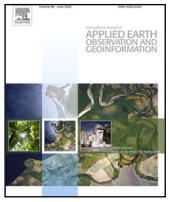
International Journal of Applied Earth Observations and Geoinformation 116 (2023) 103174

Contents lists available at ScienceDirect

International Journal of Applied Earth Observation and Geoinformation

journal homepage: www.elsevier.com/locate/jag



Distributed processing of Dutch AHN laser altimetry changes of the built-up area

Máté Cserép [a,*], Roderik Lindenbergh [b]

[a] Faculty of Informatics, ELTE Eötvös Loránd University, Budapest, Hungary
[b] Faculty of Civil Engineering and Geosciences, Delft University of Technology, Delft, Netherlands

ARTICLE INFO

Dataset link: http://www.ahn.nl/, http://dx.doi.org/10.17632/yrxvhfj5jv.1

Keywords:
LiDAR
Change detection
Object recognition
Big data
Cloud computing
AHN

ABSTRACT

The evolution and spreading of data capturing methods ranging from simple GPS devices like smart-phones to large scale imaging equipment – including very high resolution and hyperspectral cameras and LiDAR – resulted in an exponential growth in the amount of spatial data maintained by companies and organizations. At the same time methods for extracting information from such data are often behind in efficiency. In this paper we analyse the possibilities for nation-wide change detection of massive airborne laser altimetry point clouds, based on digital elevation models generated from them. The proposed workflow distinguishes modifications in the built-up area from other changes and noise. Our methodology combines different area processing spatial algorithms: object detection, noise filtering, morphological operations and clustering. Our proposed method is designed to scale dynamically on extensive datasets by processing a spatially partitioned input dataset in an easily parallelized manner. Favourable visualizations and aggregated representations of the results are examined, followed by a discussion of feasible validation methods. As a demonstration we showcase the implemented distributed evaluation of our workflow on the full Dutch altimetry archive – a dataset exceeding several terabytes of storage space – using a high-performance computing environment. While the average execution time was 47 h on a desktop computer, our solution only took less than 2.4 h to complete. The output was validated against the building layer of the TOP10NL topographic dataset, proving a 70% accuracy nation-wide and over 90% for urban areas. As a result our analysis shows that The Netherlands experienced an aggregated building volume change of 912.33 km^3 between the acquisition of AHN-2 and AHN-3.

1. Introduction

Alterations in the urban or more generally in the built-up areas can be caused by either planned, human-made changes like construction, demolition, modifications or by natural disasters like earthquakes. Detecting these changes is essential for government agencies and authorities on several fields ranging from land usage through urban planning and civil engineering to disaster management.

The advancement of remote sensing and Light Detection and Ranging (LiDAR) in the last few decades described by Zhang (2010) offered a technology capable of rapid high resolution collection of surface altimetry data through airborne laser scanning (Van der Sande et al., 2010). Development towards collecting dense point clouds from large distances allowed us to develop methods for recognizing buildings based on remotely acquired point clouds (Tomljenovic et al., 2015). Numerous methods have been developed for urban classification and building extraction from digital surface models (Weidner, 1997; Priestnall et al., 2000) or from TIN models (Wang and Schenk, 2000).

Beside detecting buildings in a single epoch, identifying their modifications between multiple data acquisitions has also received significant scientific attention in the decades. This provides an automatized, therefore more cost efficient and faster solution in contrast to expensive and time-consuming field surveys and manual data evaluation. A commonly used method is to work with and compare $2\frac{1}{2}$D digital surface models, as beside reducing the amount of data, it also simplifies the task for appointing point pairs (Butkiewicz et al., 2008; Priestnall et al., 2000). Various type of supplementary data sources can also be utilized to increase the precision of building recognition. Georeferenced aerial raster imagery can easily be used together with point clouds (Xie et al., 2006; Du et al., 2016). The method developed by Vu et al. (2004) depends on a building inventory of the scanned city, while (Vögtle and Steinle, 2004) classify the surface before the change detection procedure to achieve better accuracy. Zhou et al. (2020) use very high resolution (VHR) aerial stereo images to generate a photogrammetric point cloud and detect changes compared to a previously acquired

* Corresponding author.
E-mail addresses: mcserep@inf.elte.hu (M. Cserép), r.c.lindenbergh@tudelft.nl (R. Lindenbergh).

https://doi.org/10.1016/j.jag.2022.103174
Received 24 September 2022; Received in revised form 16 December 2022; Accepted 28 December 2022
Available online 4 January 2023

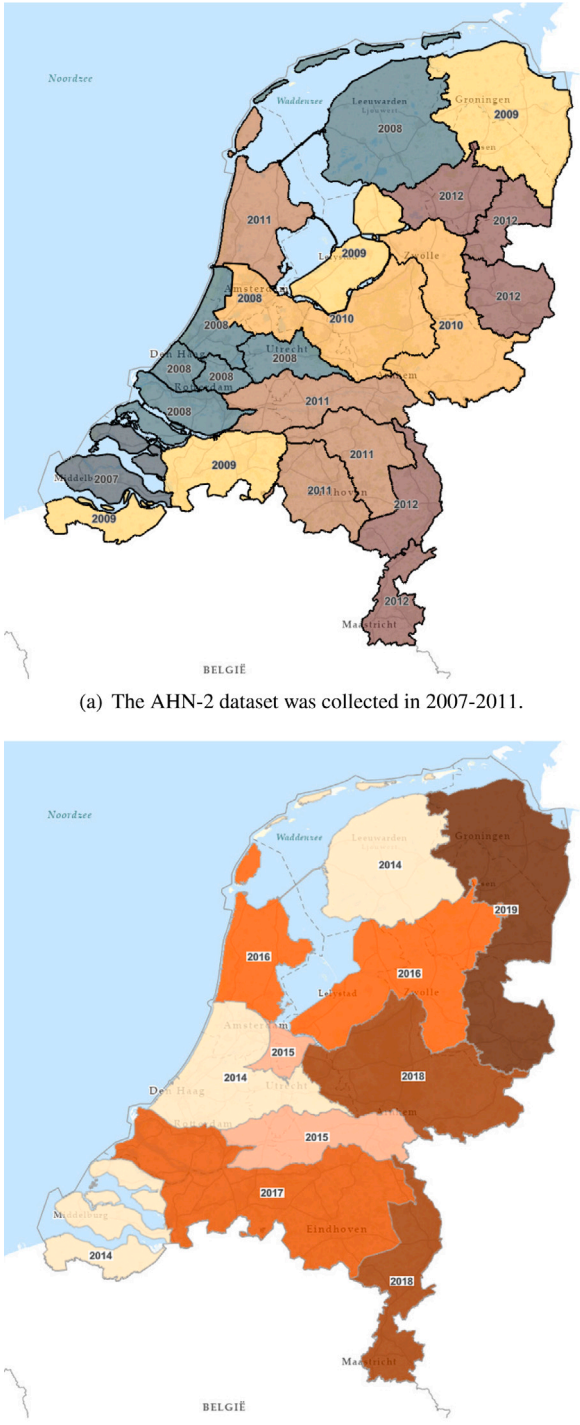
LiDAR point cloud, thus reducing the usually longer time interval (multiple years) between two LiDAR measurements of an area. The output of the change detection analysis can again be combined with other data sources, as Van Natijne et al. (2018) presented with Interferometric Synthetic Aperture Radar (InSAR) data for deformation monitoring .

Instead of working on $2\frac{1}{2}$D elevation models, change detection methods on the point cloud level also gained focus in the past decade. These approaches often measure the cloud to cloud (C2C) or cloud to plane (C2P) distance of point pairs (Richter et al., 2013). Alternatively the Iterative Closest Point algorithm (ICP) is used by Matikainen et al. (2010) or Scott et al. (2018) to detect 3D translations between point clouds from different epochs. Politz et al. (2021) apply a combined method, where within each raster grid cell, the height distribution of all points for two moments in time is considered by exploiting the Jensen–Shannon distance to measure their similarity. Nowadays, even multi-directional change detection is used to detect the dominant movement direction of the ground (Williams et al., 2021). Most recent studies achieved to generate 3D models from building footprints and airborne LiDAR point clouds (Dukai et al., 2019), even on a LoD2 level, thus representing buildings with roof shapes (Peters et al., 2022). Such detailed urban building databases can be used widely, e.g. to estimate the energy demand of residential buildings (León-Sánchez et al., 2021).

Neural networks and deep learning are currently extremely actively researched topics. Especially convolutional neural networks (CNN) have proven to be applicable for image classification tasks. Building detection based on VHR aerial images was also addressed with CNNs (Sun et al., 2017; Ji et al., 2019). Recently, Politz and Sester (2022) presented a residual neural network, which detects building changes using height and class information on a raster level based on LiDAR or photogrammetric point clouds. While these results look promising, evaluation is usually carried out on a relatively small area, often not exceeding a few dozens of square kilometres.

Meanwhile the increasing quantity and improving quality of measurements raised new challenges regarding the computation and memory efficient analysis of massive point cloud datasets. Distributed and cloud computing systems have been around for years, proven to be notably useful in static or rarely altering big data processing, applied in numerous fields including Geographic Information Systems (GIS) (Yang et al., 2010). Previous research addresses the importance of distributed, cloud-based storage (Boehm, 2014) and management (Yang and Huang, 2013) of these rapidly growing spatial datasets. Various approaches for the efficient multithreaded loading and processing of point clouds have been researched (Badenko et al., 2019). Distributed LiDAR processing towards digital elevation models also received significant attention from the scientific community (Hegeman et al., 2014; Jian et al., 2015). Recent advancements in the field propose design guidelines on how to specify and implement a complex service-oriented framework for massive multi-temporal point cloud storage, processing and visualization (Discher et al., 2019). However, analysis of spatial features on higher abstraction level is still an unsolved challenge in multiple aspects.

In our paper we propose a methodology to automatically evaluate altimetry change detection of massive multi-temporal datasets on distributed high-performance computers (HPC) or in a cloud computing environment like Hadoop[1] or Spark.[2] As example measurements, the multi epoch nation-wide AHN[3] (*Actueel Hoogtebestand Nederland*) altimetry archive of The Netherlands was selected for demonstration. The AHN project provides publicly available altimetry data for the whole territory of The Netherlands, extending across approximately 40.000 square kilometres, containing data points on the magnitude of trillions (Swart, 2010). Since the launch three data acquisitions were completed between 1996–2000, 2007–2011 and 2014–2019, while the fourth one is ongoing and planned to be completed not before the end of 2022. In our research we have selected the two most recent, complete measurements (AHN-2 and AHN-3) for comparison, to recognize the changes in man-made structures which occurred in the 7–8 years long timespan of their difference. The year of data acquisition for the different regions in the country is depicted in Fig. 1.



(a) The AHN-2 dataset was collected in 2007-2011.

(b) The AHN-3 dataset was collected in 2014-2019.

Fig. 1. Data acquisition periods for the *Actueel Hoogtebestand Nederland* dataset.

Our study focuses on larger scale changes in the built-up area, where complete buildings or building blocks were constructed, demolished or rebuilt between the analysed epochs. Such alternations could also be

[1] http://hadoop.apache.org/
[2] http://spark.apache.org/
[3] http://www.ahn.nl/

Fig. 2. Tile boundaries of the complete AHN dataset.

properly validated against national registers of buildings where publicly accessible. The most important contributions of our research are (*i*) defining an algorithm pipeline for building recognition and change detection based on digital elevation models; (*ii*) creating a robust, automatized, open-source software framework capable of processing large datasets efficiently; (*iii*) evaluating our approach on the nationwide Dutch AHN point cloud derived 0.5 m raster product, analysing and validating the results; and (*iv*) providing an interactive, publicly accessible web-based visualization of the results.

2. Methodology of change detection

In this section we present a detailed introduction to the AHN dataset and the quality of the data acquisitions, followed by an overview of the data evaluation workflow demonstrated on a suitable example area. The section continues with the comprehensive description of the provided solution and its algorithmic steps.

2.1. Dataset description

The complete AHN dataset is provided through 1.368 tiles per epoch, each of them covering an area of 31.25 km^2. (The covered area from the territory of The Netherlands can be smaller in the edge tiles.) The tile boundaries of the dataset are shown in Fig. 2.

The quality of the AHN-2 and AHN-3 measurements is similar, no significant improvement has been reported in the dominant factors. According to the quality specification (PDOK, 2013) the average density of the AHN-2 point cloud is 10 data points per square metre and the vertical error threshold of the accuracy is below 0.2 m – for 99.7 percentage of the data. The precursory specification (PDOK, 2015) of AHN-3 assessment defines only slightly better requirements in these terms, improving the vertical accuracy to 0.15 m with the same criteria as mentioned before. The datasets are downloadable in two formats: beside the point clouds, preprocessed digital elevation model (DEM) grids with 0.5 m and 5 ms resolution were also made available by the data provider. The finer raster format with half metre resolution was rendered from the point cloud with a *Squared Inverse Distance Weighting* algorithm (De Smith et al., 2015), while the more coarse 5 m grid was resampled from the other in an unweighted manner.

Managing and processing point clouds on scale of billions or trillions is a complex task as was already shown for the AHN-2 dataset itself by van Oosterom et al. (2015). Utilizing the half metre resolution digital elevation models of AHN over the raw point clouds has multiple notable advantages:

Storage space requirement can be reduced by at least a magnitude considering the point density of the cloud. In fact the gain is even more major when the possibility of multiple laser pulse returns and the typical LiDAR metadata per point (flight line, scan angle, classification, etc.) is taken into account. The raw point cloud dataset in LAS[4] format for a single tile is typically over 15 GB, while the uncompressed raster grid is roughly only 500 megabytes.

Algorithmic complexity of the comparison and change detection of AHN epochs can be greatly reduced when data points are locked in a fixed grid and the datasets properly overlap with each other on the X and Y axis.

Evaluation time as a result of the above mentioned reasons is also significantly beneficial with this condition.

While the evident loss of precision of the dataset would be inappropriate for certain tasks (e.g. monitoring changes of individual trees), it is sufficient to detect alterations in the built-up area, hence our decision fell on the examination of the half metre raster grids.

2.2. Example area

In order to properly showcase our approach, a fitting demonstration area had to be selected, preferably containing types of all of the following zones:

- *construction site*: territory where buildings – with significant size – have been constructed or demolished in the given time period;
- *stable territory*: built-up neighbourhood, but without remarkable modification in the artificial objects;
- *green area*: zone in an urban vicinity where considerable alteration occurred due to natural reasons like vegetation growth or tear down.

The campus and the surroundings of the Delft University of Technology satisfies all these criteria and was therefore selected as the sample area – shown in Fig. 3 – for this section. Altogether 9 reference locations – detailed in Table 1 – were chosen to illustrate the proposed methodology. Our expectation is that in the case of locations $A_1 - A_3$ removal of buildings should be detected, while for $B_1 - B_3$ construction of new ones should be recognized. Finally locations $C_1 - C_4$ shall reveal no change in the built-up area.

2.3. Threshold filtering

As an initial prototype of change detection, the changeset between the surface elevation models (DSM) of the AHN-2 and AHN-3 data acquisitions shows the area selected for demonstration in Fig. 4. Although the theoretical accuracy of detecting altimetry changes is 0.35 m – as described in Section 2.1 –, a higher threshold of 1 m was applied as our research focuses on larger scale alterations in the built-up area which should always meet this assumption. Altimetry changes below the absolute value of 1 m were ignored.

While the expected changes described in Section 2.2 are clearly present in the changeset, it is extremely noisy with all the fluctuation of the vegetation and other alterations not related to buildings, like the reconstruction of the Mekelpark (reference site C_3) or vehicles on the highway (reference site C_4). In the following subsections we present our methodology to filter out and remove all unwanted alterations.

[4] http://www.asprs.org/

Fig. 3. Satellite image of the TU Delft campus area with indicated study locations.

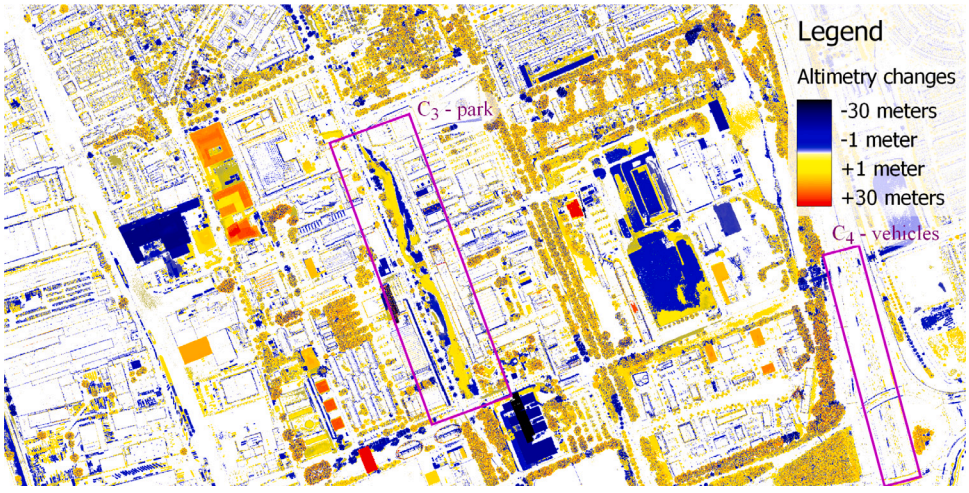


Fig. 4. Unfiltered altimetry changes between AHN-2 and AHN-3 measurements. The two locations marked with rectangles indicate a park and vehicles on a highway.

Table 1
Description of study locations at the TU Delft campus area.

Mark	Description
A_1	The old building of the Faculty of Architecture which was devastated in an extensive fire and was completely tore down afterwards;
A_2	A removed unused warehouse building;
A_3	Demolished office of the TNO Research Facilitates;
B_1	The new site of The Hague University of Applied Sciences and the InHolland University of Applied Sciences, located next to the TU Delft campus;
B_2	Newly built apartments by the student housing corporation *DUWO*;
B_3	A freshly constructed multihousehold building;
C_1	Industrial area almost west to the campus without notable artificial alteration and very low ratio of green vegetation;
C_2	A garden suburb without significant modification in the built-up area, but prosperous change of the vegetation surrounding the buildings and streets;
C_3	The Mekelpark which was formed in place of a busy road;
C_4	A highway with vehicles on it.

2.4. Detecting objects

To clean the changeset, distinguishing built-up areas is a vital part. Building detection based on digital elevation models has been thoroughly researched previously, offering various kind of solutions to handle this issue. Some of the most widely applied techniques include comparing ground (terrain) and surface level DEM (Ma, 2005; Vögtle and Steinle, 2004) to extract objects; the land cover classification of data through supplementary coloured orthoimagery (Haala and Brenner, 1999); and detecting building edges either by histogram thresholding (Vu et al., 2004) or contour extraction (Xie et al., 2006).

Alongside the surface elevation model (DSM), a terrain level model (DTM) is also provided within the AHN dataset, enabling an easy implementation of the first indicated method without any computational overhead — on the cost of doubling the storage space requirement of the input dataset. These DTMs contain no height information – marked with an extremal *nodata* value – in areas where the ground surface was covered, thus the location of such objects can be determined in a straightforward manner. According to the quality specifications referred to in Section 2.1 the DTMs were directly calculated from the point cloud data acquisitions, assuring a superior quality – in contrast when it is produced from the DSM – by the usage of possible multiple returns of laser pulses. Hence it is feasible to filter out partially transparent objects like trees solely by this observation.

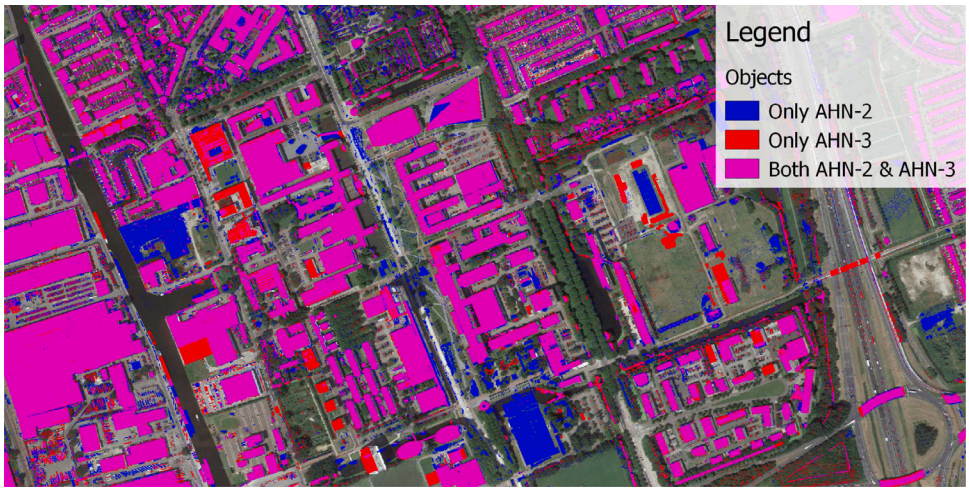


Fig. 5. Areas potentially containing objects by comparing AHN DSM and AHN DTM.

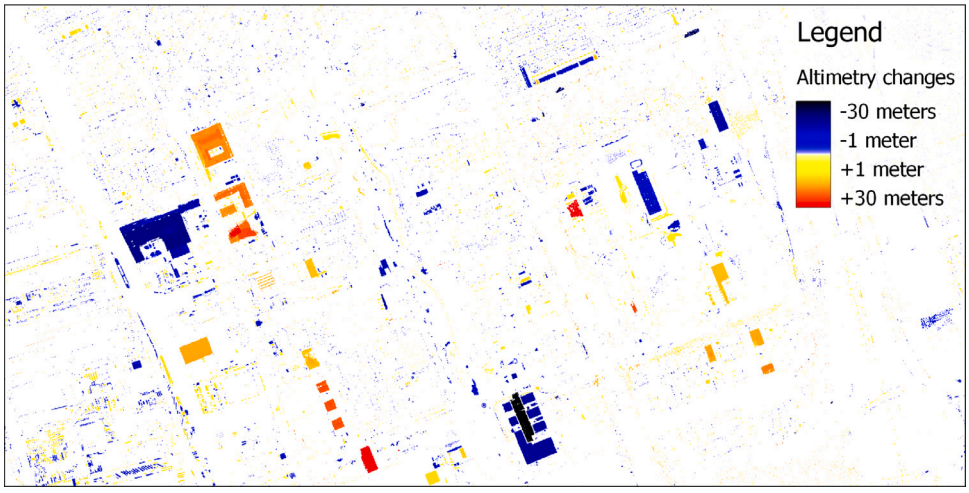


Fig. 6. Application of object extraction by DTM-DSM comparison followed by noise filtering.

Fig. 5 shows the objects identified in the example area by the comparison of the terrain and surface elevation models, distinguishing locations where the ground level was covered in only one or both of the AHN datasets. As visualized apart from infrastructure like buildings and bridges, coherent areas of trees and cars were also detected as objects concealing the ground, therefore additional filtering is required to reach an adequate result.

2.5. Changeset filtering

A common statement for the structures in the built-up area desired to be distinguished is that they are immoveable, hence when unmodified the altimetry values on a raster grid should show no significant change — although building facades may appear as modifications due to small mis-alignment between AHN-2 and AHN-3. On the other hand most vegetation provides a high frequency of noise since the laser pulses may measure significantly differing values of height data and thus can be erased from the changeset as shown in Fig. 6.

The noise filtering algorithm (Gonzalez and Woods, 2008) applied also must take into account that in some cases considerable noise may appear in the elevation change of buildings:

- when a structure is constructed on a previously irregular ground surface like a forest; or
- when after demolition the terrain remains rough or vegetation starts growing.

We defined noise as the average percentage of absolute height difference compared to the surrounding areas of 2.5 by 2.5 m ($5 * 5$ pixels). The formula shown by Eq. (1) calculates the noise percentage for a given x, y position with a given range $r = 2$ in our case, where $C_{x,y}$ defines the altimetry change for that location. To rule out false positive removals, an outcome exceeding 50% was deemed noisy.

$$noise(x,y) = \frac{\sum_{\substack{-r\le i\le r \\ -r\le j\le r}} \frac{\left|C_{x,y} - C_{x+i,y+i}\right|}{min\left(\left|C_{x,y}\right|, \left|C_{x+i,y+i}\right|\right)}}{(2r+1)^2} \tag{1}$$

Our research focused on larger, building level changes, however results still contained some negligible modifications (e.g. construction of a chimney), small movable objects like vehicles and the remnants of vegetation which managed to pass the noise filter. Therefore as a final filtering operation, small changes – below 100 m^2 of area – were discarded through a clustering algorithm (Gonzalez and Woods, 2008). Results for the example area are presented in Fig. 7 showing a clear image of all artificial modifications applied in the built-up areas, all disturbing elements are filtered and abolished.

2.6. Border reconstruction

With careful observation and comparison of Figs. 4 and 7 we visibly notice that the applied noise filter in Section 2.5 affected the outer

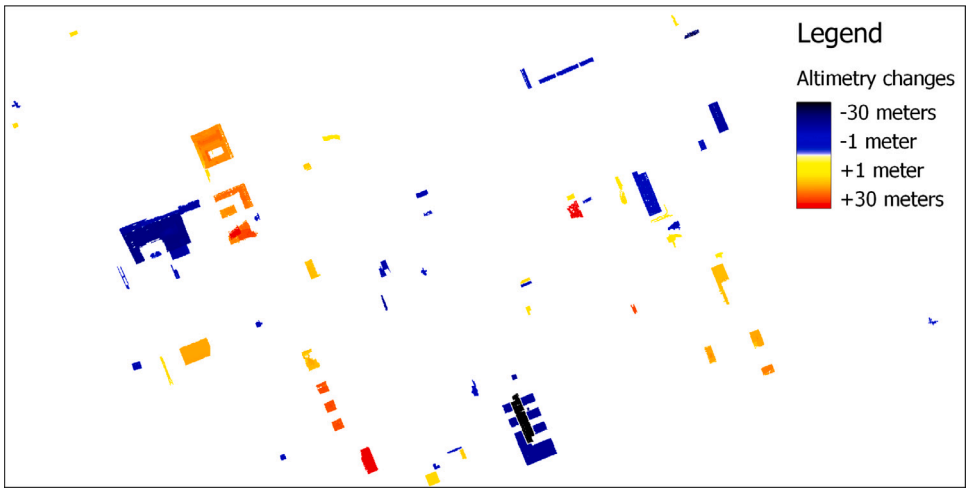


Fig. 7. Cluster filter applied to ignore modifications on an small scale.

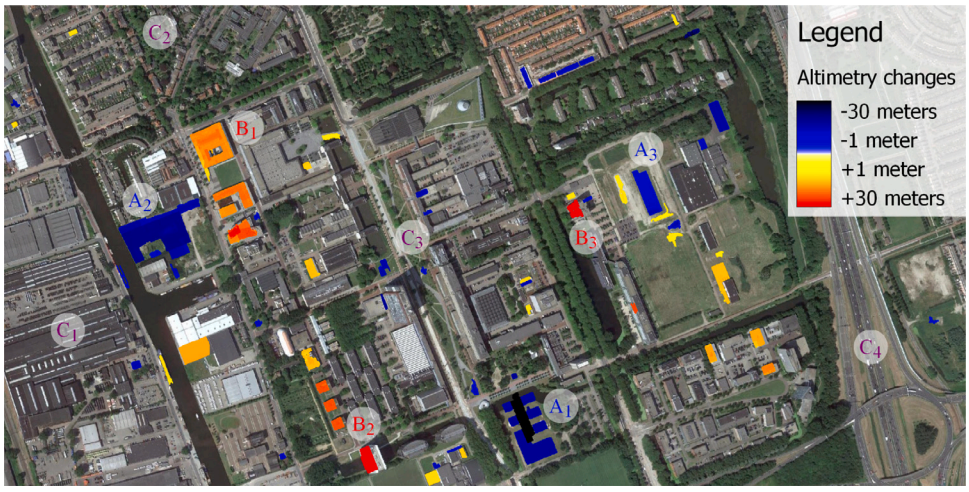


Fig. 8. Final results produced through border reconstruction of buildings.

border of – especially high – buildings. This effect was expected as the steep raise of elevation data results in a high frequency of change and thus noise on such locations. Multiple buildings contain tiny (few pixels large) holes which were either already present in the initial changeset of Fig. 4 or were introduced as an outcome of initial irregularities inside building areas (or even glass surface on top of buildings) in the AHN dataset through the object filtering demonstrated in Fig. 6.

To deal with these minor issues, boundaries were expanded through application of a morphological dilation operator (Gonzalez and Woods, 2008) in a $3 * 3$ pixel range, interpolating *nodata* values with the average of their surroundings. The before mentioned small holes were filled by the utilization of majority filtering (Liu and Mason, 2013) using a $5 * 5$ pixel range. The final results (Fig. 8) completely meet the expected outcome for the reference areas marked in Section 2.2.

2.7. Algorithm summary

The presented algorithm aimed to filter out changes in the built-up area solely based on the AHN datasets, can be decomposed into 7 major reproducible steps (and visualized in Fig. 9):

1. DSM versus DTM comparison separately on the AHN-2 and AHN-3 datasets — to filter out ground-level areas
2. Creating initial changeset by producing differences between the datasets
3. Threshold filtering — with 1 m elevation change
4. Noise filtering — with 50% relative threshold on a 3 by 3 window
5. Cluster filtering — with 100 m^2 threshold size and a 4-way connectedness
6. Morphological dilation — on a 3 by 3 window
7. Majority filter:
   (a) with a range of a 3 by 3 window
   (b) with a range of a 5 by 5 window

2.8. Aggregation overview

In a local environment, changes of single buildings can be the focus of interest. However in a broader overview of an area, the accumulated values of modifications are easier to interpret. The CBS[5] (*Centraal Bureau voor de Statistiek*) provides the official boundaries of Dutch administrative units as municipalities, (electoral-) districts and neighbourhoods in vector format. These units of territories were used to compute the following aggregated values for each area:

- *gained*: the added (built) volume of artificial content per hectare;

[5] http://www.cbs.nl/

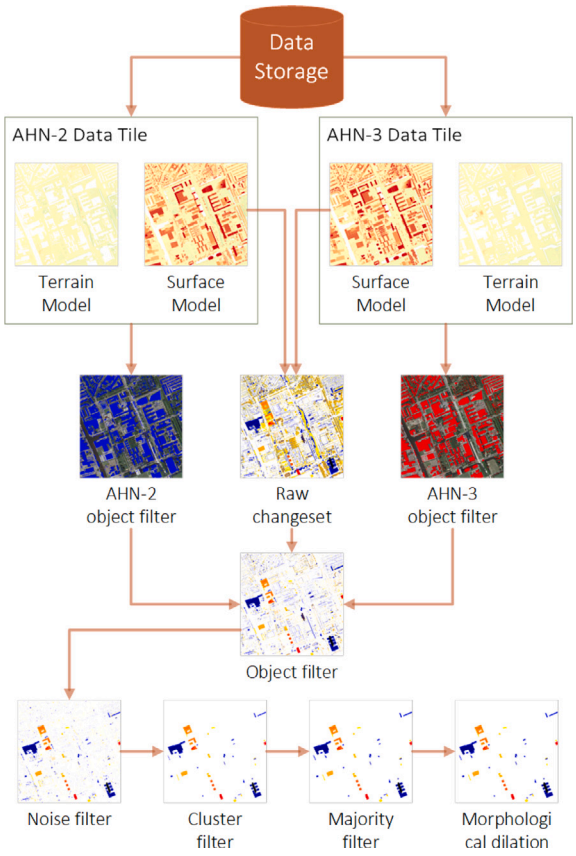


Fig. 9. The complete overview of the algorithm workflow.

- *lost*: the removed (demolished) volume of artificial content per hectare;
- *moved*: summation of *gained* and *lost*;
- *difference*: difference of *gained* and *lost*.

A sample visualization of the neighbourhood of Delft and its surroundings is shown in Fig. 10, the accumulated value of *difference*, used as the basis of the applied colour model.

3. Visualization of results

In order to enable a straightforward solution for interpreting and analysing the results, an interactive online visualization[6] was created, hence an average personal computer and a web browser is sufficient for the objective. Through this interface the users can either display the raw, building level output of the workflow or view a multi-scale vector map of the aggregated results by administrative units of municipalities, districts and neighbourhoods. Further functionalities consist of navigating, zooming, setting a base layer and selecting a location. The latter option provides information about the marked area, which depending on the type of overlay can be the exact altimetry difference or the accumulated values of the administrative unit — as described in Section 2.8. In Fig. 11 (a) the city centre of The Hague is presented with the altimetry elevation of the reconstructed central railway station building selected. In (b) the municipality aggregation view is displayed with details about Delft.

The dynamic visualization of such massive dataset is highly computation intensive and would require a powerful server cluster to provide a smooth user experience when browsing the website. Since user edit or other source of modification is not expected on the output of our workflow, the on-access recalculation of the visualization is superfluous, instead it should be rendered once and served statically. By pre-generating the web tiles for each accessible zoom level a tiled web map (also known as *slippy map*) can easily be constructed, implementing a similar concept like e.g. *Google Maps* or *OpenStreetMap* (Haklay and Weber, 2008). Aggregated views are more beneficial to be served as vector data, so the polygons can be annotated by the attributes of accumulated values.

Table 2
Validation results with TOP10NL used as reference data.

Location	Covered area	Ratio without dilation	Ratio with dilation[a]
Delft	62.50 km^2	87.81%	91.37%
The Hague	93.75 km^2	88.75%	92.48%
Amsterdam	218.75 km^2	77.63%	80.22%
Netherlands	41,865 km^2	66.99%	70.05%

[a]Reference building boundaries dilated by 1 m, as described in Section 4.

Table 3
Total and average absolute volume change in the validation areas for correctly detected buildings.

Location	Volume change	Average change
Delft	6.15 km^3	98,400 m^3/km^2
The Hague	11.78 km^3	125,653 m^3/km^2
Amsterdam	36.37 km^3	166,262 m^3/km^2
Netherlands	912.33 km^3	21,784 m^3/km^2

4. Verification and discussion

To assess the quality of our method defined in this section, a validation of results has been performed using the TOP10NL[7] (*Dutch Topographic Basemap*) dataset as reference. The nation-wide TOP10NL consists of detailed topographic features – including a building layer – of The Netherlands at scales between 1:5,000 and 1:25,000.

Since the AHN and TOP10NL projects are independent and do not share a common data acquisition plan, the best matching datasets were selected for verification. As the AHN-2 measurements were acquired between 2007–2011 (PDOK, 2013), while AHN-3 was collected between 2014–2019 (PDOK, 2015), 3 epochs of the TOPNL datasets from May 2009, November 2015 and September 2020 were utilized. A feasibility study regarding the accuracy of the TOP10NL dataset and utilization with a different dataset was previously researched by Van Altena et al. (2014).

The validation was evaluated based on a ratio of detected changes in the urban areas covered by the building layer of any selected epoch of the TOP10NL datasets, weighted with the absolute value of the altimetry difference for each location. To compensate for the inaccuracy of positioning between the datasets, which results in not properly overlapping buildings, the algorithm was also executed after applying a 1 m morphological dilation as a small tolerance to the building boundaries in TOP10NL.

The verification was showcased on 3 selected cities and its surroundings (Delft, The Hague and Amsterdam) as well as on the whole available AHN dataset. The results are shown in Table 2. The total and average volume change were also computed for these areas, as summarized in Table 3. Only changes for correctly detected buildings were accumulated as an absolute value, without separating constructions and demolitions.

As it could be expected, the proposed method produces better validation results for urban areas in general. In rural or agricultural

[6] Available at https://gis.inf.elte.hu/ahn/ahn_urban_nl.html

[7] http://www.kadaster.nl/-/top10nl

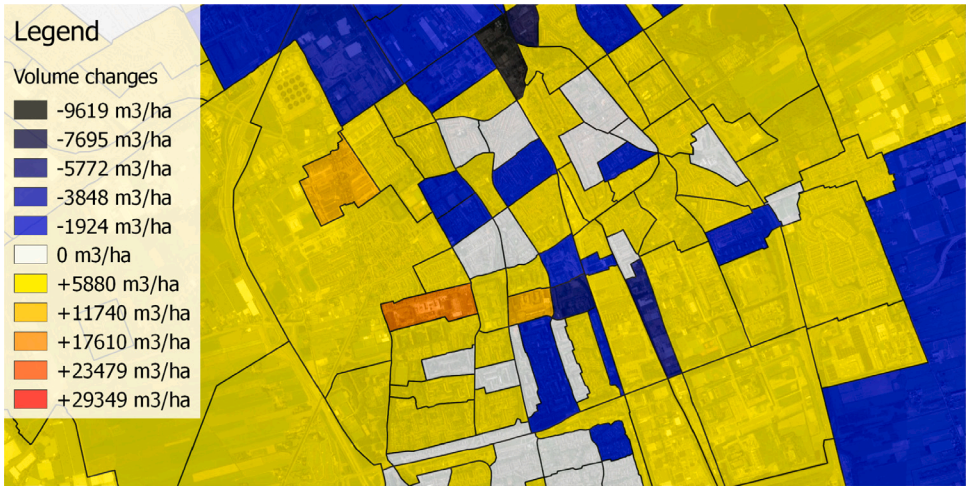
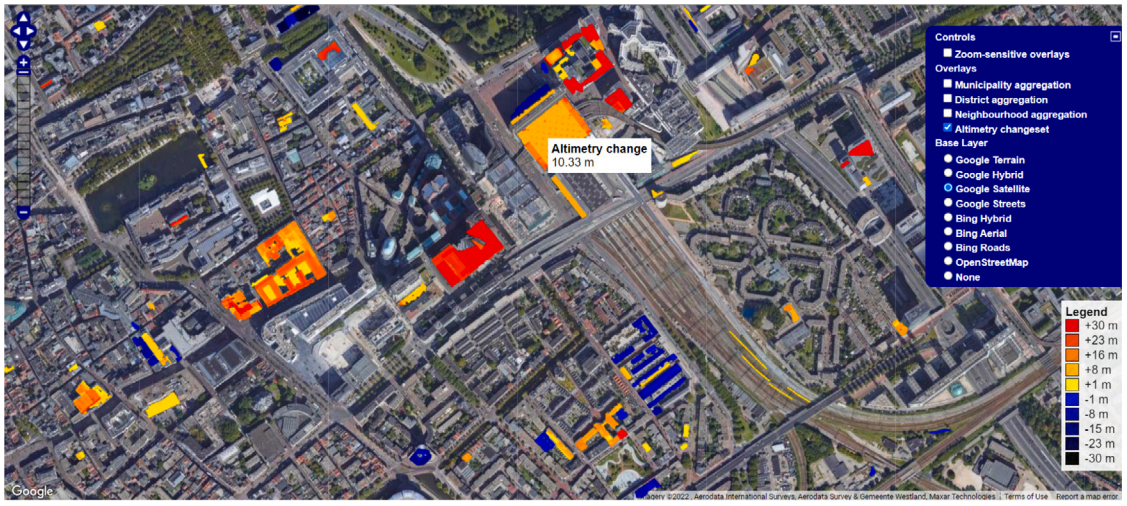
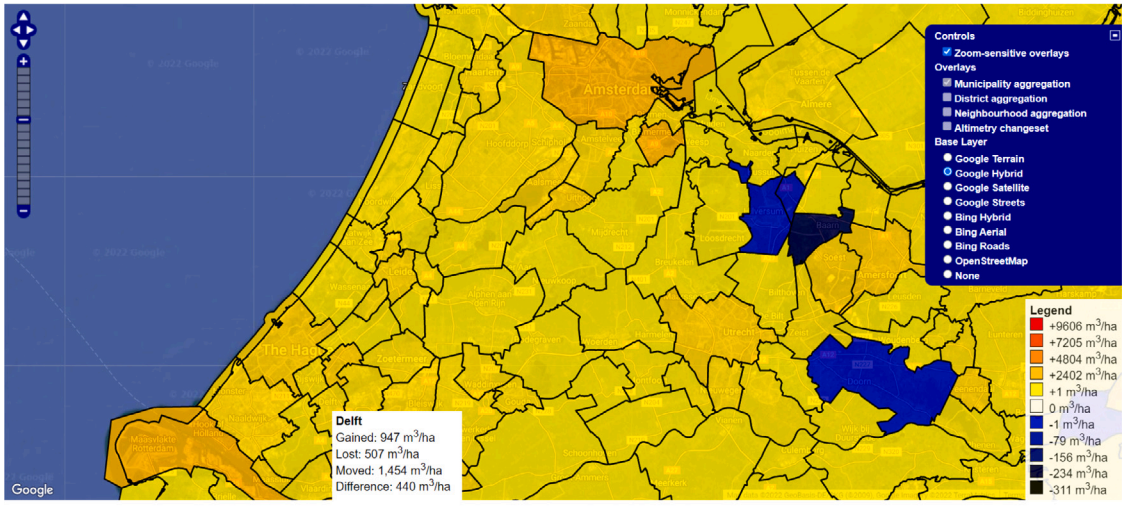


Fig. 10. Neighbourhood level aggregated overview of the city of Delft.



(a) Detailed, building-level altimetry changes in The Hague.



(b) Aggregation of changes on municipality level.

Fig. 11. Web interface of the visualization.

areas there are relatively few buildings, and in comparison there are more objects, which could be misdetected as buildings, such as small artificial hills. The main reasons of false positive detections in urban areas were the following (for visual examples, see Fig. 12):

- We detected changes not only in buildings, but also in other structures (e.g. bridges, overpasses), which are not marked in TOP10NL.
- While smaller moving objects, like cars were successfully filter by our method, larger objects like cargo ships or aircraft stationed at airports are detected.
- Beside these vehicles, artificially created hills, naturally moving dunes on the beach, etc. could also be detected by our approach.
- The AHN and TOP10NL datasets were not created at the same epoch, there can be multiple years of difference for some regions.
- In rare cases we have found existing buildings not registered in TOP10NL.

When comparing the 3 examined cities, Amsterdam produced worse results than Delft and The Hague. The main reason for this is that in the port area of the city, massive industry related changes were detected between the two AHN epochs examined, containing numerous false positive detections (e.g. large hills of debris).

5. Implementation and cloud computing

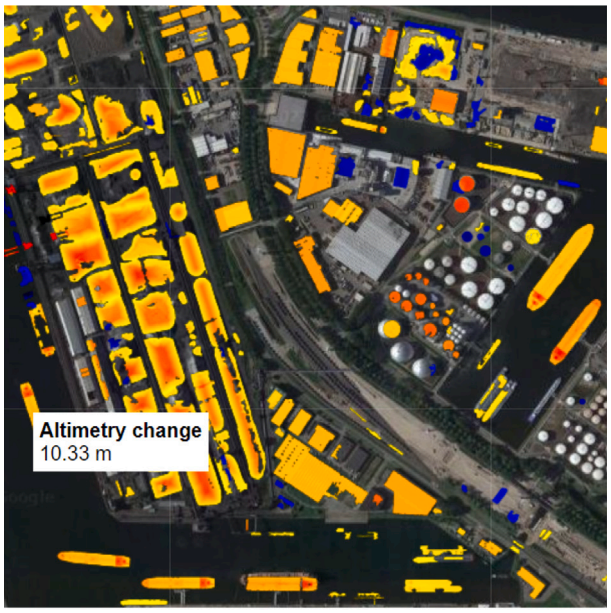
The implementation was carried out in standard C++, based on the open-source GDAL/OGR,[8] geospatial and geoprocessing software library for the input and output management of the spatial data. This decision was made on the facts that the GDAL/OGR library is actively developed and maintained, used by numerous well-known applications (*QGIS GRASS GIS*, *MapServer*, *ArcGIS*, etc.), provides a well-documented native C/C++ API, and the C++ language itself is among the most widely used programming languages both in general and also in the geospatial community. Our software is platform-independent and was compiled and tested under both Windows and Linux based operating systems.

The workflow was designed such that distributed computing can be facilitated. While each tile is processed sequentially, the parallelization was based on the already provided tiled partitioning of the dataset, processing multiple tiles in a distributed manner. The same method of distribution by dataset partitioning could also be applied on aggregating the results by administrative units as described in Section 2.8, on generating the static visualization presented in Section 3 and on validating the results with a reference dataset as showcased in Section 4.

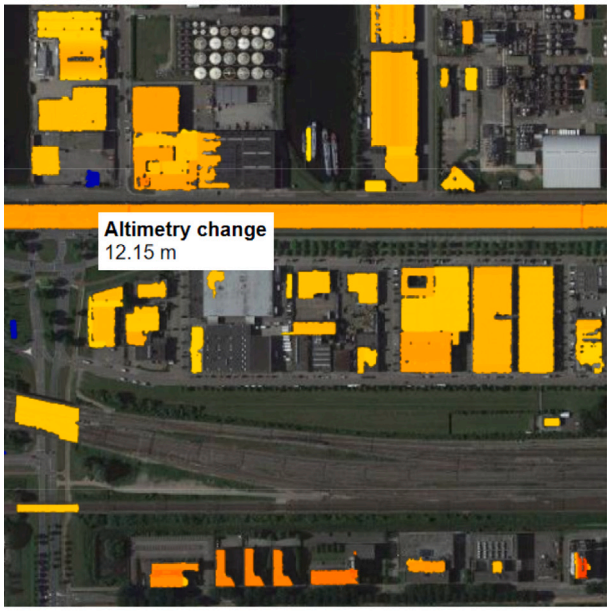
The input dataset consisted of the 1.368 tiles which were covered by both of the AHN-2 and ANH-3 measurements. For each tile $2 * 2 = 4$ raster grid files were provided, as for both AHN data acquisitions also the DSM and the DTM model had to be processed. Each DEM file allocated ca. 500 MB of disk space, the accumulated input data size was approximately 2.8 TB.

Our solution was tested in 3 hardware configurations: (*i*) on a personal notebook computer, (*ii*) on a high-performance computing environment and (*iii*) on a Hadoop cluster of low-budget desktop computers. These are discussed and compared in Sections 5.1–5.3, respectively. While ANH data collection has been conduced at intervals of several years so far, our focus here is on the efficiency of processing a nation-wide dataset. Motivation for this is that nation-wide point cloud data, such as AHN, is coming more widely available, (Virtanen et al., 2017), which has the consequence, that also methods that can handle this amount of data efficiently will become more in demand. In addition, for practical applications, it is important that intermediate large scale results can be produced efficiently, to facilitate parameter tuning.

[8] http://www.gdal.org/



(a) Port area in Amsterdam, notice hills of debris, contianer ships.



(b) Urban area in Amsterdam, notice overpass and railway bridge.

Fig. 12. Reasons of possible false positive change detections in buildings.

5.1. Desktop environment

The minimal, sequential execution environment of our program is a single core CPU with 1.5 GB RAM. The workflow was tested on a personal computer with a configuration as described in Table 4. While this notebook had 8 logical cores, we found that the optimal performance is reached with 3–4 parallel processes, due to the heavy I/O load of the workflow.

The average measured execution times with 3 distributed processes at each step is displayed in Table 5. It shows that the evaluation time of the complete workflow (including data read and write beside processing) took almost 2 days and required nearly 5 days in CPU time.

Table 4
Lenovo Y700 hardware configuration.

Component	Model	Specification
Processor	Intel® Core™ i7-6700HQ	4 physical, 8 logical cores, 2.60 GHz
Memory	Kingston® KVR21S15D	12 GB DDR4
Hard Disk	Western Digital® WD10SPCX	1 TB, 5400 RPM, 16 MB cache, SATA III (6.0 Gbps)

Table 5
Workflow execution time on a desktop computer with 3 parallel processes.

Step	Wall time (1368 tiles)	CPU time (1368 tiles)[a]	Wall time (3 tiles)[a]
Change detection	10.73 h	32.20 h	1.41 min
Aggregation	4.39 h	13.18 h	0.58 min
Visualization	22.74 h	68.22 h	2.99 min
Verification	9.41 h	28.24 h	1.24 min
Summary	47.28 h	141.84 h	6.22 min

[a]Estimated values based on the wall time of the whole dataset.

Table 6
SURFsara LISA node hardware specification.

Component	Model	Specification
Processor	Intel® Xeon® E5-2650L & E5-2650 v2	8 physical, 16 logical cores, 1.80 GHz & 2.60 GHz
Memory	N/A	32 GB/64 GB
Local storage[a]	N/A	750 GB/850 GB
Network	InfiBand FDR	56 Gbps bandwidth, 1.3 μs latency

[a]For temporary data only, the input dataset was available on a shared network drive.

Table 7
Workflow execution time on the SURFsara LISA cluster.

Nodes	Processes per node	Processes	Tiles per process	Overall time
30	5	150	9.12	2.38 h
22	7	154	8.90	3.23 h
15	10	150	9.12	4.90 h
10	15	150	9.12	4.80 h

5.2. High-performance computing environment

The development of the methodology and tuning its parameters demanded multiple executions and testing of the program, thus the extended waiting time of the desktop environment to receive and analyse the results was inappropriate. To accelerate the process, more powerful computing facilities had to be involved.

SURFsara[9] is a Dutch national collaborative ICT organization, providing computing facilities for education and research purposes. Their LISA[10] supercomputer was utilized to further scale the distributed evaluation of the workflow. The LISA cluster consists of 489 nodes, 7856 cores with a peak performance of 149 TFlops and uses the Debian Linux operating system. The configuration of a typical node is described in Table 6.

The parallelization and communication between the nodes – implemented through the platform-independent MPI[11] protocol – required a thoughtful design and benchmarking of scenarios, taking the I/O operation sensitivity of the computation into account to avoid a bottleneck on the data storage access or on the network bandwidth. In Table 7 we present multiple job configurations and their comprehensive execution time with approximately the same process count.

Table 8
Low budget desktop PC hardware configuration for Hadoop cluster.

Component	Specification
Nodes	1 master and 40 slave
Processor (per node)	2 physical, 4 logical cores, 1.20 GHz
Memory (per node)	4 GB DDR3
Storage	4 TB HDFS, SATA II (3.0 Gbps)
Network	100 Mbps bandwidth

As shown, the runtime could be easily reduced to 1 h or less. Such reduction was essential to efficiently develop a workflow based on large scale initial results. We can observe that while all nodes contained 16 logical cores, the excessive I/O management resulted in a setback of effectiveness in case too many processes per node were launched.[12]

5.3. Hadoop cluster of inexpensive computers

As an alternative for a supercomputer, we also experimented with a Hadoop[13] cluster of low budget desktop computers. We utilized Hadoop's MapReduce framework, wrapping our existing solution as a mapper. The I/O was managed by the Hadoop Streaming API.

The major benefits of the Hadoop cluster compared to a HPC is that it is assemblable from inexpensive components, easily scalable for increasing amount of input, the file distribution is handled by HDFS, and jobs are launched in a data-local manner (reducing stress on the network). The drawbacks are that all input data for a tile (4 raster DEMs) must be bundled into a single file for efficient reading from the HDFS, and it is also way more difficult to communicate between processes on separate nodes, as Hadoop was intentionally not designed for this. Our Hadoop cluster consisted of 41 inexpensive desktop computers, their technical specification is displayed in Table 8.

In this measurement, a single Hadoop job was executed on each node at once, due to their limited computing capacity and internal memory. The overall execution time was 13.14 h, which is in between the configuration of Sections 5.1 and 5.2, as expected.

6. Conclusions

Our research demonstrated that airborne LiDAR data and the preprocessed digital surface models deduced from them provide a suitable source for building recognition and their change detection of elevation. We presented a method which is applicable to process large datasets (multiple terabytes of data) with the efficient handling of computer

[9] http://surfsara.nl
[10] https://userinfo.surfsara.nl/systems/lisa/
[11] Message Passing Interface, standard available: http://mpi-forum.org/
[12] Note that while nodes were exclusively allocated for jobs on LISA, we had no control over network traffic of other jobs, which could also affect the measurements.
[13] https://hadoop.apache.org/

resources (restricted memory requirement) and can be easily executed on a distributed environment. The proposed solution was evaluated on 3 cities in the Netherlands (Delft, The Hague and Amsterdam) and on the complete territory of the country itself. As validation, a comparison with the building layer of the TOP10NL topographic dataset was performed, showing a 70% match generally and over 90% in urban areas. As a significant research contribution of our work, the evaluation and validation was carried out on a multiple magnitudes larger area (over 40.000 square kilometres) compared to most similar researches. To measure execution time, 3 different hardware configurations were tested, including two different distributed approaches: a HPC and a Hadoop cluster. With the LISA supercomputer, the processing times for the complete AHN-2 and AHN-3 datasets were only 2.38 h. An interactive web-based visualization was also developed and made publicly accessible as a further added value of our work.

Future work will include an additional building detection step in the method based solely on the DSM dataset, replacing the DTM based object detection described in Section 2.4. Therefore processing DTM files could be omitted, lowering the prerequisites of the proposed algorithm, meanwhile reducing the amount of data to be processed nearly to half. The filtering of vegetation near or in-between structures can also be revised with a more specific approach like achieved in Fekete and Cserep (2021), hence it would not filter out minor, but real human-made changes (e.g. slight modifications on rooftops). Further work could also address special cases of building detection on tile boundaries and their correct merging if deemed necessary.

Computer code availability

The prototype implementation of the algorithm was carried out in standard C++ as part of the *PointCloudTools* geospatial framework. Source code is available at https://github.com/mcserep/PointCloudTools and released under the BSD-3 licence. The project was tested to build and run on Windows 10 and Ubuntu Linux 18.04/20.04 LTS.

CRediT authorship contribution statement

Máté Cserép: Writing – original draft, Methodology, Investigation, Software, Validation, Visualization, Funding acquisition. **Roderik Lindenbergh:** Conceptualization, Methodology, Supervision, Writing – review & editing, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The AHN datasets used in the research are publicly available and can be found at http://www.ahn.nl/.

The final output of the change detection in the built-up area for the complete Netherlands was made publicly available at http://dx.doi.org/10.17632/yrxvhfj5jv.1, an open-source online data repository hosted at Mendeley Data (Cserép and Lindenbergh, 2022).

Acknowledgements

This work was sponsored by the Dutch Research Council (NWO) Physical Sciences for the use of supercomputer facilities under grant number 15454.

This work was supported by the Hungarian Government, co-financed by the European Social Fund (EFOP-3.6.3-VEKOP-16-2017-00001).

References

Badenko, V., Tammsaar, S., Beliaevskii, K., Fedotov, A., Vinogradov, K., 2019. Multithreading in laser scanning data processing. In: International Conference on Computational Science and Its Applications. Springer, pp. 289–305.

Boehm, J., 2014. File-centric organization of large LiDAR point clouds in a big data context. In: Workshop on Processing Large Geospatial Data Cardiff, Vol. 8. UK.

Butkiewicz, T., Chang, R., Wartell, Z., Ribarsky, W., 2008. Visual analysis and semantic exploration of urban LiDAR change detection. Comput. Graph. Forum 27 (3), 903–910.

Cserép, M., Lindenbergh, R., 2022. Detected changes of the built-up area comparing the Dutch AHN 2-3 datasets. Mendeley Data, http://dx.doi.org/10.17632/yrxvhfj5jv.1. (Accessed 23 November 2022).

De Smith, M.J., Goodchild, M.F., Longley, P., 2015. Geospatial Analysis: A Comprehensive Guide To Principles, Techniques and Software Tools, fifth ed. Troubador Publishing Ltd.

Discher, S., Richter, R., Trapp, M., Döllner, J., 2019. Service-oriented processing and analysis of massive point clouds in geoinformation management. In: Service-Oriented Mapping. Springer, pp. 43–61.

Du, S., Zhang, Y., Qin, R., Yang, Z., Zou, Z., Tang, Y., Fan, C., 2016. Building change detection using old aerial images and new LiDAR data. Remote Sens. 8 (12), 1030.

Dukai, B., Ledoux, H., Stoter, J., 2019. A multi-height LoD1 model of all buildings in the Netherlands. ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci. 4 (4/W8), 51–57.

Fekete, A., Cserep, M., 2021. Tree segmentation and change detection of large urban areas based on airborne LiDAR. Comput. Geosci. 156, 104900. http://dx.doi.org/10.1016/j.cageo.2021.104900.

Gonzalez, R.C., Woods, R.E., 2008. Digital Image Processing, third ed. Prentice Hall, Upper Saddle River, NJ, USA.

Haala, N., Brenner, C., 1999. Extraction of buildings and trees in urban environments. ISPRS J. Photogramm. Remote Sens. 54, 130–137.

Haklay, M., Weber, P., 2008. Openstreetmap: User-generated street maps. IEEE Pervasive Comput. 7 (4), 12–18.

Hegeman, J.W., Sardeshmukh, V.B., Sugumaran, R., Armstrong, M.P., 2014. Distributed LiDAR data processing in a high-memory cloud-computing environment. Ann. GIS 20 (4), 255–264. http://dx.doi.org/10.1080/19475683.2014.923046.

Ji, S., Shen, Y., Lu, M., Zhang, Y., 2019. Building instance change detection from large-scale aerial images using convolutional neural networks and simulated samples. Remote Sens. 11 (11), 1343.

Jian, X., Xiao, X., Chengfang, H., Zhizhong, Z., Zhaohui, W., Dengzhong, Z., 2015. A Hadoop-based algorithm of generating DEM grid from point cloud data. ISPRS Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci. XL-7/W3, 1209–1214. http://dx.doi.org/10.5194/isprsarchives-XL-7-W3-1209-2015.

León-Sánchez, C., Giannelli, D., Agugiaro, G., Stoter, J., 2021. Testing the new 3D bag dataset for energy demand estimation of residential buildings. Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. 46 (4/W1-2021).

Liu, J.G., Mason, P.J., 2013. Essential Image Processing and GIS for Remote Sensing. John Wiley & Sons.

Ma, R., 2005. DEM generation and building detection from lidar data. Photogramm. Eng. Remote Sens. 71 (7), 847–854. http://dx.doi.org/10.14358/PERS.71.7.847.

Matikainen, L., Hyyppä, J., Ahokas, E., Markelin, L., Kaartinen, H., 2010. Automatic detection of buildings and changes in buildings for updating of maps. Remote Sens. 2 (5), 1217–1248.

PDOK, 2013. In: van der Zon, N. (Ed.), Kwaliteitsdocument AHN2. Technical Report 1.3 Final, Dutch National Spatial Data Infrastructure, URL: http://www.ahn.nl/.

PDOK, 2015. Besteksvoorwaarden inwinning landsdekkende dataset AHN2014–2019. Technical Report 2.0 Final, Dutch National Spatial Data Infrastructure, URL: http://www.ahn.nl/.

Peters, R., Dukai, B., Vitalis, S., van Liempt, J., Stoter, J., 2022. Automated 3D reconstruction of LoD2 and LoD1 models for all 10 million buildings of the Netherlands. Photogramm. Eng. Remote Sens. 88 (3), 165–170. http://dx.doi.org/10.14358/PERS.21-00032R2.

Politz, F., Sester, M., 2022. Building change detection in airborne laser scanning and dense image matching point clouds using a residual neural network. Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. XLIII-B2-2022, 625–632. http://dx.doi.org/10.5194/isprs-archives-XLIII-B2-2022-625-2022.

Politz, F., Sester, M., Brenner, C., 2021. Building change detection of airborne laser scanning and dense image matching point clouds using height and class information. AGILE GIScience Ser. 2, 1–14.

Priestnall, G., Jaafar, J., Duncan, A., 2000. Extracting urban features from LiDAR digital surface models. Comput. Environ. Urban Syst. 24 (2), 65–78.

Richter, R., Kyprianidis, J.E., Döllner, J., 2013. Out-of-core GPU-based change detection in massive 3 D point clouds. Trans. GIS 17 (5), 724–741.

Scott, C.P., Arrowsmith, J.R., Nissen, E., Lajoie, L., Maruyama, T., Chiba, T., 2018. The M7 2016 Kumamoto, Japan, earthquake: 3-D deformation along the fault and within the damage zone constrained from differential lidar topography. J. Geophys. Res. 123 (7), 6138–6155.

Sun, L., Tang, Y., Zhang, L., 2017. Rural building detection in high-resolution imagery based on a two-stage CNN model. IEEE Geosci. Remote Sens. Lett. 14 (11), 1998–2002.

Swart, L., 2010. How the Up-to-date Height Model of the Netherlands (AHN) became a massive point data cloud. NCG KNAW 17.

Tomljenovic, I., Höfle, B., Tiede, D., Blaschke, T., 2015. Building extraction from airborne laser scanning data: An analysis of the state of the art. Remote Sens. 7 (4), 3826–3862.

Van Altena, V., Bakermans, J., Lentjes, P., Nijhuis, R., Post, M., Reuvers, M., Stoter, J., 2014. Generalisation of a 1:10k map from municipal data. In: 17th ICA Workshop on Generalisation and Multiple Representation. Vienna, Austria, 23 September 2014.

Van der Sande, C., Soudarissanane, S., Khoshelham, K., 2010. Assessment of relative accuracy of AHN-2 laser scanning data using planar features. Sensors 10 (9), 8198–8214.

Van Natijne, A., Lindenbergh, R., Hanssen, R., 2018. Massive linking of PS-InSAR deformations to a national airborne laser point cloud. Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. 42 (2), 1137–1144.

van Oosterom, P., Martinez-Rubi, O., Ivanova, M., Horhammer, M., Geringer, D., Ravada, S., Tijssen, T., Kodde, M., Gonçalves, R., 2015. Massive point cloud data management: Design, implementation and execution of a point cloud benchmark. Comput. Graph. 49, 92–125.

Virtanen, J.P., Kukko, A., Kaartinen, H., Jaakkola, A., Turppa, T., Hyyppä, H., Hyyppä, J., 2017. Nationwide point cloud—the future topographic core data. ISPRS Int. J. Geo-Inf. 6 (8), 243.

Vögtle, T., Steinle, E., 2004. Detection and recognition of changes in building geometry derived from multitemporal laserscanning data. Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. 35 (B2), 428–433.

Vu, T.T., Matsuoka, M., Yamazaki, F., 2004. LIDAR-based change detection of buildings in dense urban areas. In: Geoscience and Remote Sensing Symposium, 2004. IGARSS '04. Proceedings. 2004 IEEE International, Vol. 5. pp. 3413–3416. http://dx.doi.org/10.1109/IGARSS.2004.1370438.

Wang, Z., Schenk, T., 2000. Building extraction and reconstruction from lidar data. Int. Arch. Photogramm. Remote Sens. 33 (B3/2; PART 3), 958–964.

Weidner, U., 1997. Digital surface models for building extraction. In: Gruen, A., Baltsavias, E.P., Henricsson, O. (Eds.), Automatic Extraction of Man-Made Objects from Aerial and Space Images (II). Birkhäuser Basel, Basel, pp. 193–202. http://dx.doi.org/10.1007/978-3-0348-8906-3_19.

Williams, J.G., Anders, K., Winiwarter, L., Zahs, V., Höfle, B., 2021. Multi-directional change detection between point clouds. ISPRS J. Photogramm. Remote Sens. 172, 95–113.

Xie, M., Fu, K., Wu, Y., 2006. Building recognition and reconstruction from aerial imagery and LIDAR data. In: 2006 CIE International Conference on Radar. IEEE, pp. 1–4. http://dx.doi.org/10.1109/ICR.2006.343296.

Yang, C., Huang, Q., 2013. Spatial Cloud Computing: A Practical Approach. CRC Press, Boca Raton, FL.

Yang, C., Raskin, R., Goodchild, M., Gahegan, M., 2010. Geospatial cyberinfrastructure: Past, present and future. Comput. Environ. Urban Syst. 34 (4), 264–277, Geospatial Cyberinfrastructure.

Zhang, J., 2010. Multi-source remote sensing data fusion: status and trends. Int. J. Image Data Fusion 1 (1), 5–24. http://dx.doi.org/10.1080/19479830903561035.

Zhou, K., Lindenbergh, R., Gorte, B., Zlatanova, S., 2020. LiDAR-guided dense matching for detecting changes and updating of buildings in airborne LiDAR data. ISPRS J. Photogramm. Remote Sens. 162, 200–213.