectations and Skeye' resulting datasets. The research question corresponding to this mission is: How can the understanding of point cloud data and the related imagery be improved for the inexperienced user. Together with Skeye we concluded that only by going through all the steps we can find a truly innovative and out of the box idea for this topic. Owl's goal is to gain the whole experience of acquiring and processing aerial data before tackling Skeye' data communication issue.

The plan is to start by acquiring a similar dataset as Skeye with equivalent quality and accuracy they achieve by using UAVs but without using one. This is due to the no-flying zone in Delft but it can also be seen as a chance to understand every element of the acquisition process. This dataset will be acquired of the Bouwbup, a side building of the BK City. From that point on we will do the processing and application in our own way and work towards a solution to our problem.

This final paper is to summarize and synthesize the findings and results. The structure is mostly based on the chronological order of our project. Every main chapter starts with a introduction. We begin with the methodology after that the three steps of the geo-information process are being treated namely; acquisition, processing, application. Every is step includes a literature review about the topic, the best candidate. an implementation plan, the results and an evaluation. After the tree steps of the process we will face our product with testing and continue with the results, an exploitation plan, system characteristics and final conclusions. We complete the report with suggestions upon further research and acknowledgements.

We hope this report shows a complete image of the project and the scientifically related subjects. In addition to fulfilling the needs of the different stakeholders Owl also hopes to contribute to the scientific community with an innovative solution concerning human interpretation of point cloud data, and the research area of visualization and dissemination, and data fusion of explorative point clouds as mentioned by Van Oosterom and Verbree (European LiDAR Mapping Forum, Amsterdam, 2014).

D Appendix Studiedag – Visie Rijkswaterstaat

Rijkswaterstaat
Ministerie van Infrastructuur en Milieu

Wet and dry point cloud acquisition and applications within RWS

D.J. ten Napel

Inhoud

- Wat doet Rijkswaterstaat
- Voorbeelden van gebruik van point clouds
  - Dieptemetingen
  - Echoscope
  - Kwaliteit asfaltverhardingen
- Toegevoegde waarde van point clouds voor RWS
- Hoe kan bruikbaarheid point clouds vergroot worden?

Missie van Rijkswaterstaat

Rijkswaterstaat werkt aan:

- bescherming tegen overstromingen
- voldoende en schoon water
- vlot en veilig verkeer over weg en water
- betrouwbare en bruikbare informatie
- een duurzame leefomgeving

Rijkswaterstaat geeft dagelijks vorm aan die beheertaak in drie maatschappelijke rollen: publieksgerichte netwerkmanager, toonaangevende projectmanager en slagvaardige crisismanager.

Beheergebied Rijkswaterstaat

Rijkswaterstaat beheert drie rijksinfrastructuurnetwerken

Hoofdwegen

Hoofdvaarwegen

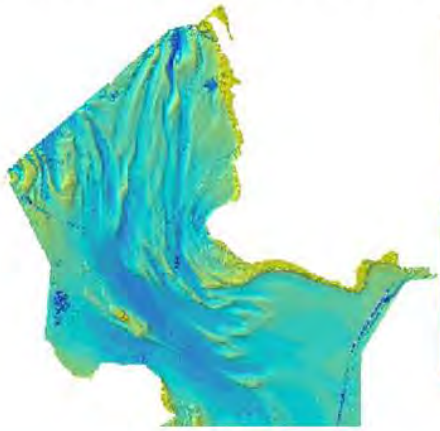
Hoofdwatersystemen

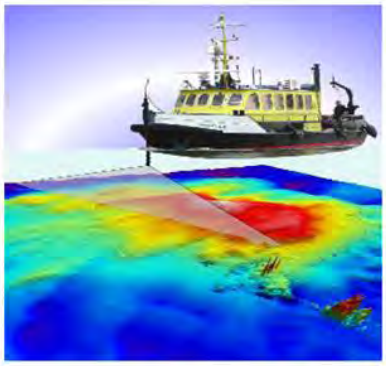
Informatievoorziening

- Dagelijks worden grote hoeveelheden data ingewonnen
  - Waterstanden
  - Waterkwaliteit
  - Waterdiepte
  - Verkeersintensiteiten
  - Kwaliteit assets (kunstwerken, asfalt)
- Deze data wordt ingewonnen door RWS en ook door marktpartijen in opdracht van RWS
  - Deels in de vorm van point clouds
  - Data is veelal beschikbaar als open data

Voorbeelden point clouds bij RWS: dieptemetingen

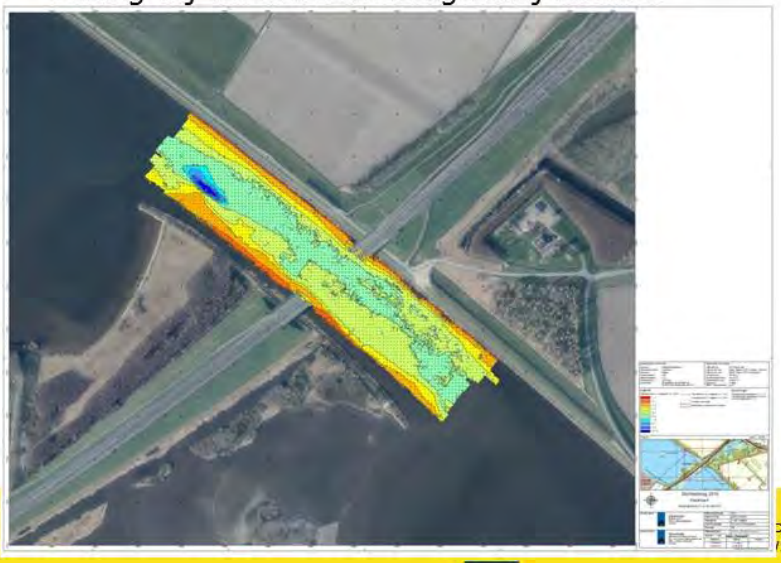
- Dieptemetingen
  - Diepte vaarwegen en watersystemen wordt periodiek gemeten





Dieptemetingen

- Voorbeeld:
  - Peiling bij de Stichtse brug nabij Almere



Dieptemetingen

- Meting met multibeam levert puntenbestand op (bijv. 200 waarnemingen per m²)
- Puntenbestand wordt veelal omgezet naar grid (1x1 meter)
- Hieruit worden dieptelijnen gegenereerd
- Voordeel grid: bestanden zijn kleiner en hanteerbaarder dan puntenbestand
- Nadeel grid: er gaat informatie verloren. Dit is nadelig op plaatsen met veel variatie in de bodemhoogte

Dieptemetingen

- In een aantal gevallen wordt het *punten*bestand gebruikt

Wrak **Stortemelk**

Voorbeelden point clouds bij RWS: echoscope

- Echoscope
  - Coda Octopus Echoscope
  - Echoscope is een "high resolution 3D imaging sonar" voor inspectiedoeleinden
  - 16.384 datapunten per 'ping'
  - 12 ping's per seconde
  - Minder schaduwwerking dan bij sonar of multibeam
  - Geometrische nauwkeurigheid lager dan bij andere systemen. Bij nieuwere uitvoeringen verbetert dit.

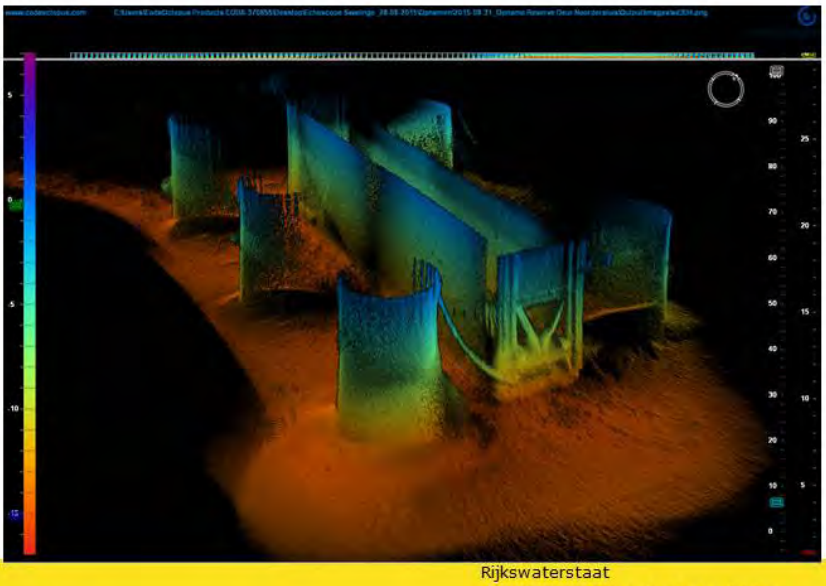
Echoscope: meetopstelling

Meetopstelling aan boord
van het schip "de Amasus"

Echoscope sonar Head

Bovenaanzicht

Echoscope: inspectie reservesluisdeur



Voorbeelden point clouds bij RWS: verhardingen

- Voor goede meerjarenplanning van onderhoud van verhardingen zijn betrouwbare meetgegevens nodig
- In het verleden waren dit visuele inspecties:
  - Objectiviteit onvoldoende
  - Gevaarlijk en afsluitingen (delen van) wegen noodzakelijk
  - Tijdrovend en duur

Voorbeelden point clouds bij RWS: verhardingen

- Huidige methode (operationeel sinds 2012)
  - Automatische inspectie met LCMS-systeem

LCMS system Source: www.pavemetrics.com

Meting kwaliteit asfaltverhardingen met LCMS

- Veilig en snel: voertuig rijdt mee in verkeer
- Goede reproduceerbaarheid
- Vervangt meeste visuele inspecties
- Geeft voorspelling planjaar onderhoud
- Accurate classificatie van type verharding is mogelijk
  - ZOAB (Zeer Open Asfaltbeton)
  - DAB (Dicht Asfaltbeton)
  - ..

Toegevoegde waarde van point clouds voor RWS

- Veel informatie
- Geven in een aantal situaties meer inzicht dan andere dataformaten (bijv. grids)
- Wanneer point clouds direct gebruikt kunnen worden (bijvoorbeeld in een viewer) zijn conversies en bewerkingen niet nodig. Dit scheelt tijd.

Hoe kan bruikbaarheid point clouds vergroot worden?

- Point clouds worden nu vooral gebruikt door specialisten
  - Datasets zijn groot
  - Kennis van software en bewerkingsmethoden is vereist
- Wensen:
  - Betere toegankelijkheid van datasets
  - Software-omgeving die goed bruikbaar is voor medewerkers (niet-specialisten)
  - Applicaties die toepasbaar zijn in meerdere werkprocessen
  - Goede beschrijvingen van methoden om point clouds te kunnen gebruiken
  - Welke andere toepassingen zijn mogelijk?? Ideeën en nieuwe producten zijn welkom

Met dank aan:

- Simon Bicknese
- Gert Jan van den Brenk
- Wim van Ooijen
- John van Schie

E Appendix Studiedag – Insight explorative point clouds

Insight through explorative point clouds Connecting indoor and outdoor

Geomatics Synthesis Project 2015-Q1

Edward Verbree
Delft University of Technology

Management of Massive Point Cloud Data
8 December 2015

TU Delft
Technische Universiteit Delft

Acknowledgements

- Students MSc Geomatics
  - Pointless

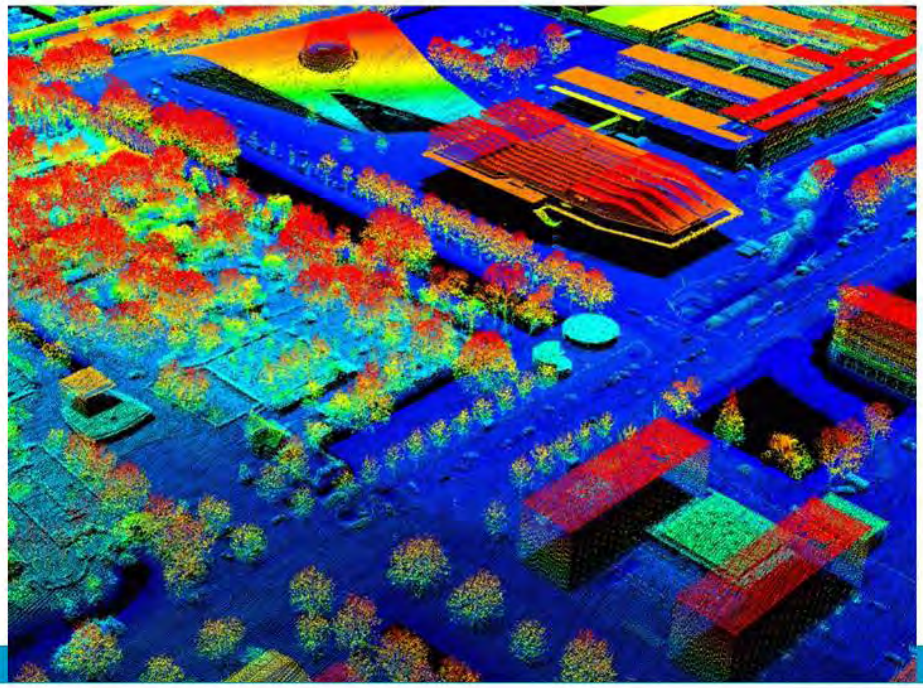
  Erik Heeres; Ivo de Liefde; Olivier Rodenberg; Florian Fichtner; Tom Broersen
  - XYZ

  Adrie Rovers; Irene de Vreede; Merwin Rook; Stella Psomadaki; Tim Nagelkerke
  - OWL

  Matthijs Kastelijns; Pieter Soffers; Jan ten Kate; Martijn Koopman; Anna-Maria Ntarladima

TU Delft

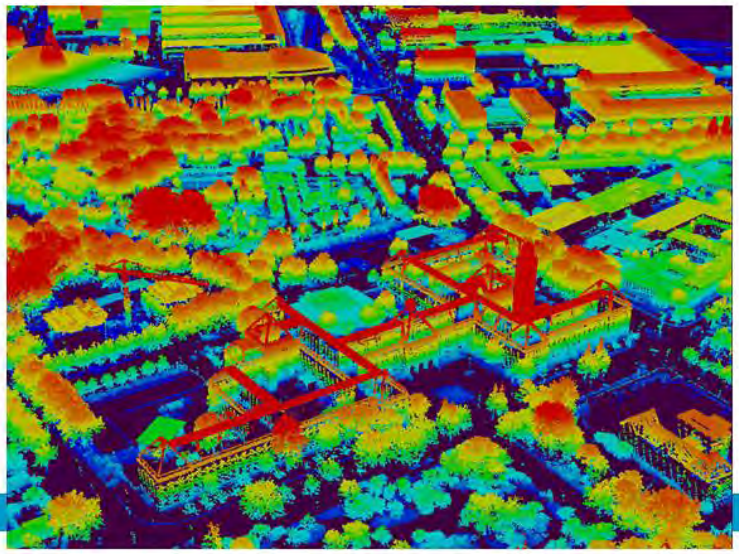
Technology that shapes our society



3

TU Delft

Key-Example: AHN2
'Actual' Height Model of The Netherlands



TU Delft

Modelling the World in 3D: Processing takes throughput time

- Example: 3D TOP10NL of the Netherlands
  - 650.000.000.000 AHN2 height points (2008-2011)
  - 15.000.000 TOP10NL objects (2012-2014)
- Fast processing by some super computing power
  - Not an Actual 3D Model
  - Semantically processed and enriched AHN2 point cloud
    - (Sander Oude Elberink)

TU Delft

Potential of Point Cloud Data

The world is its own best model.
It is always exactly up to date.
It always has every detail there is to be known.
The trick is to sense it appropriately and often enough.

- Rodney Brooks, 1990, Elephants don't play chess

TU Delft

Potential of Point Cloud Data

- Decision making processes
- Many expert-users from different professions
- Strong urge to access the original measured data
- Interactive visualisation tools
- Visual interaction

7

TU Delft

"Every time I walked through I found something I have not seen before."

- Curator David Rooney
- Science Museum - The Shipping Galleries 3D Model

8

http://youtu.be/gDTbFhFZl9I

TU Delft

Science Museum

Science Museum

Explorative Point Clouds
Research questions

- How do we store and transmit point cloud data to non-specialist users;
- How can we provide interactive visualisation tools, such that point clouds appear for the society (end users) as a virtual world, rather than technical sensor output;
- How can we provide point cloud analysis tools, such as object reconstruction, profiling and change detection;
- What kind of explorative spatial data analysis tools should be available to support domain-experts in extracting semantic information from the point clouds attribute values.

11

TU Delft

Vision: Explorative Point Clouds
Point Clouds for immediate use and analysis

- Direct Human Interpretation
- Visualisation tools and techniques: Smart Mapping / Cartography
- Interactive cartography and geo-visualisation: interaction and interfaces

- Direct Tools to identify, measure, process, and analyse
- Visibility analytics: line of sight
- Measurement tools: length, area, and volume calculations
- Analysis tools: path-planning
- Explorative Spatial Data Analysis tools: histograms, distributions, trends

12

TU Delft

Syntheses project
Goals (8 weeks)

- A synopsis of the different disciplines
- Practice teamwork in small groups
- Experience the **entire geo-information process**
  - From project definition, over measurement (data gathering), data processing and analysis, to presentation and delivery, and application.
- Encourage collaboration between students and staff of the groups involved

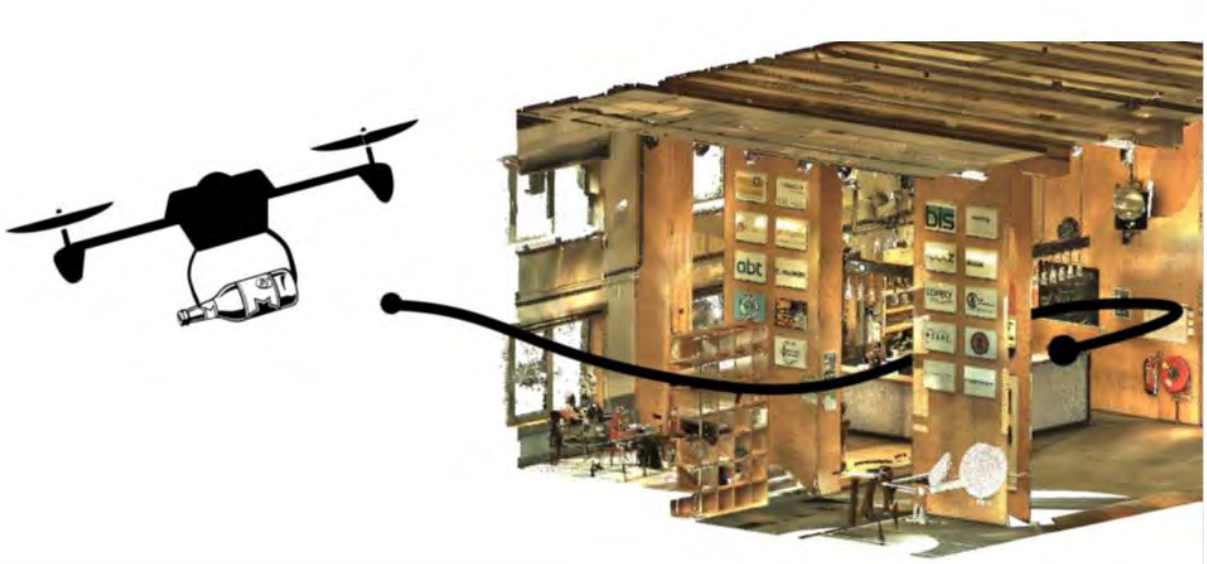
TU Delft

Syntheses Project
Point cloud key-examples

- Pointless
  - Terrestrial / Indoor: Leap3D (Zeb1)
  - Processed directly to obtain navigable space
- XYZ
  - Mobile: Cyclomedia
  - Processed directly to segment and classify objects
- OWL
  - Airborne / Drone: Skeye
  - Processed directly to link dense matched points to images

TU Delft

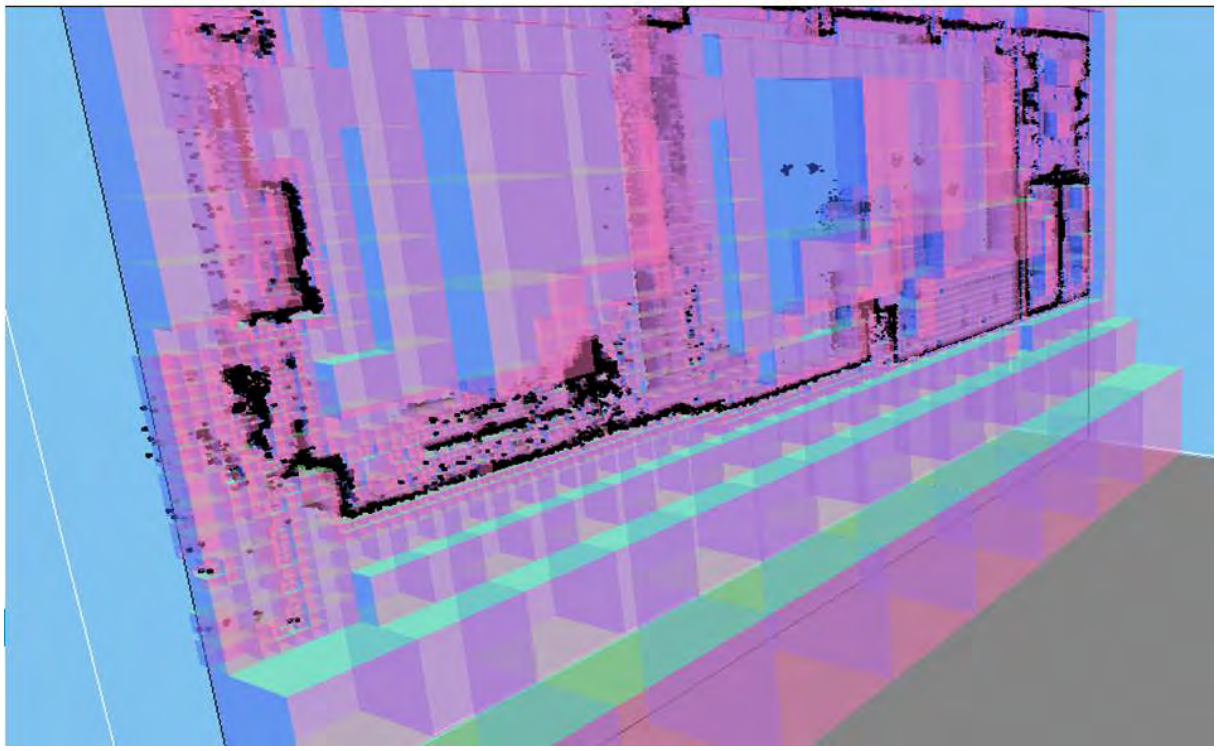
Pointless



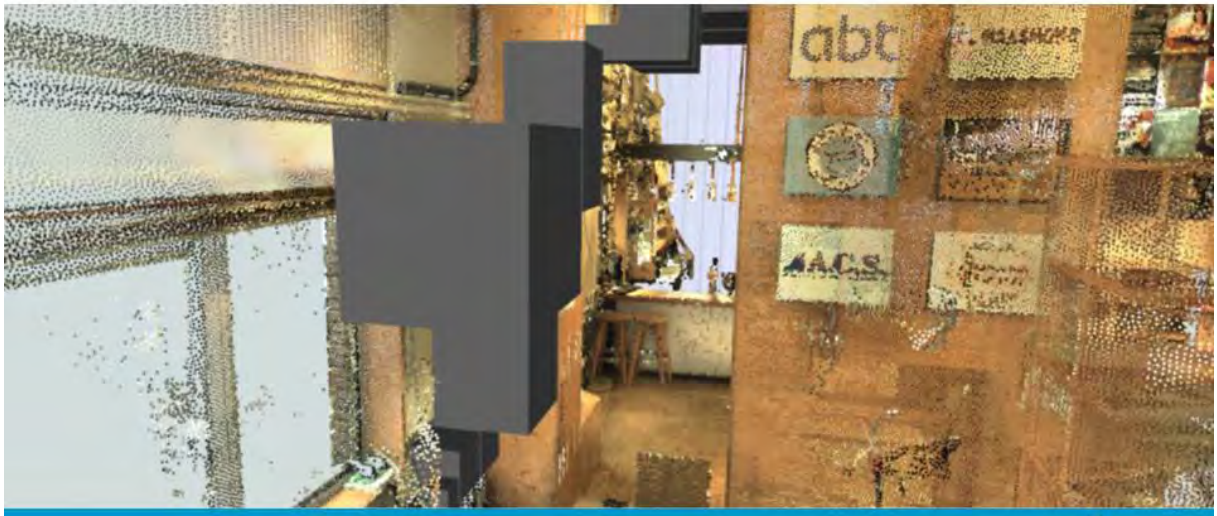
15

TU Delft

Pointless



Pointless



17

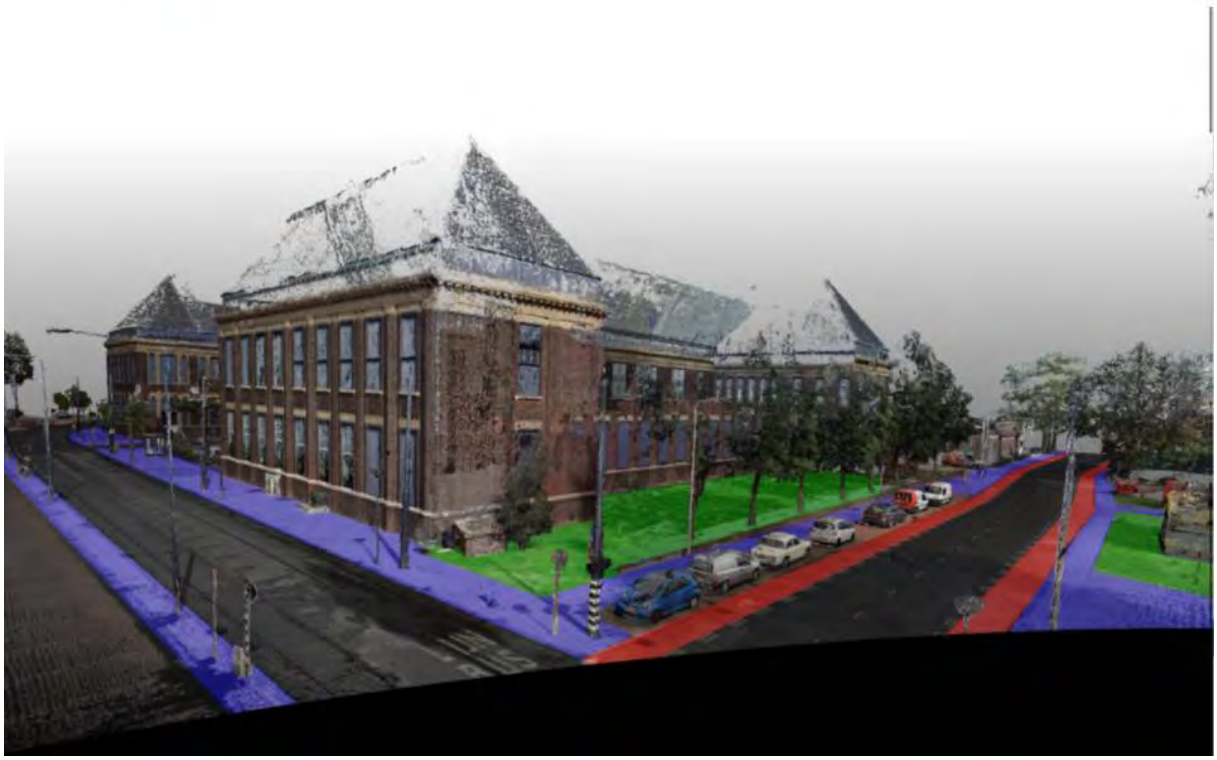
TU Delft

XYZ

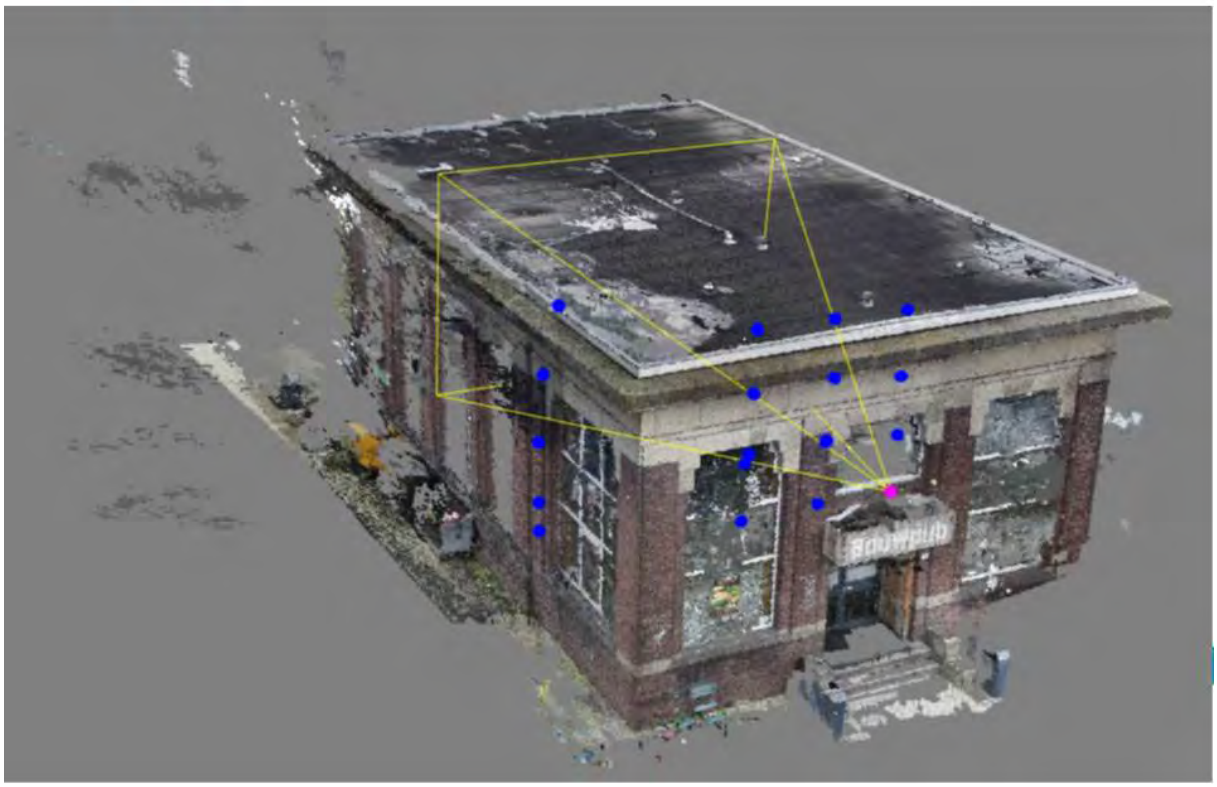
18

TU Delft

XYZ



OWL



Conclusions
Modelling the World

- Point Clouds have more value than the (derived) models
  1. Point Clouds keeps the details
  2. Added value to be revealed by user
- Connecting data acquisition and information retrieval:
  - The world (read: Point Cloud) is its own best model.
  - It is always exactly up to date.
  - It always has every detail there is to be known.
  - The trick is to sense it appropriately and often enough.

TU Delft

OTB – Onderzoek voor de gebouwde omgeving
Faculteit Bouwkunde, TU Delft
Julianalaan 134, 2628 BL Delft
Postbus 5043, 2600 GA Delft

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