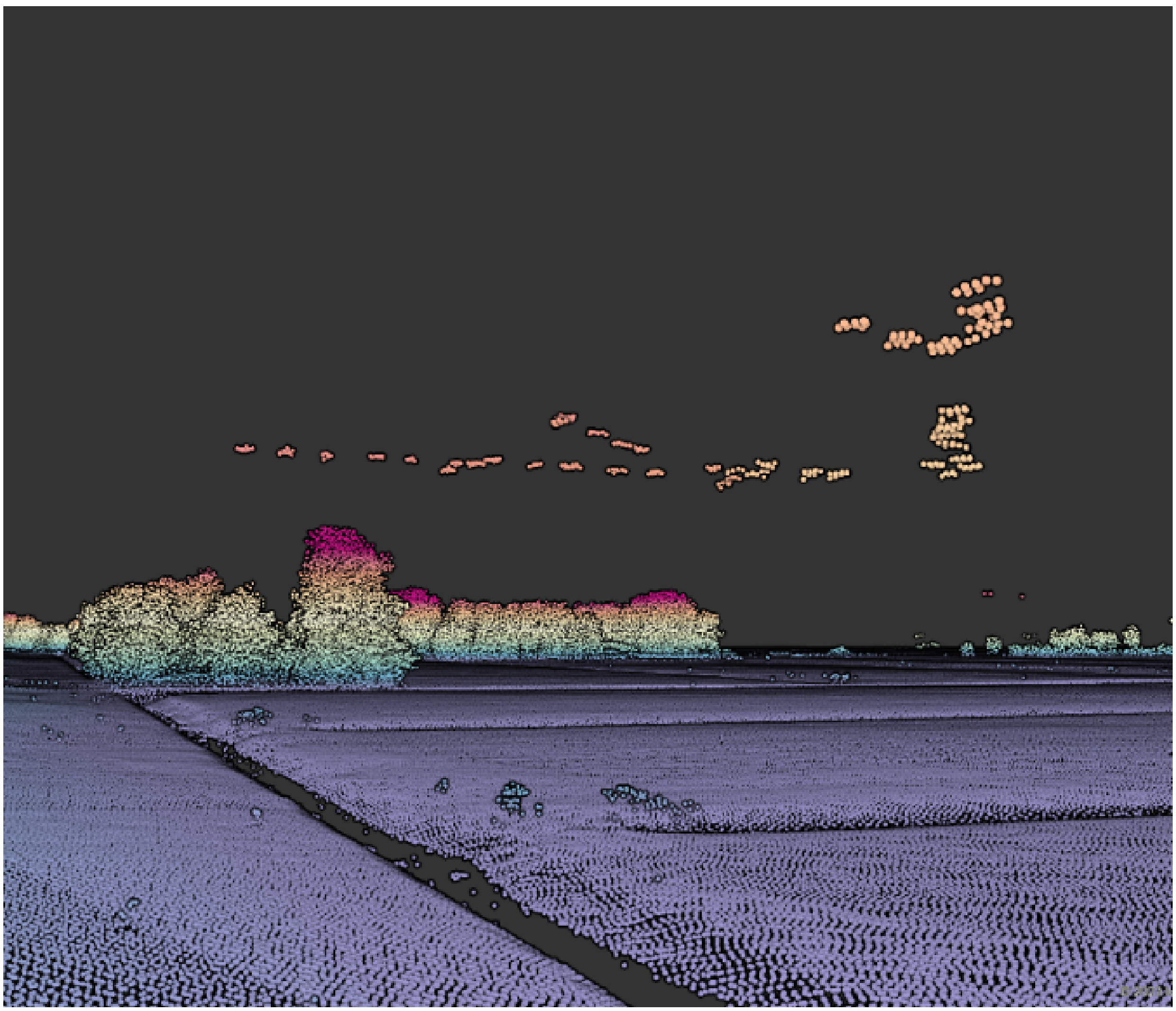
Internship Report Geomatics for the Built Environment

Automatic Misclassifications Identification of Semantically Segmented AHN5 Point Cloud

Sharath Chandra Madanu

January 2024

TU Delft Delft University of Technology



The cover illustration depicts detection of birds in airborne LiDAR scanned AHN4 point cloud data.

Internship Report Geomatics for the Built Environment

Automatic Misclassifications Identification of Semantically Segmented AHN5 Point Cloud

Sharath Chandra Madanu

January 2024

Geomatics Supervisor: Shenglan Du

Company: Geodelta

Internship Supervisor: Bob Valten

This report forms the written record of an internship carried out in the period 01.07.2023 - 31.12.2023. The internship is an elective of the MSc Geomatics for the Built Environment, Delft University of Technology, awarded with 10 ECTS.

Abstract

The project is undertaken during an internship at Geodelta, and involves developing algorithms to identify errors in the semantically segmented point cloud data of the Netherlands' Actueel Hoogtebestand Nederland (AHN). The report details the methodologies used, including the abstraction of 3D point cloud to 2D grid layers, because it is difficult to handle the data as it is owing to its irregular data structure. Further report discusses the algorithms developed for identifying various types of misclassifications and inconsistencies. For small-scale inconsistencies which are not previously observed in AHN4, new algorithms are developed called *Mixed-Classification* detectors. The report also includes findings and reflections on the impact of scanning technology, flight strip overlap, and production tile inconsistencies which are difficult to identify automatically. The first phase of the project is now online at: link here.

1 Introduction

1.1 Description of the company

Geodelta is a renowned engineering consulting firm, specializing in geodesy, photogrammetry, and laser scanning. The firm prides itself on providing technical advice and developing user-friendly software, with a keen focus on precision and reliability in spatial measurements. Their range of services is extensive, encompassing data management, quality control, education, process optimization, and calibration. This diverse portfolio caters to a wide array of clients, including state governments and local specialists, all of whom demand the highest standards in geometrical quality.

Geodelta has developed a variety of specialized software tools. These include Eyebase, a stereo mapping tool with a user friendly interface, fully optimised for mapping and updating large scale basemaps; Raida, designed to turn images into point cloud or ortho photos; and Omnibase, an all-in-one 3D environment software for viewing and handling point clouds, panoramic images, nadir and oblique images. Also noteworthy are MODUPS, a software for photogrammetry based deformation analysis, and ASSETHUB, created in collaboration with TNO, which focuses on crack detection in brickwork through image analysis.

Geodelta's expertise extends to quality control services, particularly for projects involving aerial images, as well as LiDAR and drone survey data. These services are crucial for organizations that depend on aerial imagery for maintaining large-scale topographic registrations, such as the Basisregistratie Grootschalige Topografie (BGT) in the Netherlands. Furthermore, Geodelta plays a pivotal role in the quality control of the nationwide height model, encompassing both AHN4 and AHN5, underscoring their critical contribution to national geospatial projects.

1.2 Description of the project

1.2.1 Background of Project

Actueel Hoogtebestand Nederland (AHN) is the height model of the Netherlands. It is a digital dataset that provides information about the elevation and topography of the Dutch landscape. AHN has made various datasets available, such as point clouds in LAZ format, and raster images in GeoTIFF format for Digital Terrain Model (DTM) and Digital Surface Model (DSM) at resolutions of 0.5 m and 5.0 m. It is widely used across industries for various projects. As a considerable portion of the land lies below sea level, AHN plays a crucial role in flood risk management and water resource planning. Classified point cloud has utility in various fields: such as urban planning applications need information on buildings and manmade ground structures (bridges, canals, etc.); forest monitoring needs trees structure; autonomous driving uses object detection and segmentation ([5]). Currently, four editions of this data have been made publicly available. The latest, AHN5, is in the stages of data collection and processing, poised to further enhance the model's applications and accuracy.

The collection and classification of the AHN5 data is done by the organisation Miramap Aerial Surveys. Flights are flown in the northern part of The Netherlands and the airborne lidar data is collected as the first part of the whole project. The flight paths and dates on which they were flown are depicted in Figure 23. The first phase of the project, the data collection phase, started in mid-December 2022 and went on till the end of February 2023.

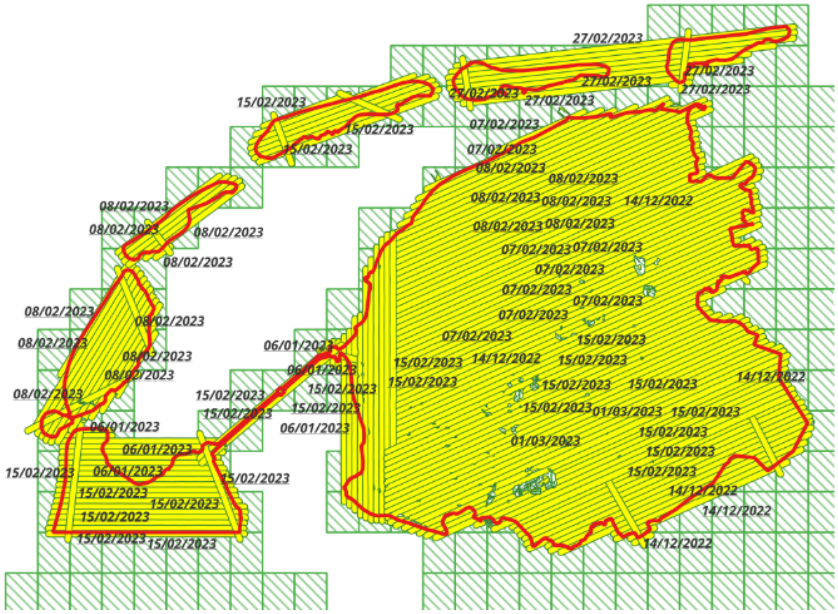
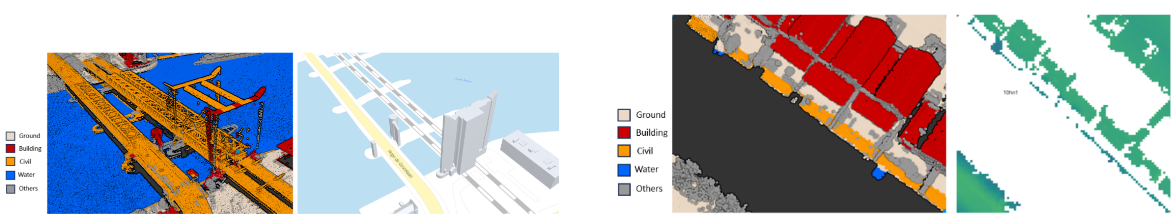


Figure 1: AHN5 flight path

1.2.2 Problem Statement

The internship project addresses the critical problem of detecting misclassifications in semantically segmented AHN5 point cloud data by developing precise mathematical algorithms. As the AHN data is public data and is provided by a government organisation, so it is vital for the quality control company, Geodelta, working on this project to give all-round fitness information about the point cloud. No data is hundred percent perfect, the main task of quality control would be to identify where the data is faulty and by what degree it is faulty. Meaning, that both the quality and quantity of imperfections have to be identified and measured.

The misclassifications in segmented point cloud data can lead to errors in mapping terrain (DTM and DSM), identifying structures, and assessing the landscape's features (vegetation, water bodies). Misclassified buildings can also show up in 3D BAG (Figure 2a). The ultimate goal is to automate and streamline the quality control process, ensuring no misclassification goes unnoticed, and making sure all detected errors are within the specified limits (as discussed in Appendix-A).



(a) Misclassified AHN4 and its reflection in 3D BAG (b) Misclassified *ground* and its reflection in DTM

Figure 2: Point cloud misclassification and its implications

1.3 Data Handling

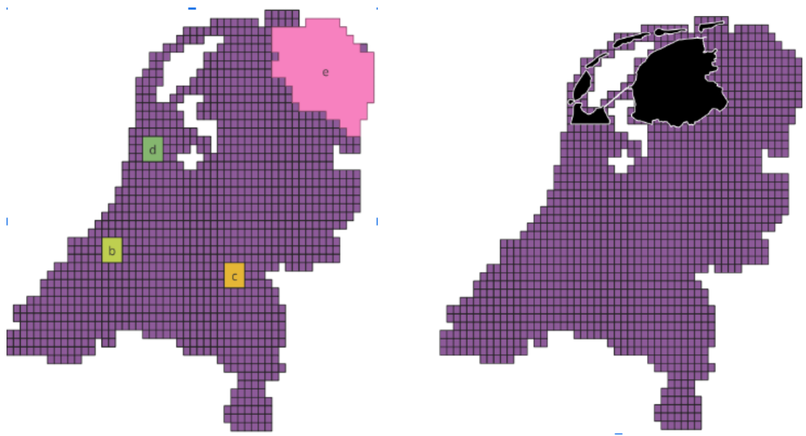
Point cloud data, being inherently three-dimensional and unstructured, poses significant challenges for processing and analysis. Point clouds, unlike images, are 3D data structures, making standard image processing techniques inapplicable [4]. One common approach to manage this complexity is through voxelization, where the point cloud is converted into a regular grid of voxels (3D pixels), simplifying the data structure at the cost of some detail and resolution. Another method is projecting the point cloud onto different planes, such as the XY plane, and processing the projected data [1]. This approach, while simplifying the data for analysis, can lead to loss of valuable three-dimensional information. However, it offers advantages in terms of processing speed and ease of understanding. These abstraction techniques are very commonly used with machine learning algorithms to semantically segment point cloud data ([5]). Recent research in the field has focused on developing more efficient processing methods for point clouds. For instance, dynamic voxelization ([3]) is a technique that allows for more flexible and efficient processing of point clouds by dynamically adjusting the voxel grid based on the data's needs. This method can extract features more effectively while maintaining the advantages of a structured data format.

In this project, the point cloud data is projected onto a gridded XY plane. Each grid cell captures information about points within its XY boundaries. This method is highly effective for most detection tasks, but it presents challenges in certain scenarios. The Chapters 2 and 3 elaborate on the advantages and drawbacks of this approach. The primary objective of the internship is to develop efficient mathematical algorithms for identifying misclassifications in the data. Targets detected by these algorithms are subsequently manually reviewed to ascertain if they are indeed misclassified or falsely identified (false positives) as errors. The ultimate goal is to refine the algorithm to minimize the need for manual verification of these false positives, thereby reducing labour-intensive tasks.

2 Methodology

2.1 Data Used

Geodelata had previously been involved in ensuring the quality of AHN4 data, creating algorithms to detect misclassifications. Due to delays from Mirmap Aerial Surveys, the initial month of the internship focused on understanding these existing algorithms and software. This period involved testing the algorithms on four AHN4 data batches (Figure 3a), providing practical insights into their performance. This training enhanced agility and productivity, which was beneficial when working with the newly received AHN5 data. The first batch of classified point cloud data for quality control pertained to the Northwest region of the Netherlands (Figire 3b).



(a) Four batches of AHN4 training data

(b) First semantically labelled delivery data of AHN5

Figure 3: AHN point cloud tiles used for internhip

Additionally, Basisregistratie Adressen en Gebouwen (BAG) and BGT datasets are also used during manual checking, when in doubt of whether the given label is valid or not.

2.2 Equipment and Software Used

Miramap Aerial Surveys used Leica CityMapper - 2 High-performance Urban Mapping Sensor for collecting the data. Their responsibilities also included classifying the point cloud. This process involved calibrating the data to address any system-specific inaccuracies, geo-referencing to ensure the LiDAR data aligns with precise geographic coordinates, and removing noise, outliers, and artifacts from the raw data to enhance its quality. The final step in their process was to categorize the points into specific classes.

At Geodelta, the software used for viewing and handling the point cloud is *Omnibase*, a browser-based software developed by Geodelta, which is faster than any freely available software. Omnibase is developed using Potree technology, an open-source WebGL-based point cloud renderer for large point clouds. To use a web-based viewer, first, the LAZ/LAS data has to be converted to octree LOD structure using PotreeConverter ([2]). The preprocessing to convert the point cloud to octree data structures is time-consuming, but it is worth it to make viewing and handling of the point cloud smooth and fast. Omnibase has features like different standard point sizes, adaptable point sizes based on neighbourhood point density; view by intensity, flight strip numbers; turning on/off different data layers; and measurement tools (Figure 4).

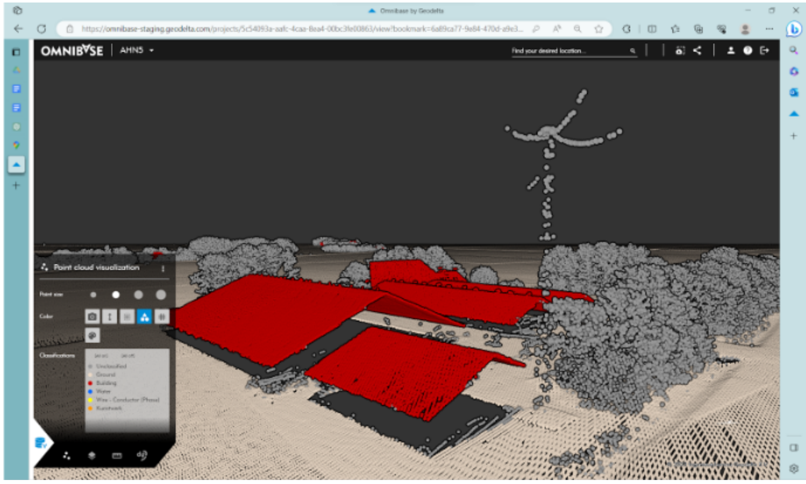


Figure 4: Omnibase, a web-based point cloud viewer

CheckMate, an in-house software built on top of Omnibase, was used for manual error handling. In CheckMate targets can be rejected, accepted, modified, sorted based on multiple parameters (serial number, error type, description of error, and area) etc. (Figure 5)

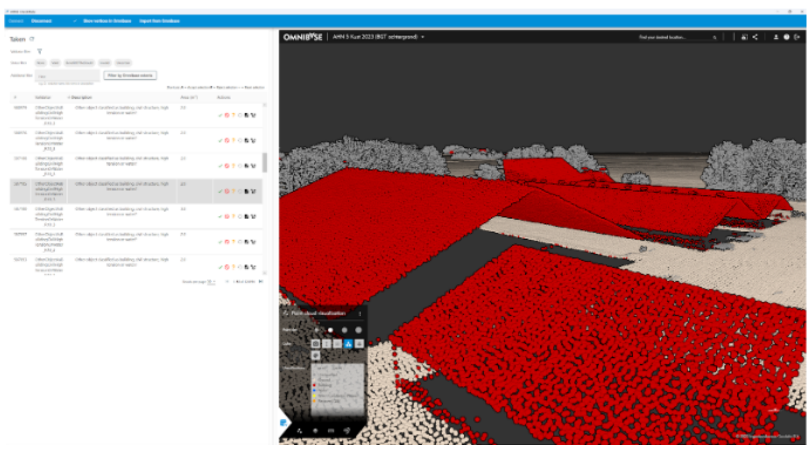


Figure 5: CheckMate, manual error handling software

And finally, QGIS was also used in parallel. QGIS is used for viewing 2D data like aerial photos, BAG buildings, and multiple BGT layers (water, high tension lines, bridges and jetties, house boats, etc). Further, the all the detection algorithms are developed to be in 2D, so the potential misclaffied targets are also be visualized on QGIS.

2.3 Abstraction of 3D Point Cloud to 2D Grid Layers

The point cloud at such a huge scale, at the country level, is humongously large (several terabytes) in size and any computation on it is very intensive and takes a lots of time. Point clouds are inherently irregular and unstructured, which pose significant challenges in efficient data exploitation and feature extraction ([6]). So, to handle this efficiently the errors in 3D classification are reduced to problems of 2D, so the processing is fast.

Firstly, the project area is tiled into grid cells of 1m x 1m. The implemented data structure is made of multiple grid layers, and each grid layer is exclusive to a particular segmentation class (1, 2, 6, 9, 14, 26), for illustration refer Figure 6. Each layer is composed of cells of dimension 1m x 1m; and each cell stores the maximum and minimum height of the points within the cell of the corresponding class. If both the maximum and minimum heights are not available then it implies the cell has no points of that class. Further, if both maximum and minimum height are same to the last decimal point, then it could be inferred that there is only one point of that class in a cell. Most importantly, the data structure is not built to store the number of points that are actually present within the cell.

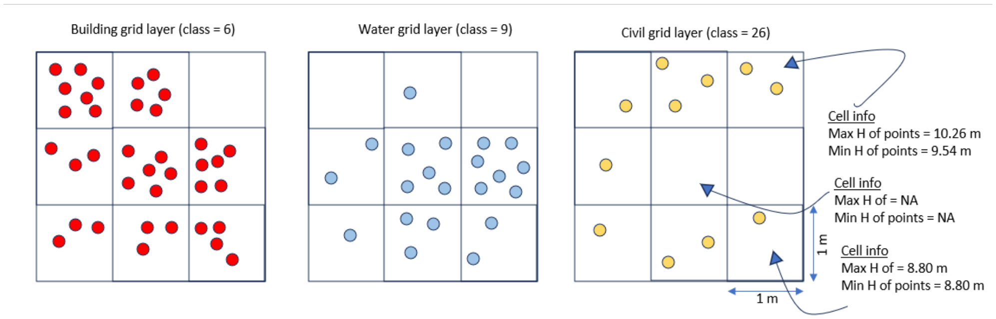


Figure 6: Point cloud abstracted to 2D grid layers

2.4 Developed Algorithms

The detection algorithms could be categorized into two parts. One category is for the main detection checks and these errors must be within the specified limits. The errors that come under this category are from R1 to R10, *outliers*, *Ground in Building*, and *Houseboats/ships misclassification* (refer Appendix A). I'm calling it *Standard Checks*. The second is for detecting inconsistencies, referred to by me as *Inconsistency Checks*.

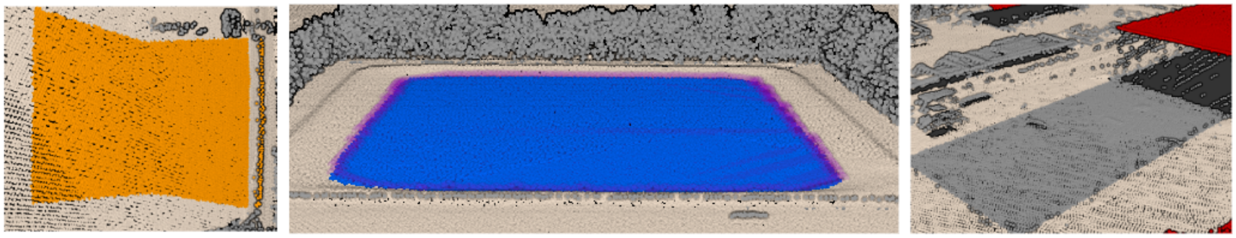
2.5 Standard Checks

For all of the errors that fall in this category, a separate algorithm is developed. Nevertheless, both R9 and R10 are further classified into separate errors within themselves (R9_1, R9_2, R9_3, R10_1, R10_2 and R10_3). This is because the *other* classified points are made of low vegetation (<0.5 m), high vegetation (>0.5 m) and other objects, and they all have to be treated differently.

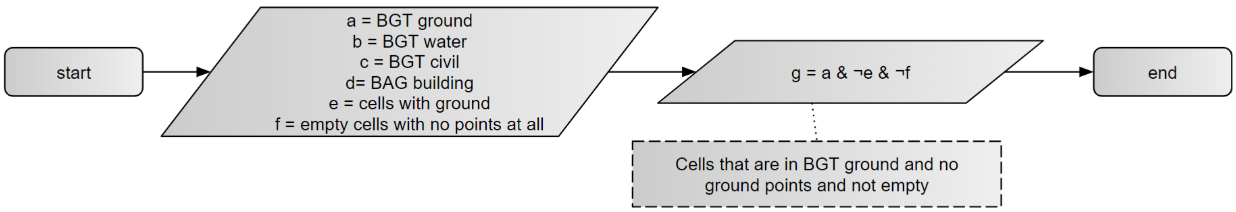
Depending on the error check BGT, BAG layers or AHN4 point cloud are used as a base reference. BAG provides the data of buildings and residences in 2D vector maps. BGT has well well-maintained 2D dataset of buildings, roads, watercourses, terrains, vegetation, and railway lines. In short, BGT is the design of the physical environment. For the standard checks, these three datasets are the core.

From R1 to R10 the algorithms and few detections are presented in Figures from 7 to 15. If the algorithm says 'c = *cells with ground*', it means that c is a variable with a 2D AHN5 ground layer which is abstracted (projected). Further, '$\neg c$' means cells with no ground points at all.

R2 check, which detects all manmade objects classified as ground, gives lots of detections (targets/candidates), which should be manually checked again. For the current delivery of AHN5 more than 10,000 targets. This is because all manmade structures: buildings, jetties, bridges, and high-tension have to be checked, with multiple reference datasets. And a lot of them are false positives.

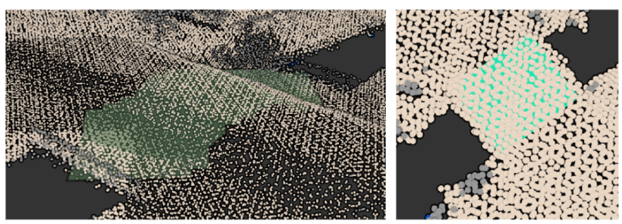


(a) R1 detected errors

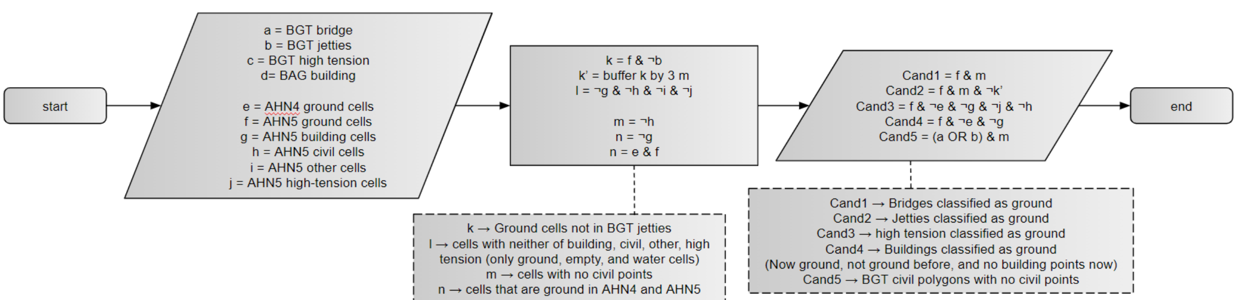


(b) R1 detection algorithm

Figure 7: R1 - Ground not labelled as ground

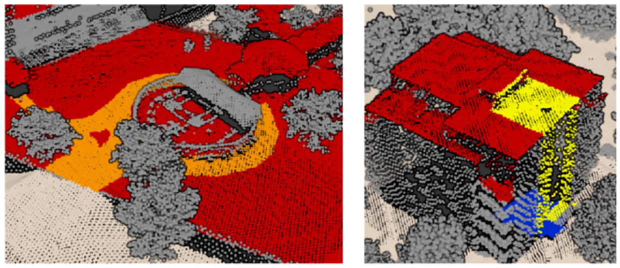


(a) R2 detected errors (bridges classified as ground)

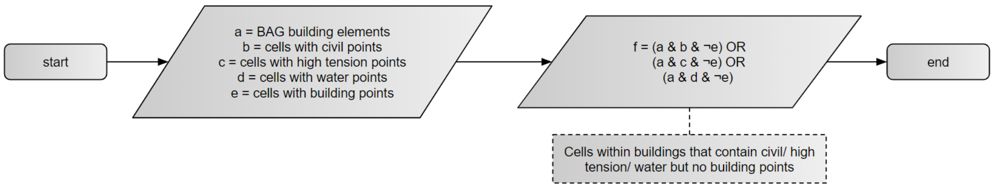


(b) R2 detection algorithm

Figure 8: R2 - Man-made objects as ground

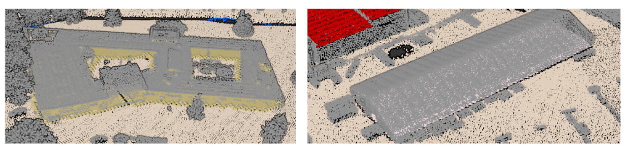


(a) R3 detected errors (building as civil (left) & building as high tension (right))

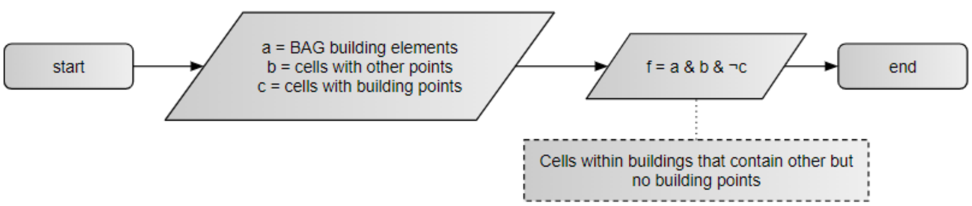


(b) R3 detection algorithm

Figure 9: R3 - Building as Civil/High-tension/Water

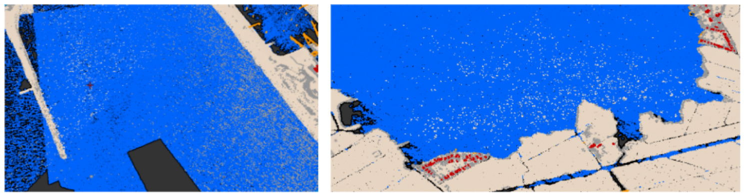


(a) R4 detected errors

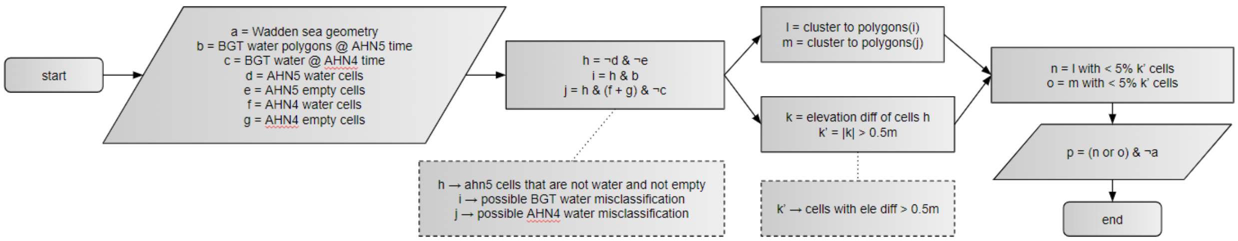


(b) R4 detection algorithm

Figure 10: R4 - Building labelled as Others

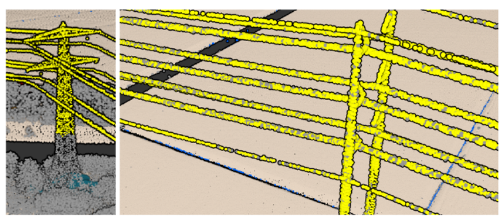


(a) R5 detected errors

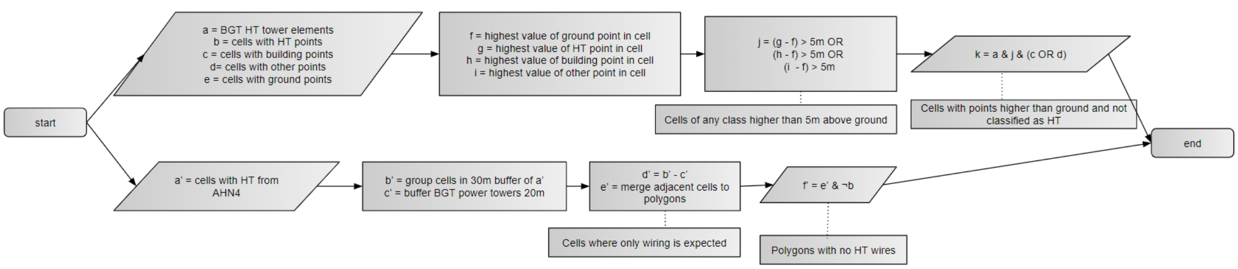


(b) R5 detection algorithm

Figure 11: R5 - Water not as water

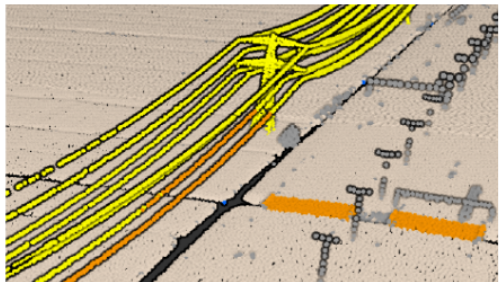


(a) R6 detected errors

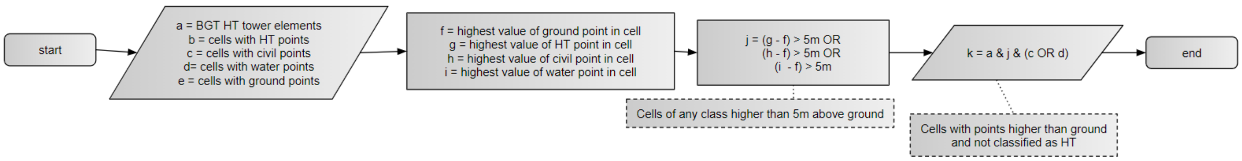


(b) R6 detection algorithm

Figure 12: R6 - High tension as Building/Others

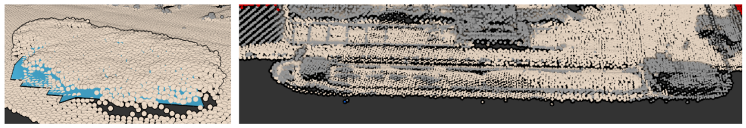


(a) R7 detected errors

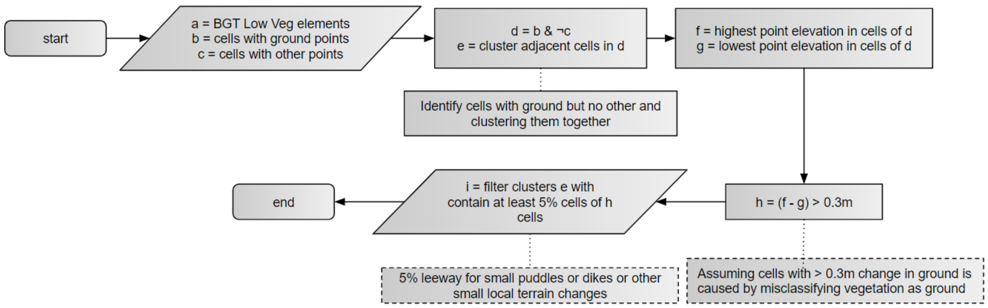


(b) R7 detection algorithm

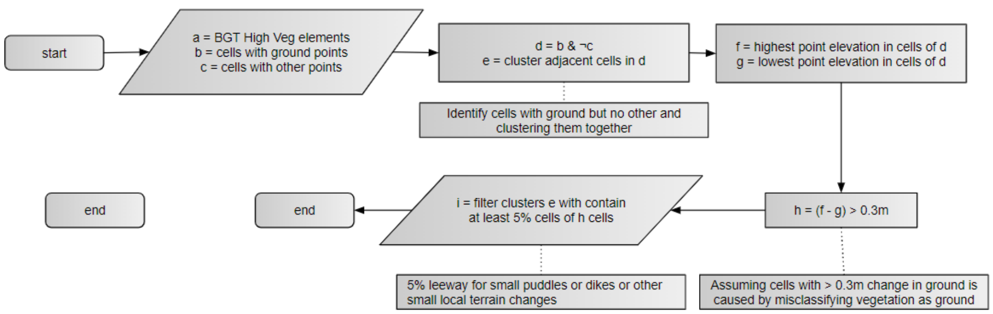
Figure 13: R7 - High tension as Water/Civil



(a) R9 detected errors - (low vegetation as ground (left) & ship as ground (right))

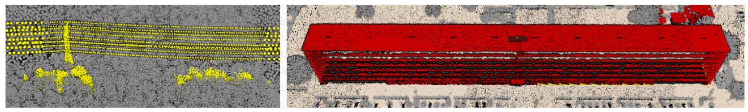


(b) R9_1 detection algorithm - to detect low vegetation labelled as ground

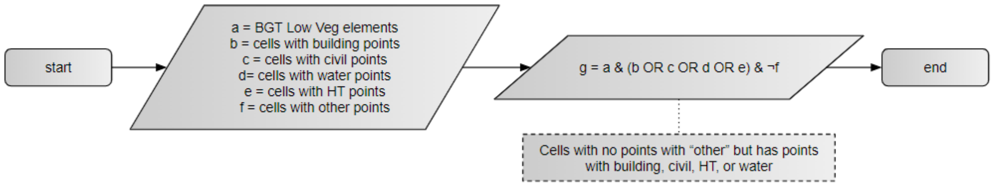


(c) R9_2 detection algorithm - to detect high vegetation labelled as ground

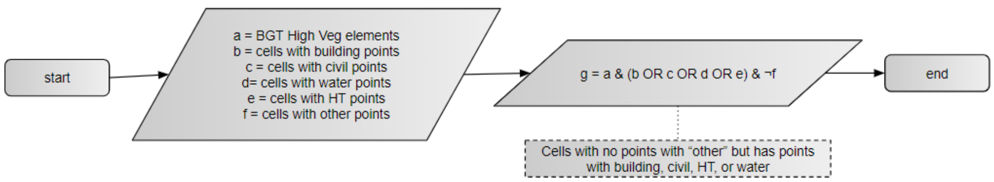
Figure 14: R9 - Others as ground



(a) R10 detected errors - (high vegetation as high tension (left) & building facade as building (right))



(b) R10_1 algorithm - to detect low vegetation labelled as Building/Water/High-tension/Civil



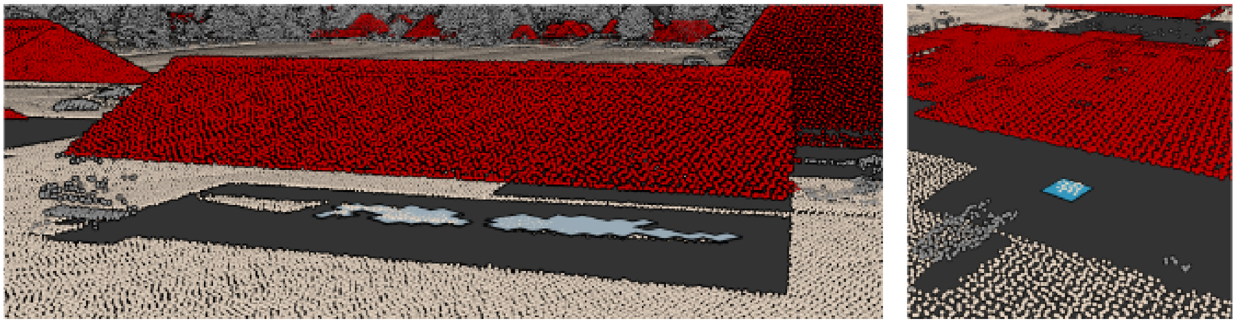
(c) R10_2 algorithm - to detect high vegetation labelled as Building/Water/High-tension/Civil

Figure 15: R10 - Others as Building/Water/High-tension/Civil

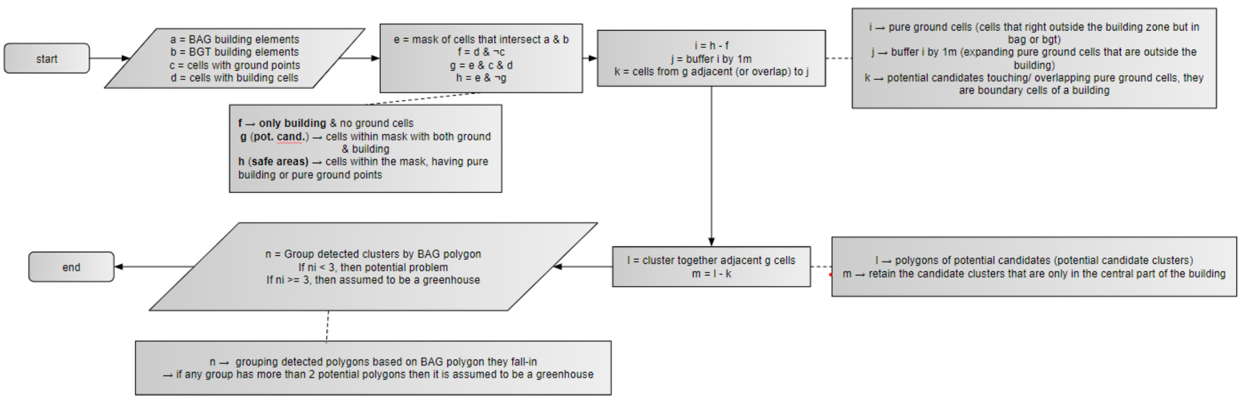
R3, R4, R6, and R7 error checks are near-perfect algorithms. For R3, almost no buildings are misclassified as civil/high-tension, it is very rare and is easily within the acceptable limits. For, R6 and R7, very rarely high-tension lines and poles are misclassified, and the detection area is also limited to the area of high-tension poles which is not big compared to the size of the whole data.

2.5.1 Ground in Building Detection

It is one of the crucial checks and very strict. One of the main functions of AHN data is to make DTM and DSM maps of the Netherlands, so it is highly important to make sure that there are no misclassified ground points within the buildings; otherwise they will reflect in the DTM, and on surface water modelling, flow happens through the buildings. Greenhouses are an exception, as long as the reflected ground points are of the same elevation as the ground outside the greenhouses. This is because neither BAG nor BGT has a greenhouse layer.



(a) *Ground in Building* detected errors

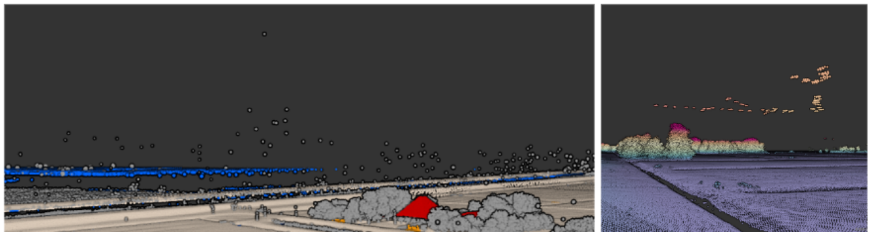


(b) *Ground in Building* detection algorithm

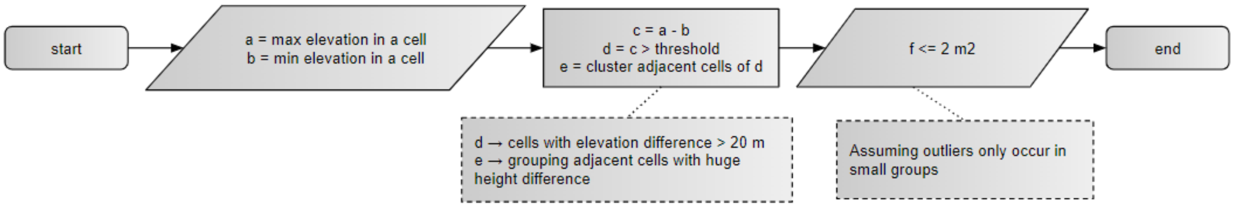
Figure 16: *Ground in Building* detection

2.5.2 Outliers

These are points that don't actually represent anything we're interested in. For example, reflections in the air due to moisture, reflections on birds (Figure 17a) and reflections below ground level. If they not cleaned properly they are reflected in the DSM map. It is assumed that outliers don't appear in groups, and the algorithm finds the range of elevation difference within a cell, and groups adjacent cells together with a range greater than 20 m. However, many buildings easily extend beyond 20 m, so to reduce the number of false positive hits, clusters with areas less than 2 square meters are checked. However, birds flying in formation very close to each other will be skipped, because their cluster is bigger than 2 m^2. Birds flying closer to the ground are also missed in detections.



(a) *Outliers* detected errors

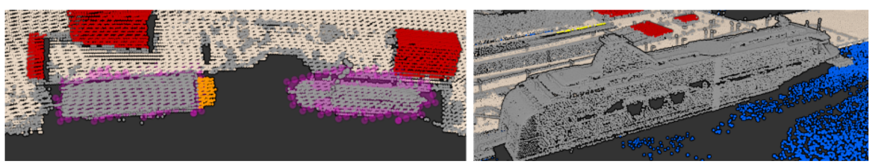


(b) *Outliers* detection algorithm

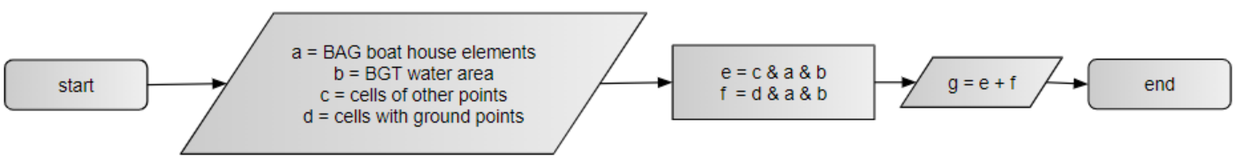
Figure 17: *Outliers* detection

2.5.3 Houseboats/ships classified not as *Buildings*

Not all the boats are used as houses are living. However, the onces which are used for that purpose are already mapped in BGT, and they should be classified as *Buildings*. Along with the roof, the entire boat/ship structure (walls, deck, etc.) has to be classified as a building. The algorithm is developed to detect only other points and ground points at the ship's location.



(a) Houseboats/ships misclassified detected errors



(b) Houseboats/ships misclassification algorithm

Figure 18: *Houseboats/ships* misclassification detection

2.6 Inconsistency Checks

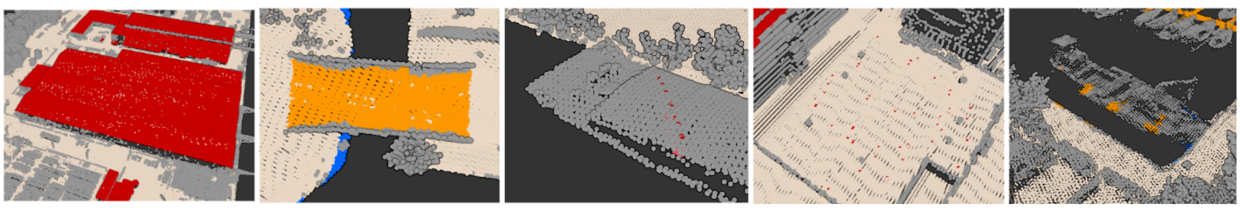
Inconsistencies are not like standard errors where a significant part of an area is entirely misclassified by a commonality. Instead, these are either small-scale inconsistencies that appear frequently in many places or just a messy overlapping classification. These errors are hard to detect and even harder to quantify.

2.6.1 Mixed Classification Detector

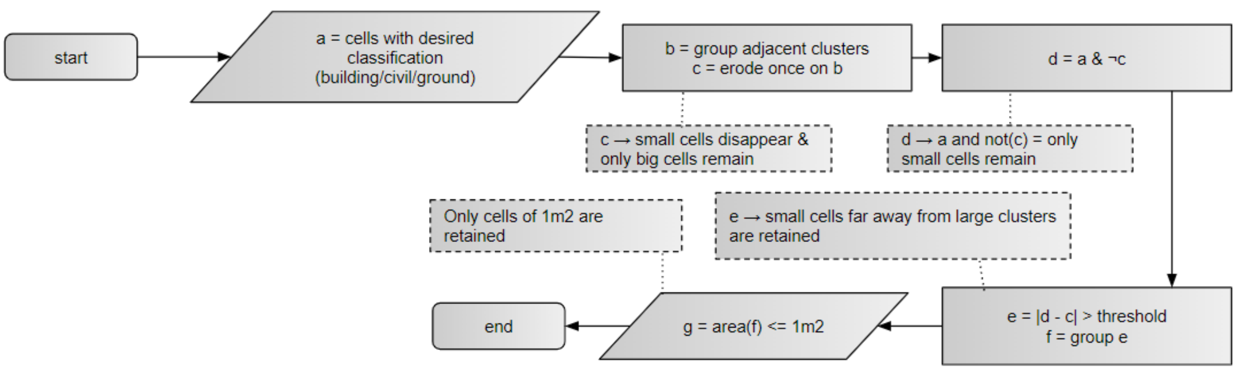
Mixed classification detector is developed looking at the data, and there is also no prescribed error classification mention in *AHN guidelines*. In a particular place, points of multiple overlapping classifications are observed in these errors (Figure 19a). Looking at the data Mixed errors are classified into five categories: *Mixed Bridges-Ground*, *Mixed Bridges-Others*, *Mixed Jetties-Ground*, *Mixed Building-Civil*, *Mixed Building-Other*. *Mixed Bridges-Ground* and *Mixed Bridges-Other* have same algorithms, just that later uses *Other* grid layer rather than *Ground* grid layer.

2.6.2 Single Cell Detector

Ground, *building*, and *civil* are sporadically present in the most unexpected of the places (Figure 20a). Building points are observed on the ground, in the canals, and on the bridges. Civil points are present on the ground, on buildings and trees. The unacceptable of the errors is misclassifying the ground points, as these errors affect the national DTM map, these ground points could be seen on the bridges, on top and in the middle of the buildings, in the canals, and also on the ships, etc. To identify these, three subdivisions of this error check are developed, i,e,.*Single Cell Building*, *Single Cell Civil*, *Single Cell Ground*.



(a) Single cell detected errors



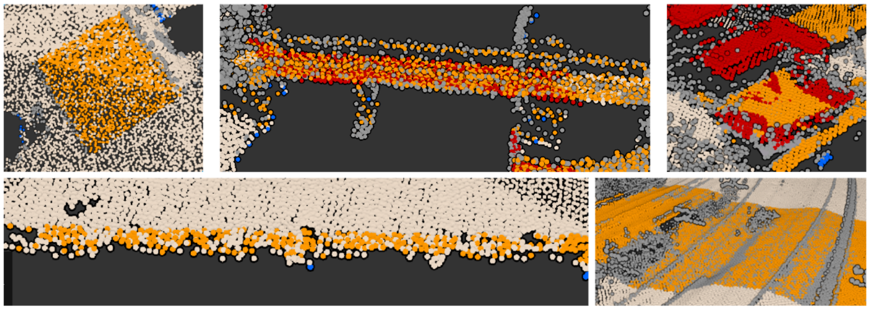
(b) Single cell detection algorithm

Figure 20: Single cell detection

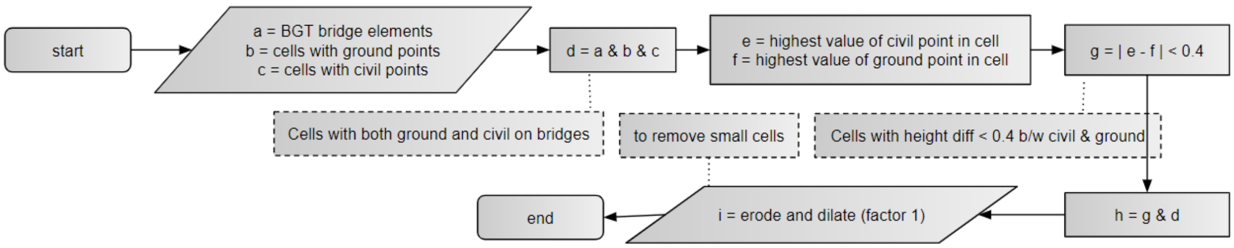
2.7 Analysis Methods

All the detected potential targets from the previous step are then accessible within *ChackMate* for manual inspection. User then can individually marks each error as *accepted*, *rejected* or *validity unknown* (to later come back for review). If the potential errors of any type are way too many, a fraction, usually $\frac{3}{4}$ or $\frac{1}{2}$, of the errors are manually checked and then general assumption is made and extrapolated to the whole error size.

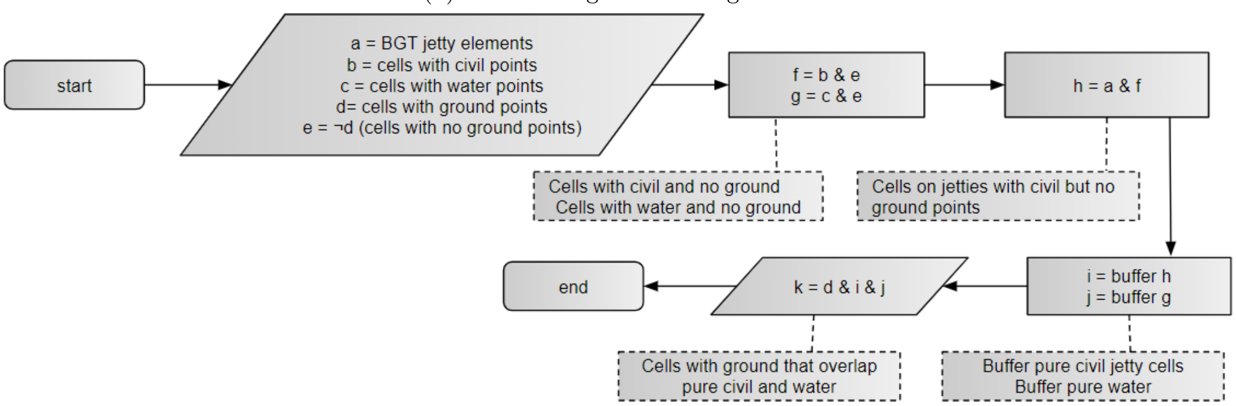
The acceptance or rejection of the classified point cloud happens at the tile level. From time to time after manual inspection, an automatic report generator is used to generate a report to help verify the status of each tile (Figure 21). The report contains details of each tile's error types and the corresponding error



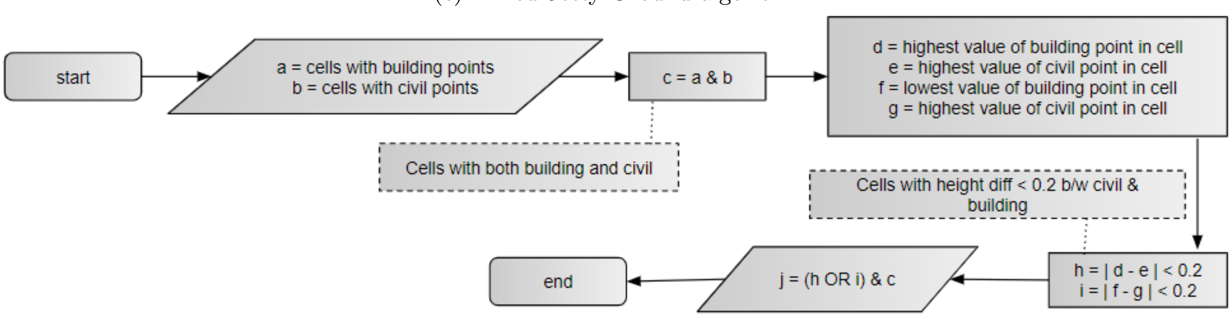
(a) top left to right: Bridge-Ground, Bridge-Other-Building, Building-Civil. Bottom: Jetty-Ground, Bridge-Other



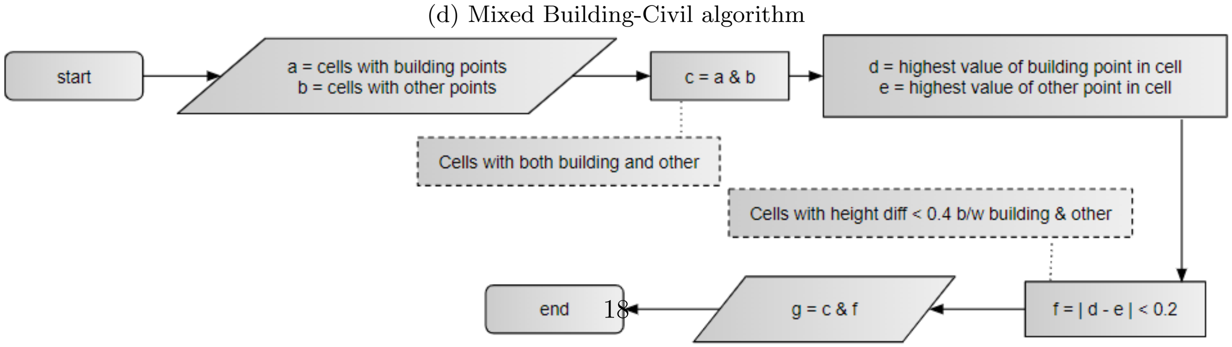
(b) Mixed Bridge-Ground algorithm



(c) Mixed Jetty-Ground algorithm



(d) Mixed Building-Civil algorithm



(e) Mixed Building-Other algorithm

Figure 19: Mixed classification detector

status.

When the rejections exceed threshold (Appendix A for more details), status is ***Invalid***, and if rejected targets are less than threshold then the status of the tile is ***Invalid***. If the manual checking is yet to be done and potential targets exceeds the threshold then the status is ***Risk***.

	A	B	C	D	E	F	G	H	I
1		Ground classified as something else [Area] - R1				Other object classified as ground [Objects] - R9			
176	14en2	0	0,0194	0,271049496	valid	0	0	2,71049496	valid
177	14ez1	0	0,0171	0,3125	valid	0	0	3,125	valid
178	14ez2	0	0,0911	0,3125	valid	1	1	3,125	valid
179	14fn1	0	0	0,2019808	valid	0	0	2,019808004	valid
180	14fz1	0	0,4315	0,311529428	risk	0	0	3,11529428	valid
181	14fz2	0	0	0,066011822	valid	0	0	0,660118224	valid
182	15bn1	0	0,0049	0,043192011	valid	5	5	0,431920108	invalid
183	15bn2	0	0,0249	0,29312158	valid	11	11	2,9312158	invalid
184	15bz1	0	0,025	0,025266107	valid	3	4	0,252661067	invalid
185	15bz2	0	0	0,167122436	valid	0	0	1,671224362	valid
186	15en1	0	1,570627682	0,3125	risk	1	1	3,125	valid
187	15en2	0	0	0,3125	valid	0	0	3,125	valid
188	15ez1	0	0,0141	0,237416927	valid	0	0	2,374169274	valid
189	15ez2	0	0,0141	0,258322945	valid	0	0	2,583229449	valid
190	15fn1	0	0,0217	0,3125	valid	0	0	3,125	valid
191	15fn2	0	0,0151	0,3125	valid	0	0	3,125	valid
192	15fz1	0	0,0519	0,167341383	valid	0	0	1,673413826	valid
193	15fz2	0	0,0145	0,219474533	valid	4	4	2,194745331	invalid
194	15gn2	0	0	0,002622622	valid	0	0	0,02622622	valid
195	16an1	0	0	0,3125	valid	0	0	3,125	valid
196	16an2	0	0,1075	0,3125	valid	2	2	3,125	valid
197	16az1	0	0,0099	0,306911949	valid	4	4	3,069119489	invalid
198	16az2	0	0,0063	0,3125	valid	4	4	3,125	invalid
199	16bn1	379,5077776	379,5133776	0,3125	invalid	0	0	3,125	valid
200	16bn2	379,5077776	379,6936776	0,3125	invalid	0	0	3,125	valid
201	16bz1	0	0,0747	0,256518895	valid	2	2	2,565188954	valid
202	16bz2	0	0,0735	0,310793832	valid	0	0	3,107938316	valid
203	16cn1	0	0	0,047849534	valid	0	0	0,478495345	valid
204	16cn2	0	0,0171	0,099496168	valid	2	2	0,994961675	invalid

Figure 21: Error report generated automatically

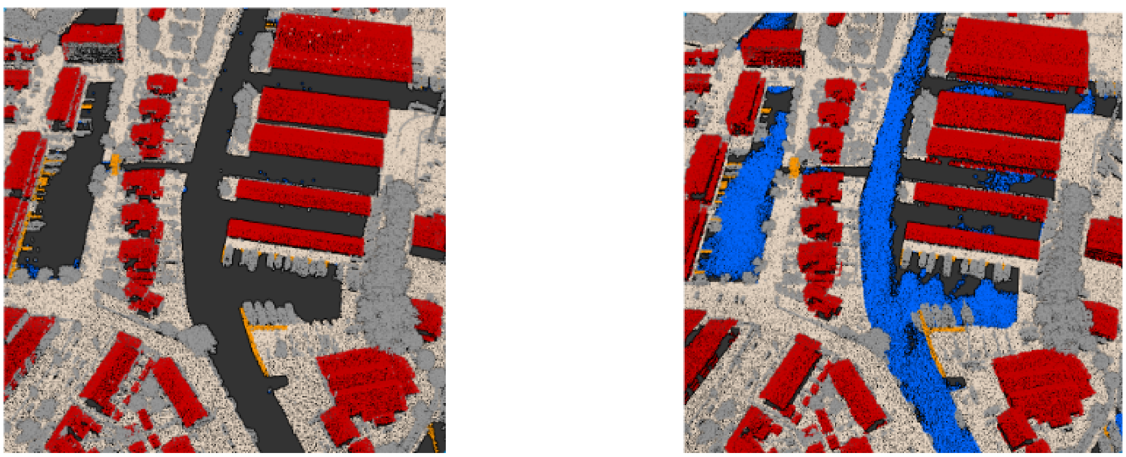
3 Results, Findings and Reflection

The detections mentioned in the previous chapter work mostly fine, with a few checks needing more time than usual because of a lot of false positives. For instance, R2 detection leads to lots of false positives because, for that detection, the algorithm uses a lot of reference data (BGT, BAG, AHN4). And this data at most of the places is not updated, like demolished buildings, removed jetties, newly constructed bridges, etc. Outlier detection also gives a lot of targets, but most of them are acceptable because points reflected from tall towers, and church spires are detected.

There are other classification errors, inconsistencies and also sometimes geometrically incorrect point clouds that escape the algorithms from the previous chapter. They are mentioned in the below subsections and the reasons for a few errors are also explained.

3.1 Scanning Technology and its Impact on Point Cloud

One of the main purposes of the AHN is to generate DTM and DSM maps of the country using the collected point cloud data and to keep a watch on flood-prone areas, and how they are changing over the years. Obviously, no LiDAR can effectively detect deep waters, however, it is crucial to detect shallow waters. If not, then the purpose of AHN is defeated.



(a) AHN5 (b) AHN4

Figure 22: Shallow water detection from AHN5 and AHN4

AHN5 has captured water poorly when compared to AHN4 (Figure 22). This could be attributed to the fact that AHN5 data is collected using Leica CityMapper - 2 which uses conical scanning technology, meaning only an oblique scan pattern is used. One of the plausible explanations is the lasers have to travel more in water and more energy is absorbed when compared to AHN4 (where push broom - nadir and near-nadir scanning pattern is used) and this results in less water data in AHN5.

3.2 Flight Strip Overlap

AHN data is airborne LiDAR data and data collected from multiple flight strips (Figure 23) have to be merged into one single big point cloud. But, sometimes it is not an easy task combining them for various reasons. For instance, if the merging has to be done on the irrigation fields, the grass might have grown within the time of the next flight strip collection, and the Iterative Closest Point (ICP) algorithm used might have had difficulty in aligning the point clouds. There could also be a systematic error or calibration mistake in one of the flight strips. As the exact reason is unknown, the possible reasons could only be speculated.

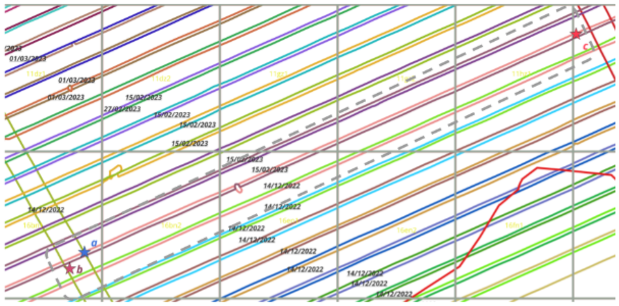
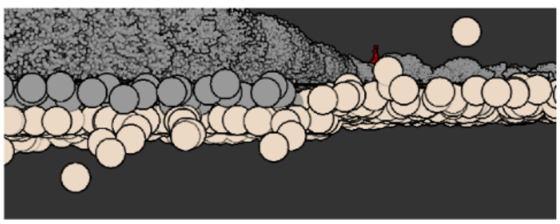
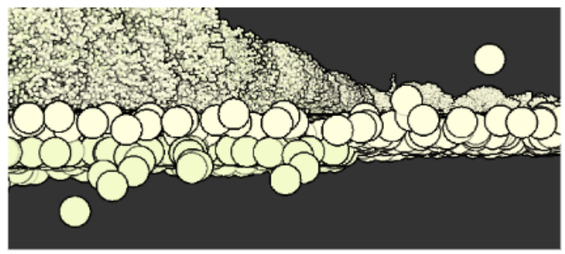


Figure 23: Flight paths and their overlap

In the highlighted region of Figure 23 (grey dotted rectangle), the two strips of data are collected at different times, one strip is captured on 14 Dec 2022 and the other on 15 Feb 2023. In the time gap of two months, as the overlap region is majorly constituted of agricultural fields, the fields definitely must have undergone some changes. It is very apparent in the point cloud (Figure 24) at a few places. However, it is hard to tell if the vegetation grew or if it's a faulty merging of both data. This leads to some serious inconsistency errors. As it could be seen from Figure 24a, points from a strip abruptly transitioned from ground to other classification, making a step of ground points.



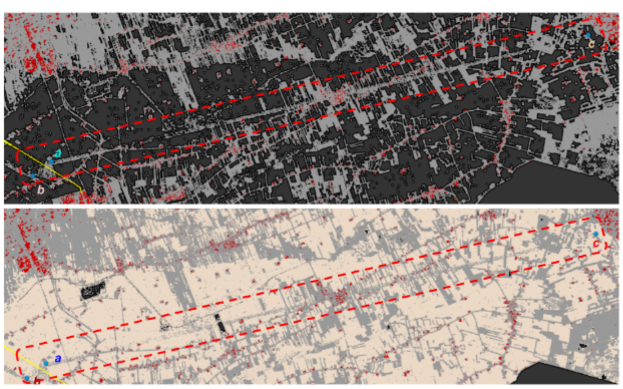
(a) AHN5 colour based on class labels



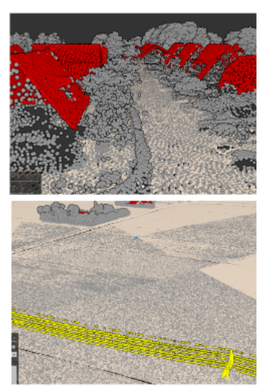
(b) AHN5 colour based on flight path

Figure 24: Flight path overlap issue

One of the major errors that led to the rejection of the first delivery of AHN5 data is also caused by an overlap issue. The *ground* points and *other* points are mixed over an area of four million m^2 (Figure 25). And misclassification at such a scale is not acceptable.



(a) Overlap affected area



(b) Mixed *ground* and *other* all over the place

Figure 25: Flight strip overlap - *ground* points mixed with *other* points

The Mixed classification errors are also mostly observed in the areas of flight path overlap (26). The exact reason is unknown but it could be speculated that each of the flight strip's data has been semantically segmented independently and the automatic classification algorithm struggled to label them. Meaning, that classifying point clouds of a place from different scans leads to different classifications.

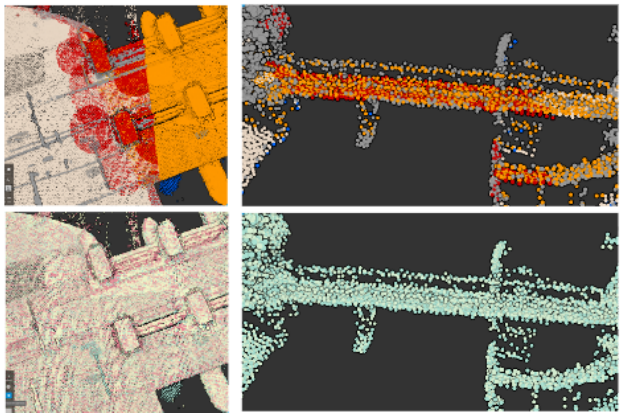
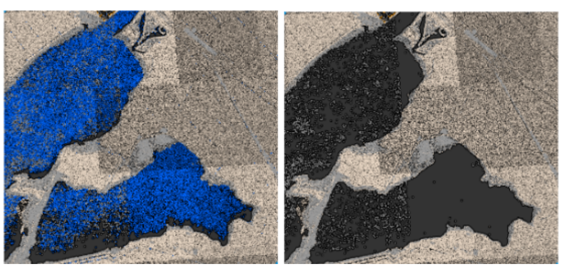


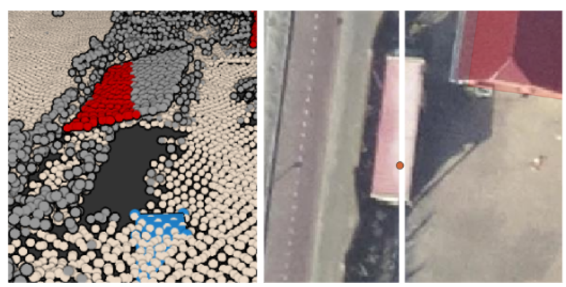
Figure 26: Mixed classification errors (top) and flight paths (bottom)

3.3 Production Tile Inconsistencies

We have seen how inconsistencies appear in the areas of intersection of data from different flight strips. Further inconsistencies are also observed at the intersection of AHN tiles. Cleaning of the data is done at the tile or sub-tile level, and the break in continuity of cleaning is very apparent. A sudden shift in classification (Figure 27b), points inside the buildings, *ground/other* points in water (Figure 27a), etc. are apparent. No specific algorithm has been implemented to find these inconsistencies, they could only be identified while manually inspecting or in other detections.



(a) Other points on water cleaned only in one tile



(b) Building labelled partially on one tile

Figure 27: Production tile inconsistencies

3.4 Blind Compliance with Reference Data

In the same way, as in quality control, the reference data is used for detecting misclassifications, it appears that in the classification process reference data has been used as well. But this blind compliance with reference data (AHN4, BAG, BGT) can lead to some unexpected results, cause they are sometimes not up-to-date. For instance (Figure 28), jetties might have been removed and BGT not updated; AHN4 has a jetty but not anymore in reality; incorrect BAG building polygon. No special detection technique was developed, could only be found if any other detection finds them.

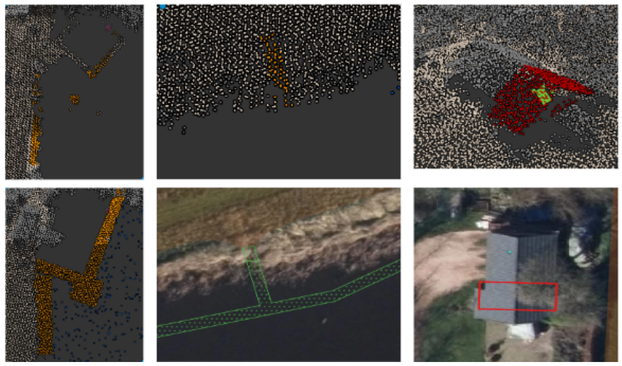


Figure 28: Blind compliance: AHN5 on top, reference at bottom (AHN4, BGT, BAG)

3.5 Detections that could be developed

Water is very important to be classified correctly. Water should not be misclassified, and also something else should not be misclassified as water. But in AHN5 jetties in few neighbourhoods are inconsistent with classification. Especially jetties could be seen either with *ground* points bordering it or *water* points bordering it 29.

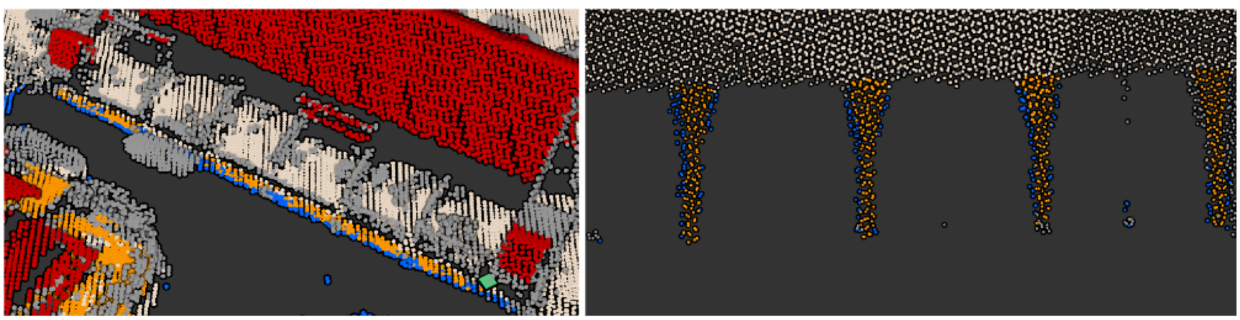


Figure 29: Water bordering jetties

Further, special attention has to be given to regions of flight strip overlap and production tile boundaries. Because in these areas a sudden shift or absurd shift in classification labels is observed, a detection which can identify these sudden shifts in these areas will give some interesting results and also some time is saved in manual inspection.

4 Conclusions and Recommendations

Detecting water is one of the main aims of the AHN project. However, it is very conspicuous when comparing AHN4 to AHN5, that AHN4 had captured water better than AHN5. This can be attributed to differences in the equipment used. And further, when compared to AHN4, AHN5 is also inferior in quality when comparing jetties. Jetties at some places are inconsistent in classification with mixed water and ground. Building classification is also not consistent. As the definition of the building changed from AHN4 to AHN5, where only the roofs are supposed to be classified as *building* and the rest as *others*, the inconsistency is very conspicuous. Roofs are well classified but the rest is mostly messy with mixed classification of *building-other*.

The acquisition of AHN5 data started in December 2022 and was made publicly available in December 2023. Even though it is a huge area to cover, it is still an incredibly long time to make data publicly available. The workflow from acquiring, preprocessing, classifying, and quality control is very time-consuming, and multiple companies are involved. Not compromising data quality takes time and effort, and it is evident with AHN5.

The quality control of misclassification is now dependent on more than fifteen different types of detection algorithms. Despite that, there are still even more errors which are difficult to find with hard-coded algorithms, and there is a need for some more algorithms. All the major error detections, from R1 to R10 errors are well detected except R2, where a lot of false positives are detected. However, inconsistencies are even harder to find. For instance, mixed detectors failed to detect jetty-ground inconsistencies. Further, the big overlap of ground-other points was also not detected. One of the reasons it is difficult to identify small inconsistencies is because of the data abstraction to 2D, where data can't be captured below 1 m resolution, but inconsistencies happen at finer scales.

Algorithms can only be hard-coded if something is expected to be wrong, if not it can't be coded. For example, mixed detectors are developed for AHN5, and these detections were not there for AHN4 because they are not seen in AHN4. A lot of human inspection is needed to figure out different types of errors that are present in the data. Then the algorithms have to be developed, tested, and improved, based on feedback it has to be improved again. Which works perfectly fine in most cases. But this is tricky with few errors. To this end, I see a possibility of integrating machine learning models into the quality control process. I believe a well-trained model can identify the errors, and give them labels. As it is hard to give labels to inconsistencies, they can all be categorized under one name. Then a 2D algorithm can classify the detections based on machine learning outputs.

A Appendix - Information on AHN Point Cloud Classification and Error Metrics

A.1 Point Cloud Segmenmtation

The classification labels given to points and their definitions are mentioned in Table 1. Concerning the semantic segmentation of the point cloud there are a few changes in the meaning given to points over the versions of AHN:

1. Prior to AHN5, all the points of the building, including walls, roofs, windows, sunshades, etc, are given *building* classification. However, with AHN5, only the points that reflected from the roof are classified as *buildings*, and the remaining are of classification *others*.

2. In AHN3, there is no classification of *high-tension*, all the high-tension cables and towers are classified as others. In AHN4, only the high-tension cables are given the classification of *high-tension*, but the towers that hold them are of classification *others*. Coming to AHN5, both the cables and towers are labelled as *high-tension* (Figure 30).

Label	Definition
0	never classified
1	other
2	ground
6	building
9	water
14	high-tension
26	civil structure (bridges and jetties)

Table 1: AHN point cloud label values and definitions

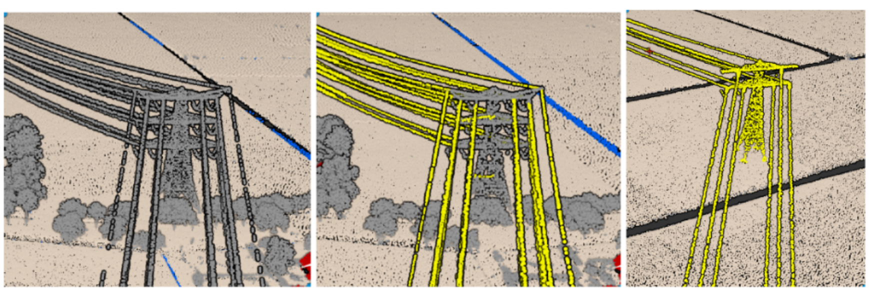


Figure 30: High-tension classification evolution - AHN3 (left), AHN4 (middle), AHN5 (right)

A.2 Misclassification Errors and types

Misclassification errors are of mainly ten classes, and they are presented in Table 2. There are other errors other than these major ten classes, and they are *Extremes* (outliers, ground points within buildings); *Completeness*, when points are not classified; and *Structural*, when a neighbourhood has repeating pattern of same mistake. The threshold values of acceptance for each AHN tile of size 5 km x 6.25 km are in Table

		Assigned class					
		Ground	Building	Water 3	High-tension	Civil	Other
Actual Class	Ground		R1	R1	R1	R1	R1
	Building	R2		R3	R3	R3	R4
	Water	R5	R5		R5	R5	R5
	High-tension	R2	R6	R7		R7	R6
	Civil	R2	R8	R8	R8		R8
	Other	R9	R10	R10	R10	R10	

Table 2: Types of classification Errors

Error type	**Threshold values**
Extremes	Maximum of 1 per 1000 hectares → 3.25 pts/tile
Completeness	0 → all points should be classified
Clustered	None allowed
Structural	None allowed
Large areas with classification errors	Smaller than 1 hectare
R1 - Ground not classified as ground	1 ha in 10000 ha → 0.3125 ha/tile
R2 - Man-made objects as ground	1 object per 1000 ha → 3.125 objects/tile
R3 - Building as water/high tension/civil	1 building per 1000 buildings
R4 - Building as other	1 building per 100 buildings
R5 - Water not as water	1 water body per 10000 ha
R6 - High tension as building/other	1 high tension tower per 100 towers
R7 - High tension as water/civil	1 high tension tower per 1000 towers
R8 - Civil as building/water/high tension/other	2 civils per 100 civils
R9 • Low vegetation (<0.5 m) as ground • High vegetation (>0.5 m) as ground • Other objects (poles, boats, etc) as ground	• 1 ha in 10000 ha → 0.3125 ha/tile • 1 ha in 10000 ha → 0.3125 ha/tile • 1 object in 1000 objects
R10 • Low veg as building/water/high-tension/civil • High veg as building/water/high-tension/civil • Other objects as building/water/high-tension/civil	• 1 ha in 50000 ha → 0.625 ha/tile • 1 ha in 50000 ha → 0.625 ha/tile • 2 objects in 100 objects

Table 3: Error's per tile threshold values

References

[1] Yong He, Hongshan Yu, Xiaoyan Liu, Zhengeng Yang, Wei Sun, Yaonan Wang, Qiang Fu, Yanmei Zou, and Ajmal Mian. Deep learning based 3d segmentation: A survey. *arXiv preprint arXiv:2103.05423*, 2021.

[2] Markus Schütz, Stefan Ohrhallinger, and Michael Wimmer. Fast out-of-core octree generation for massive point clouds. In *Computer Graphics Forum*, volume 39, pages 155–167. Wiley Online Library, 2020.

[3] Zhaoyu Su, Pin Siang Tan, and Yu-Hsing Wang. Dv-det: Efficient 3d point cloud object detection with dynamic voxelization. *arXiv preprint arXiv:2107.12707*, 2021.

[4] Richard Szeliski. *Computer Vision: Algorithms and Applications.* Springer Science Business Media, 2010.

[5] Yuxing Xie, Jiaojiao Tian, and Xiao Xiang Zhu. Linking points with labels in 3d: A review of point cloud semantic segmentation. *IEEE Geoscience and remote sensing magazine*, 8(4):38–59, 2020.

[6] Qijian Zhang, Junhui Hou, Yue Qian, Yiming Zeng, Juyong Zhang, and Ying He. Flattening-net: Deep regular 2d representation for 3d point cloud analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

B Personal Reflection on Internship

B.1 Reflection of MSc Geomatics from the lens of my internship experience

The knowledge accumulated in the first year of geomatics gave a nice starting point to understand what point cloud is and how it could be used. The concepts of Inertial Measuring Units (IMU) and Global Navigation Satellite Systems (GNSS) devices, and their role in point cloud data acquisition using LiDAR are very well discussed in the GEO1001 Sensing Technologies and GEO1003 Positioning and Location Awareness. Theoretically, the concepts and implications of errors in point cloud-like, geometrical errors, systematic errors, and registration errors are discussed, but how to practically rectify them is not. Further, the basic automatic techniques of classifying *ground* points, for instance, *Cloth Simulation Filter*, are discussed in GEO1015 Digital Terrain Modelling.

In practice, especially for large-scale data acquisition projects, the first crucial step is to georeference the captured point cloud or image data. Along with georeferencing, aligning multiple lidar datasets is also a very common task after the data acquisition step. It would be a great addition if the mathematical concepts behind these techniques could be added to Geomatics curriculum.

Geodelta uses Potree technology ([2]) for a fast web viewing experience, where octree data structures are employed to store the point cloud. The concepts of octrees and efficient data retrieval of the point cloud are explained in GEO1006 Geo Database Management Systems.

B.2 Learnings from Internship and Personal Application

The internship was a great learning experience, which gave me a basic understanding of the point cloud from the beginning to the end of the pipeline. An in-detailed understanding of the quality control process especially for classification errors is developed. Here are a few things I learned during the internship:

1. Basic understanding of the pipeline of point cloud data, from acquiring to classification and delivering.
2. Types of errors that can be expected in a point cloud, which are introduced at multiple levels in the whole process. For instance, systematic errors are introduced at the acquisition stage; and Geometrical or Registration errors during the point cloud merging stage; Classification errors build on all the previous stages and also by faulty algorithms.
3. Types of classification errors in the point cloud, and ways of detecting them, both algorithmically and manually. Effects of misclassifications and their implications on products developed based on AHN like DTM, DSM, and 3D BAG.
4. Advantages and disadvantages of abstracting point cloud data to 2D.
5. Applicability of Potree for smooth viewing of massive point clouds, and scope for building applications on top of it.
6. Finally, the experience of working at a company gave me an understanding of how a professional life differs from a student life.

I have started using Potree for my thesis project. In traditional software like CloudCompare, each time Point Cloud has to be loaded the software takes a lot of time, and for large datasets like AHN, the interaction is very slow and also occasionally crashes. Potree takes a bit of time upfront to understand how it works, but it is incredibly helpful later. Especially for me because my thesis is on point clouds, and I interact with point clouds daily, so a lot of time is saved just by not loading my data each time I have to look at it. Further, it also provides customizability with data points to make good images for my reports.

B.3 Topics That Could be Added to Geomatics

I feel after this internship it would be great if Geomatics could offer a course entirely on point clouds as an elective. Which includes acquisition, merging, error handling, the importance of sensors inside and how to calibrate them, automatic classification algorithms on point cloud, and how to set up Potree for easy handling of the data. Such an elective can help bring together most of the topics discussed in other Geomatics courses and consolidate the understanding and practicality of it. Data structures on how to handle point cloud should also be explained, and be made part of the curriculum. Using point cloud just as it is, in its original form is challenging and a skill in itself. Finally, I believe point cloud is a representation of the real world, and so are aerial or drone images. Integrating images and point clouds and joining them together can give a lot of value, so handling multiple 2D and 3D datasets together should also be part of Geomatics.