

Locating PS-InSAR derived deformation using LiDAR point clouds

A.L. van Natijne



Locating PS-InSAR derived deformation using LiDAR point clouds

by

A.L. van Natijne

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Monday August 27, 2018 at 13:00.

Student number: 4089723
Project duration: November 2017 – August 2018
Thesis committee: Prof. dr. ir. R. F. Hanssen, TU Delft
Dr. R. C. Lindenbergh, TU Delft
Prof. dr. ir. P. J. M. van Oosterom, TU Delft

An electronic version of this thesis is available at <http://repository.tudelft.nl/>.

Abstract

Built mainly on soft soil, the Netherlands is at high risk for the effects of deformation. Interferometric Synthetic Aperture Radar (InSAR) is successfully used to monitor the deformation trends at millimetre level. Unfortunately the InSAR deformation trends suffer from poor geolocation estimates, limiting the ability to link deformation behavior to objects, such as buildings, streets or bridges. A nationwide, high resolution, airborne LiDAR point cloud is available in the Netherlands. Although the LiDAR point cloud itself is unsuitable for deformation estimates, linking the InSAR location to the geometries outlined by the LiDAR point cloud can improve the geolocation estimates of the InSAR trends. In this thesis three methods are shown to link deformation estimates to the LiDAR point cloud or reconstructed features thereof. As a test, 3.1 million TerraSAR-X InSAR Persistent Scatterers are linked to 3 billion LiDAR points, covering the city of Delft and surroundings. 85% of the scatterers could be linked to the point cloud. Furthermore an outlook at the possibilities of an implementation on a national scale using Sentinel 1 data is given.

Contents

1	Introduction	1
1.1	Background	1
1.2	Objectives and contribution of this work	3
1.3	Research questions	3
1.4	Outline	4
1.5	Case studies	4
2	Background	5
2.1	InSAR	5
2.2	LiDAR	11
2.3	Related work	12
2.4	Remaining challenges	15
3	Actueel Hoogtebestand Nederland	17
3.1	Available data	18
3.2	Error model	20
3.3	Data preparation	23
3.4	Data volume	29
4	Methodology	31
4.1	Common visualization of InSAR and LiDAR data	32
4.2	First Nearest Neighbor linking	32
4.3	Linking to a single surface	34
4.4	Linking to multiple surfaces; dihedral and trihedral geometries	35
4.5	Quality attributes and indicators	36
5	Results & validation	39
5.1	Web viewer	39
5.2	Matching	40
5.3	Regional trends	47
5.4	Discussion	47
6	Conclusions	51
6.1	Conclusions	51
6.2	Recommendations	53
	Acknowledgements	55
	Bibliography	57
A	Additional figures	61
B	Data preparation commands	67
C	ISPRS Conference Paper	71
D	NAC Conference Poster	81

Introduction

1.1. Background

Subsidence in the Netherlands, and deformation in general, is threatening building integrity, damaging infrastructure and lowering the land with respect to sea-level. Deformation occurs at all scales, from single pillar failure at the 't Loon shopping mall in 2011 [Chang and Hanssen, 2014] to complete regions suffering from effects such as subsidence relative to the water table [Boersma, 2015].

Deformation processes include: peat compaction in the west and north of the country [Boersma, 2015]; induced subsidence and seismic activity due to gas extraction in the Groningen area [Ketelaar et al., 2006]; land uplift and cavity formation due to the flooding of old mine works in Limburg [Bekendam and Pottgens, 1995].

Interferometric Synthetic Aperture Radar (InSAR) is used to monitor deformation from satellites. The monitoring technique used in this thesis is based on *Persistent Scatterers* (PS-InSAR), points or objects that show consistent behavior over longer periods of time (years). For these points the deformation time series is compiled from subsequent radar acquisitions. Based on this time series a deformation velocity is estimated, with millimeter per year accuracy.

Unfortunately the source of the deformation signal is, in general, less accurately known: geolocation estimates of PS-InSAR are known with meters precision at best, depending on the sensor. This process of estimating the PS position, the geolocation, is geocoding. After geocoding the results can be mapped to an irregular map of deformation measurements, as shown in figure 1.1. This map shows in a 2D representation of individual deformation trends.

Although this will allow for deformation estimates up to street level [Ketelaar et al., 2006], the deformation signal can not be attributed to a single geometric feature. The location of this scatterer is of key importance in the understanding and interpretation of the deformation behavior: a subsiding garden house or street will require different precautions than a subsiding bridge pillar.

LiDAR (Light Detection and Ranging), is an active remote sensing technique where distance is measured by illuminating the target with a laser light and is also known as laser scanning or 3D scanning, [Baek et al., 2017]. Either from airborne, mobile or terrestrial platforms it provides high density point clouds.

The LiDAR point density is high enough to reconstruct (parts of) the real-world geometry. Matching the LiDAR and PS-InSAR data sets will allow improvement of the location and enrichment of the spatially sparse PS-InSAR measurements with geometric information derived from the LiDAR point cloud. An example of a LiDAR data set can be seen in figure 1.2. A comparison of the point density of Persistent Scatterers from TerraSAR-X to the density of Actueel Hoogtebestand Nederland over the centre of Delft¹ shows there are on average 20 points per square meter in AHN, but only one Persistent Scatterer

¹AHN tile 37EN1_10, figure A.1.

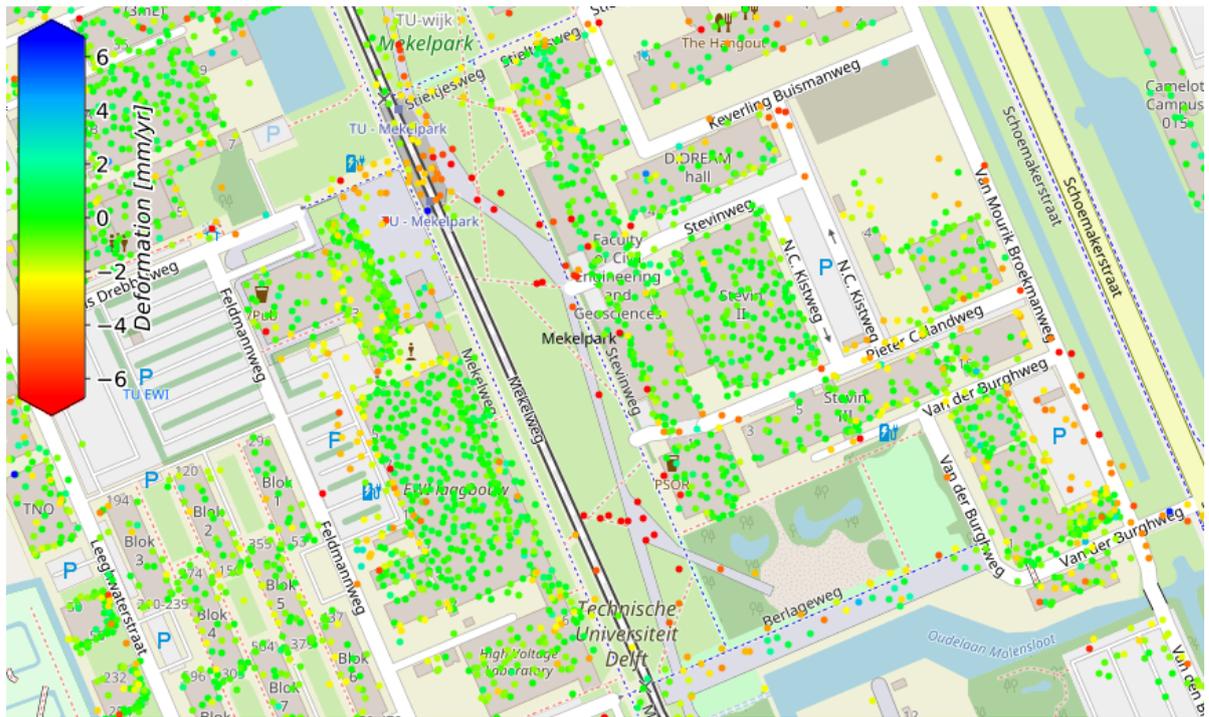


Figure 1.1: An example of PS-InSAR points on the TU Delft campus, colored by the linear trend in their deformation behavior, based on two years of TerraSAR-X data. (Background: OpenStreetMap)



Figure 1.2: The colored AHN3 point cloud of the TU Delft campus, shown in Potree. The area is similar to figure 1.1, the view is towards the north west. The Potree viewer enables the user to fly through the point cloud in a web browser.

every 50 square meters (0.02 pts. per sq. m).

1.2. Objectives and contribution of this work

The objective of this work is to find the location of the dominant scatterer in a LiDAR point cloud to improve the estimated origin of the deformation signal. Such combination will allow for linking radar scattering to geometric features in a scene.

An example of a traditional (2D) InSAR deformation map can be seen in figure 1.1. Van Natijne et al. [2018] enabled the visual integration of both data sets in a single 3D web viewer. This 3D visualization already aided the interpretation of the PS-InSAR signal, over the classical 2D mapped interpretation. Explicit geometric 3D linking of PS-InSAR geolocation estimates to 3D LiDAR point cloud data will also give a quantifiable improvement of the geolocation.

Geometric (3D) linking of the data sets could provide:

- Linking of the deformation signal to specific parts of the infrastructure, for maintenance planning and early warning.
- Assessment of differential deformation, as multiple deformation signals can be *attached* to the same building geometry.
- Detection and mitigation of (regional) errors and trends in the radar processing.
- Improved interpretation of the data.

This thesis is dedicated to the creation of this missing link between those data sets, by truly integrating both data sources. Given the (free) availability of a nationwide airborne LiDAR data set (Actueel Hoogtebestand Nederland), the Netherlands form a perfect test bed for the integration of radar data sets with point clouds. Furthermore the existing online point cloud viewer, based on Potree [Schütz, 2018], can be extended to visualize this link between the laser point cloud and radar data.

Limitations and assumptions

In this project only Persistent Scatterers (PS) are considered, Distributed Scatterers (DS) are explicitly not discussed here. Scattering of Distributed Scatterers is governed by an ensemble of weak scatterers. In the context of this work a single (strong) scatterer is expected to dominate the response (scattering behavior). Only then the response can be pin-pointed to a single (geometric) feature.

It has to be noted that in urban environments scattering behavior might be complex and persistent scattering behavior might be the result of the larger geometry, [Samiei Esfahany, 2017]. Modelling of the surroundings, for example based on LiDAR information, may allow for analysis of such complex scatterers, [Yang et al., 2017].

If the assumption is made that scattering behavior of PS can be attributed to a single feature one may use data sets other than LiDAR to analyze those features. In this way the properties of the radar measurement (e.g. amplitude variations, polarimetry) may be extended with properties of the joined data sources. These sources include all sources that may add information or attributes to the radar measurements, such as a data set with addresses or building outlines. In this study the focus will be only on joining LiDAR point clouds.

1.3. Research questions

The primary objective is improving the location of PS-InSAR scatterers by combining the positioned PS data with high resolution point clouds. These point clouds can be of any origin, but the focus is on LiDAR data.

By using data redundancy the uncertainty can be estimated and reduced. Information concealed in the properties of both data sets (geometry, deformation behavior, ...) can be used to combine measure-

ments of similar properties. Preferably developed method is scaleable, which allows the technique to be applied at regional scale, providing useful data to the operator in the field.

The leads to the principal question:

How to improve geolocation of InSAR point scatterers using detailed 3D point clouds?

This question is subdivided into the following sub-questions:

1. What aspects of PS-InSAR deformation assessment can be improved by linking to 3D surface geometry?
2. What statistical model applies to the data, and how was it defined?
 - Is the statistical model valid?
 - Is the data suitable for the planned application?
3. How are PS-InSAR and LiDAR point clouds best spatially joined?
 - How to represent the point cloud in segments of similar features?
 - How to join a persistent scatterer to the surface outlined by LiDAR point cloud data?
 - How to represent the uncertainties of this match?
4. What is the quality of the alignment between the data sets and can the alignment be improved?
 - Is there a constant offset or another bias between data sets? Is this offset local or regional?
 - Can this offset be mitigated, based on the spatial join?
5. How to scale the matching process to regional or national scale?

1.4. Outline

A review of previous work and an overview of the data available to this study is given in chapter 2. As the *Actueel Hoogtebestand Nederland* (LiDAR data set) has a primary role in this combination a separate chapter is spent on the properties, preparation and validation of this data set (chapter 3). Chapter 4 and 5 form an unity, discussing the method applied, quality indicators for the provided result and the results of the method over the study areas. A conclusion is drawn in chapter 6.1. Additional figures are provided in the appendix.

1.5. Case studies

Two main case studies were planned based on their spatial extent and sensor type:

1. Delft and surroundings

A combination of TerraSAR-X data, Actueel Hoogtebestand Nederland 3 (Airborne LiDAR) and Mobile Laser Scanner data. The TerraSAR-X data covers the city of Delft, parts of The Hague and neighboring fields and villages. Mobile Laser Scanning data is available of the perimeter of the Delft University of Technology campus, while Actueel Hoogtebestand Nederland is available for the whole of the Netherlands.

2. The Netherlands

A nationwide coverage, based on Sentinel 1 PS-InSAR data. Combined with AHN3 where available and AHN2 otherwise.

2

Background

This chapter is an overview of previous work and the required enhancements to achieve the aforementioned goal of improving the estimated position of the PS-InSAR deformation signal. The principles of InSAR monitoring will be discussed, along with the data available to this study. The principles of LiDAR data acquisition will be discussed in this chapter, the LiDAR data used is introduced in chapter 3. This chapter is concluded with an overview of related works and the challenges remaining to be solved.

2.1. InSAR

InSAR (Interferometric Synthetic Aperture Radar) and PS-InSAR (Persistent Scatterer InSAR) are established deformation monitoring techniques, providing centimeter to millimeter resolution deformation estimates. In this section the acquisition method, processing and resulting data, used in this thesis, will be introduced.

Principles

The principles of InSAR are covered in detail in many sources. This section is a summary of some of those sources, sources that cover the mathematical or physical background of the matter discussed. For an in depth introduction the reader is referred to: Hanssen [2001], Rees [2001], van Leijen [2014] and Samiei Esfahany [2017].

A radar instrument, mounted on a satellite, is operated as it flies over the earth below. Each measurement starts with the emission of a (short) radio pulse, that is received with a slight delay - due to the distance between the satellite and the earth. As the instrument is mounted at an angle ('side-looking') longer delays correspond to a longer distance, on flat terrain this equals a distance further to the side, [Rees, 2001].

The azimuth (direction of flight) resolution of such 'Real Aperture Radar' (RAR) or 'side-looking radar' (SLR) system is dependent on the height and the size of the antenna. The range (perpendicular to the direction of flight) resolution is limited by the length of the pulses transmitted. Due to practical limitations (the minimum orbit height, maximum practical size of the antenna, the high energy requirements of short pulses, etc.) possibilities for improvement are limited. The altitude dependence of the azimuth resolution can be circumvented by the use of 'Synthetic Aperture Radar' (SAR). Instead of a single acquisition multiple acquisitions are used to form a single image, artificially enlarging the antenna to all integrated acquisitions. Range limitations are circumvented by *chirping*, shortening of the transmitted pulse by means of signal processing, [Rees, 2001]. Unlike in surveillance radar systems the time difference between transmission and receiving the pulse as such is not of interest. Instead, after processing, only the phase and amplitude of the (complex) signal are stored for each *range cell* (which is effectively a time bin).

As active Remote Sensing technique, radar is dependent on this return of the emitted signal from the object of interest. The recorded phase and amplitude are the result of the reflections on all objects within the ground footprint of the resolution cell and noise. The strength of the reflection is dependent on the physical and electrical properties of the objects met and on the properties of the signal itself such as wavelength and incidence angle, [van Leijen, 2014].

Two radar images, a *master* and a *slave*, acquired at different times, can be combined into an interferogram. The two (radar) images, taken from approximately the same position and with the same viewing geometry are aligned first. The interferogram is created by the multiplication of the complex master image with the complex conjugate of the slave image. The interferometric phase is a summation of the (residual) errors and the interferometric (deformation) signal, [van Leijen, 2014].

Especially man-made structures may have backscattering characteristics that are consistent over longer periods of time, where the signal is dominated by a single scatterer or an ensemble of scatterers, consistent between acquisitions. Due to their consistent behavior these points are referred to as Persistent Scatterers (PS), [van Leijen, 2014]. This property allows extraction of the same feature in consecutive radar images, and the compilation of a time series of its movement. Millimeter per year accuracy can be achieved in deformation trend estimation based on those measurements, [van Leijen, 2014].

PS-InSAR monitoring is opportunistic: the presence of persistent scatterers can not be controlled, except by the placement of artificial reflectors. Frequent satellite radar acquisitions, for example the 6 day repeat frequency of Sentinel 1 allow for more and more detailed analysis of subsidence behavior. Higher resolutions, for example of TerraSAR-X (maximum 3×3 m over large areas, up to 25 cm for small areas) [Werninghaus and Buckreuss, 2010], allow smaller, weaker, scatterers to be detected. Combined with increased geolocation accuracy and precision this enables the monitoring at building or feature scale, [Gernhardt et al., 2010, Werninghaus and Buckreuss, 2010].

Geolocation

Geocoding, the process of mapping the position of the scatterer from the radar geometry (azimuth, range and cross-range) to a 3D Terrestrial Reference Frame, [Dheenathayalan et al., 2016]. The positioning precision is first estimated in radar coordinates, and then transformed to a (local) Cartesian coordinate system. The standard deviation for range (σ_r) and azimuth (σ_a) follow directly from the properties of the radar, based on formal error propagation of radar system properties and precision of: timing, (sub-)pixel positioning, atmospheric effects, solid earth tides and plate tectonics, [Schubert et al., 2015]. The precision in range and azimuth are assumed to be uncorrelated, although, timing (errors) for example, are shared error sources, [Dheenathayalan et al., 2016].

The error in the cross-range direction can only be estimated after solving for the cross-range, which exploits the baseline between acquisitions that causes small differences in the look angle and range. This cross-range precision depends on six factors: sub-pixel positioning; temporal (phase) stability; phase unwrapping; the number of images; the baseline distribution and phase noise. Only the first two are considered by Dheenathayalan et al. [2016]: sub-pixel positioning and temporal phase stability. Unwrapping is assumed to be error free. Again, the covariance between range, azimuth and cross-range is assumed to be negligible, [Dheenathayalan et al., 2016].

Then the estimated precision for all points be described in radar coordinates as a diagonal matrix, of variances only:

$$\mathbf{Q}_{RAC} = \begin{bmatrix} \sigma_r^2 & & \\ & \sigma_a^2 & \\ & & \sigma_{cr}^2 \end{bmatrix}, \quad (2.1)$$

where \mathbf{Q}_{RAC} is the diagonal matrix of the variance of the range (σ_r^2), azimuth (σ_a^2) and cross-range (σ_{cr}^2) geolocation estimate.

The viewing geometry of the radar satellite is expressed as a rotation matrix relative to world coordinates (RDNAF, the Dutch national coordinate system). A rotation matrix \mathbf{R} is found using the S-transform, based on the transformation of the points from radar coordinates to the terrestrial reference frame, [Dheenathayalan et al., 2016]. The quality model of the radar geolocation (\mathbf{Q}_{RAC} , equation 2.1) can

be transformed to the covariance matrix, \mathbf{Q}_{ENU} , relative to world coordinates (in *East*, *North* and *Up*), using this rotation matrix, \mathbf{R} :

$$\mathbf{Q}_{ENU} = \mathbf{R}\mathbf{Q}_{RAC}\mathbf{R}^T = \mathbf{R} \begin{bmatrix} \sigma_r^2 & & \\ & \sigma_a^2 & \\ & & \sigma_{cr}^2 \end{bmatrix} \mathbf{R}^T = \begin{bmatrix} \sigma_e^2 & \sigma_{en}^2 & \sigma_{eu}^2 \\ \sigma_{en}^2 & \sigma_n^2 & \sigma_{nu}^2 \\ \sigma_{eu}^2 & \sigma_{nu}^2 & \sigma_u^2 \end{bmatrix}. \quad (2.2)$$

Represented by the σ_{en}^2 , σ_{eu}^2 and σ_{nu}^2 elements, is the covariance between the coordinates introduced by the transformation to world coordinates.

Important geometries

Corner reflectors have strong reflecting capabilities over a wide angular range (figure 2.1). In such a corner reflector the geometric configuration creates a strong backscatter of the incoming radar signal, [Richards et al., 2010]. Due to their insensitivity to the angle of incidence corner reflectors may show up in radar data, even with changes to radar geometry, when scattering from planes is lost. Their strong scattering characteristics make them major features in InSAR data, their consistent appearance important features in PS-InSAR.

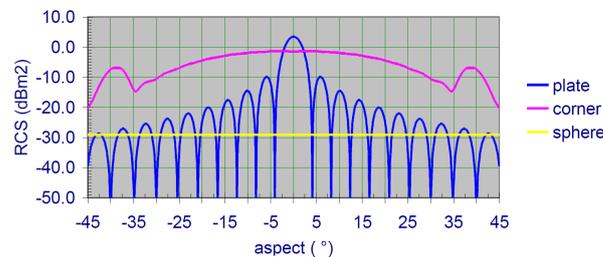


Figure 2.1: Radar cross section (RCS) as function of angle of incidence (aspect), for corner (trihedral) reflectors, planes and spheres, [Yarovoy et al., 2017]. The RCS of a plane fluctuates strongly with slight variations in the incidence angle, while corner reflectors are relatively insensitive to such changes. A sphere has an equal surface, regardless of the incidence angle.

Nevertheless the importance of corner reflectors is doubted by Perissin and Ferretti [2007] stating that only 13% of radar scatterers are formed by dihedral scattering and 8% by trihedral scattering. More than 50% of the scattering is due to monohedral scattering on roofs, [Schunert and Soergel, 2016].

PS-InSAR preparation

PS-InSAR are extracted from time series of SAR images. Time series allow for the separation of the deformation signal from noise sources, such as atmospheric distortions. First a series of stable features has to be found, those features will be assumed stable in the separation of the various signals. This selection could, be based on *Normalized Amplitude Dispersion*, for example, where the amplitude is used to estimate the stability of the phase, [van Leijen, 2014]. The phase itself is unlikely to be stable, even for stable scatterers, due to aforementioned noise sources. After extraction and modelling of the noise sources their contribution can be corrected for and the deformation signal extracted from the images, [van Leijen, 2014].

Instead of evaluating the phase difference for all measurements in the radar image, Persistent Scatterer Interferometry evaluates the signal only at selected scatterers, thereby creating a time series at those scatterers. An example of such point and its time series is shown in figure 2.2.

The data used in this thesis was acquired by TerraSAR-X, processing from raw data to persistent scatterers was done by Delft' firm *SkyGeo*.

Detection of underlying deformation processes

Deformation analysis can be conducted at a regional or national scale, such as shown by Cuenca et al. [2011] (figure 2.3). It requires only low spatial resolution to reveal large scale trends in surface

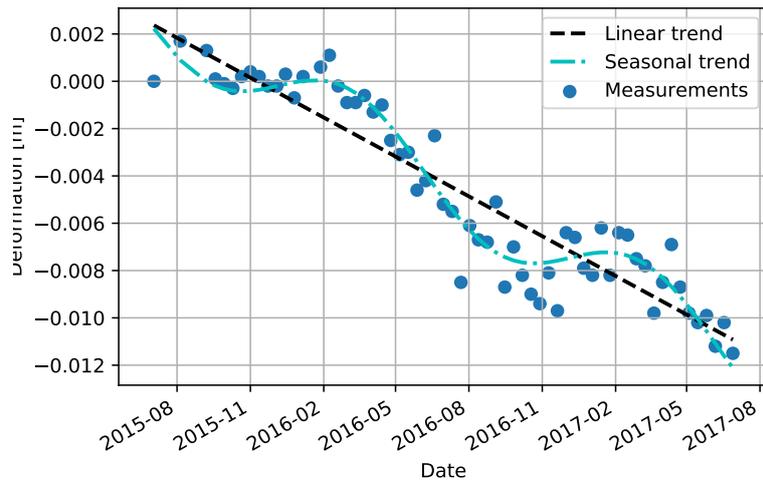


Figure 2.2: An example of a PS-InSAR time series (TerraSAR-X) of an artwork at the Mekelpark in front of the CEG faculty building. The subsidence signal is measured as an increase in the (slant-)range over time. Superimposed are a linear trend, and a linear trend with a seasonal variation.

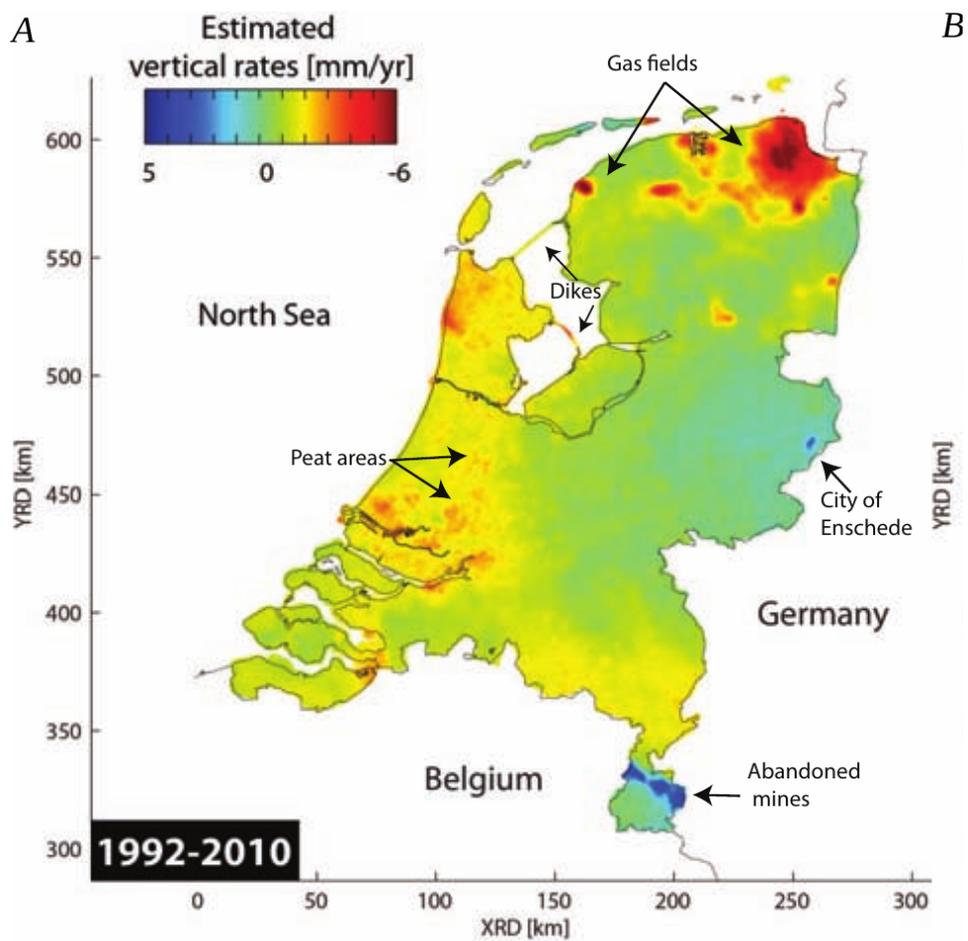


Figure 2.3: Estimated vertical rates for the period 1992-2010, showing large scale deformation in the Netherlands. [Cuenca et al., 2011]

deformation, but such an analysis does not differentiate between shallow and deep subsidence or other effects with influence on the scattering characteristics.

The origin of the PS-InSAR signal is not predefined. It could be on a building, rigidly fixed to a stable layer; subsiding with the shallow subsurface; thermal effects (e.g. on steel structures); influence of groundwater or influence of multipath effects with surrounding features. This problem is addressed by Ketelaar and Hanssen [2003], who suggest the use of polarization to differentiate between even and odd bounce scatterers and the linking to a 3D city model to determine if a scatterer is on a building or not.

Different scattering regimes react differently to these subsidence regimes, and understanding of the origin of the signal will help in understanding and modelling the deformation signal. An illustration of this combination of subsidence regimes and their effect on the radar signal is given in figure 2.4. A recent, innovative, approach is to use techniques from deep learning to automatically identify deformation regimes without prior knowledge of their physical background, [e.g. van de Kerkhof et al., 2017].

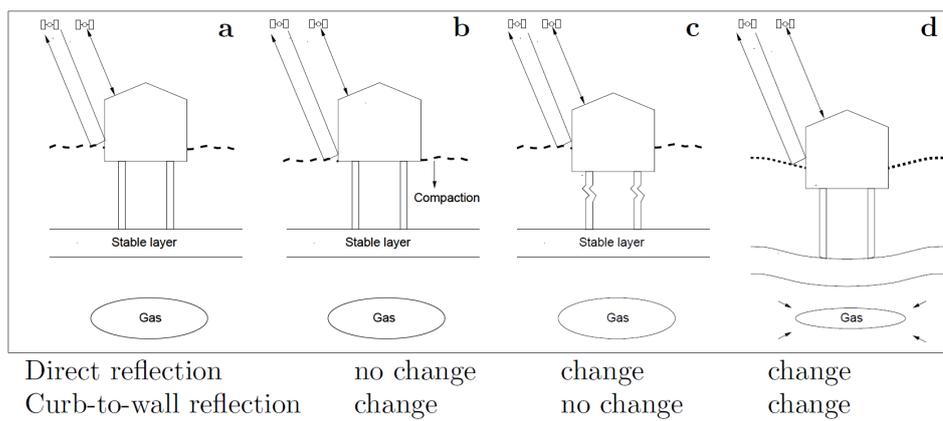


Figure 2.4: "Deformation regimes and their effect on single and double-bounce reflections: (a) reference situation; (b) shallow mass displacement (compaction), affecting only the double-bounce reflection; (c) structural instabilities (foundation), affecting only the single-bounce reflection; (d) deep mass displacement (gas extraction), affecting both the single-bounce and the double-bounce reflection." [Ketelaar and Hanssen, 2003]

To distinguish between different deformation processes understanding of the source of the radar signal is required. This solves two questions: what object does the scatter originate from and how does it behave. Such analysis is based on either the radar data itself (e.g. polarization) or on the addition of auxiliary data, for example a point cloud or 3D building model, [Ketelaar and Hanssen, 2003].

Dheenathayalan and Hanssen [2013] showed that a reasonable distinction could be made between ground and building pixels by separating them based on the estimated height, based on the estimation of a ground surface from other points - when no such surface is yet available. Another approach is to model the radar signal using ray-tracing to reconstruct the source of the scattering signal, [Yang et al., 2017].

TerraSAR-X

The German (DLR) TerraSAR-X Synthetic Aperture Radar (SAR) mission, was launched in 2007 and delivers high resolution radar imagery ever since. As a Public-Private Partnership the mission combines scientific and commercial interests of X-band, land oriented monitoring applications. With an 11-day repeat cycle and resolutions up to 3×3 m in StripMap mode the mission can provide deformation data of high spatial and temporal resolution over large areas, [Werninghaus and Buckreuss, 2010].

The PS-InSAR time series in this thesis, extracted from radar data of two TerraSAR-X orbits was processed and provided by SkyGeo. The data consists of both ascending and descending orbits over the same region and covers Delft, surrounding fields, Rijswijk and parts of The Hague (the Netherlands). The extends are shown in figure 2.5, or in relation to the Netherlands as a whole in red on figure 3.1.

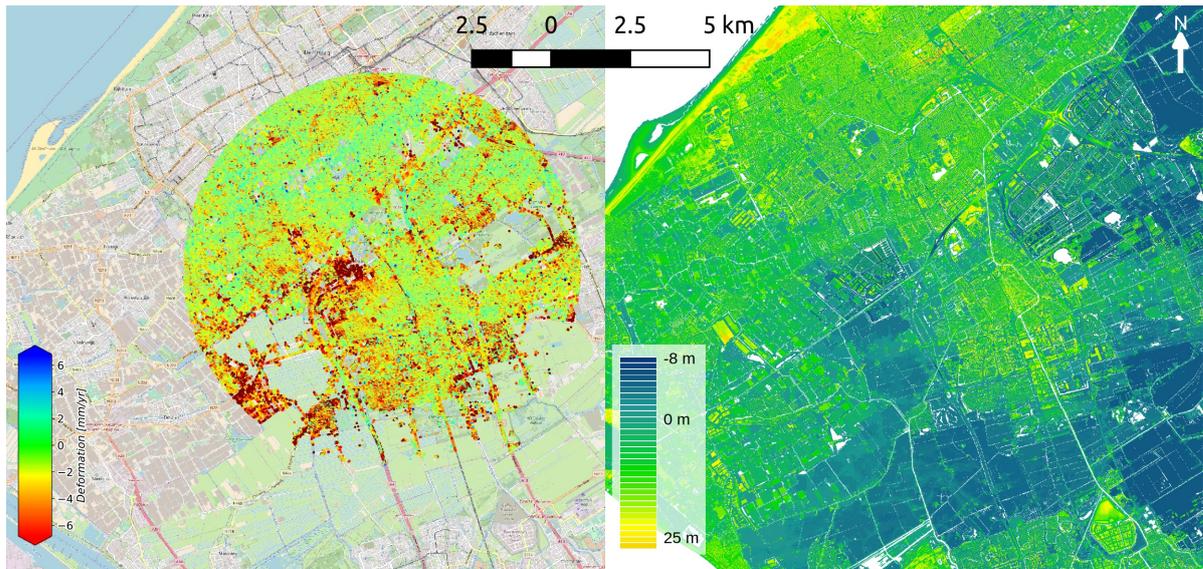


Figure 2.5: Extend of the TerraSAR-X data set available to this study. Left: PS-InSAR velocity estimates from both ascending and descending orbits. The city of Delft is hidden below the points. The Hague can be seen at the top and the northern part of Rotterdam at the bottom. (Background: OpenStreetMap) Right: digital surface model extracted from AHN3 for the same region.

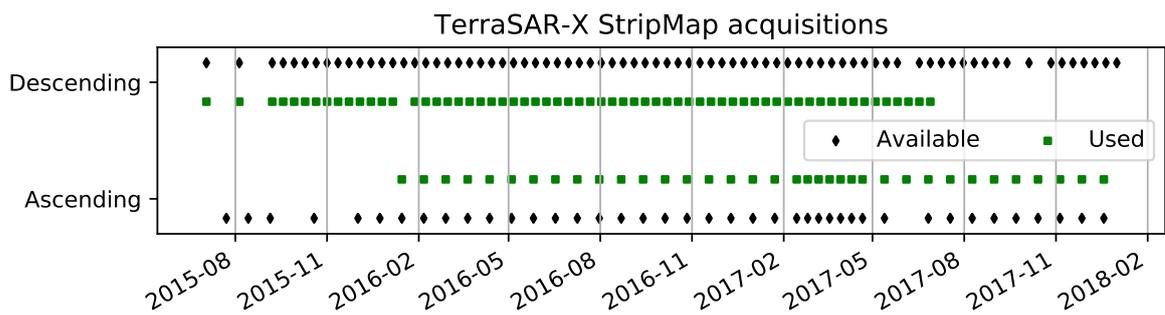


Figure 2.6: Availability of TerraSAR-X StripMap acquisitions over Delft between July 2015 and January 2018, [Airbus D&S, 2014]. Acquisitions used in the time series production are marked in green.

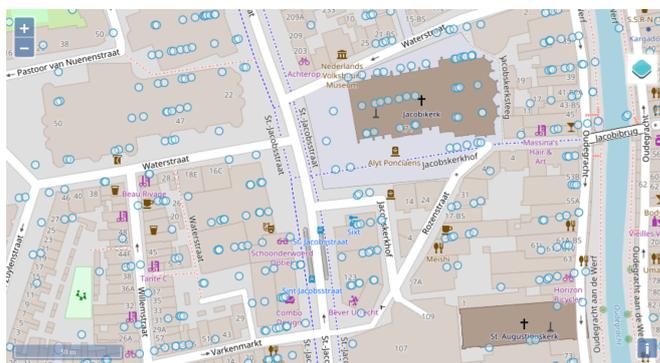


Figure 2.7: Alignment patterns as can be seen in the Sentinel 1 persistent scatterer geolocation estimates over Utrecht, due to lack of sub-pixel positioning in the range and azimuth direction. The small variations visible are due to the variation in height, mapped on the range direction. (Background: OpenStreetMap)



Figure 2.8: Approximate coverage of the Sentinel 1 data set that was available to this study. (Background: PDOK [2018])

A total of 118 radar images were acquired by TerraSAR-X between 2015-07-03 and 2018-01-01, of which 98 were used: 36 ascending and 62 from a descending orbit. Their temporal distribution can be seen in figure 2.6. For each orbit roughly two years of data was included in the processing. For unknown reasons acquisitions of 2017-06-04 (ascending) and 2017-06-06 (descending) are no longer available in the archive.

SAR Interferometry (InSAR) was applied to extract deformation signals by analyzing the time series of phase changes. Pixels that can be tracked consistently over multiple acquisitions are Persistent Scatterers (PS). These *coherent* pixels denote the deformation behavior of the same scatterer over longer periods of time [Hanssen, 2001]. A linear deformation trend (in time, $\frac{\text{mm}}{\text{yr}}$) is estimated for those points.

Data of the descending orbit contains 1.7 million PS-InSAR points, the ascending data contains 1.4 million points. For all points a geolocation estimation is provided in WGS84 coordinates, RD coordinates and a height above NAP (*Normaal Amsterdams Peil*). Both data sets span the same area of 123 km², of which 60 km² is over urban (built) terrain, where the highest density of persistent scatterers is to be expected [CBS, 2017, Hanssen, 2001].

Error model

The geolocation precision of TerraSAR-X was estimated by Dheenathayalan et al. [2016] and verified using a field experiment. The ratio of the error ellipsoid in range (direction of radar signal), azimuth (direction of flight) and cross-range (perpendicular to signal and flight direction) was estimated to be 1/2/22 for typical consistent scatterers. With an estimated standard deviation of 0.128 m, the axes of the error ellipsoids are 0.128 m (range, σ_r), 0.256 m (azimuth, σ_a), 2.816 m (cross-range, σ_{cr}). These values were assumed constant for all scatterers, as the per point estimates, or the parameters required to estimate them, were not present in the data available.

The direction of flight was defined to be 192° for the descending and 350° for the ascending orbit, with a elevation angle of 65.9° for both orbits. Both are considered error free and equal over the whole data set.

Sentinel 1

Given the free availability of Sentinel 1 data, that covers large parts of the Europe and the world, new studies on a national level are being planned [e.g. Novellino et al., 2017]. As indicated by Dheenathayalan et al. [in press] a high spatial precision (decimeters) can be achieved for trihedral reflectors, albeit at a lower spatial resolution. This makes it comparable to TerraSAR-X in that respect and suitable for the application of the same or a very similar matching technique. Sentinel acquisitions were processed by the TU Delft radar group for large parts of the Netherlands, as can be seen in figure 2.8. Unfortunately the processing was aimed at large scale deformation estimates, rather than geolocation estimates. As a result geolocation estimates are poor and follow the regular pattern of the original radar image, as can be seen in figure 2.7. This makes the data unsuitable for the planned application of geometric linking (error too large).

2.2. LiDAR

Optical distance measurements can roughly be divided in two main branches: triangulation (angle measurements) and time/coherence measurements (time-of-flight and interferometry). In this thesis only time-of-flight measurements are covered, as most mid- and long-range sensors are of this type. Lasers provide much higher angular and range resolutions (for single measurements) than possible with radio waves (radar), [Vosselman and Maas, 2010]. Just like with pulse based radar, given the known propagation velocity of light in air, (travel) time and distance are linked. This principle is illustrated in figure 2.9.

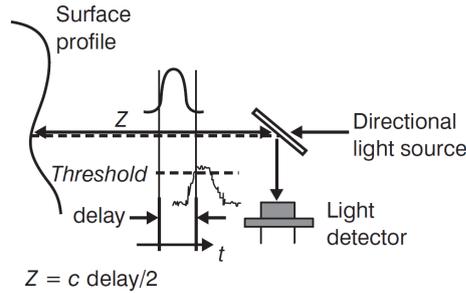


Figure 2.9: Principle of a time-of-flight laser scanner. The distance (Z) is calculated from the travelling time of the emitted pulse (*delay*), to the surface and back to the sensor, [Vosselman and Maas, 2010].

The range, ρ , is directly related to half of the the travelling time, τ ,

$$\rho = \frac{c \tau}{n 2}, \quad (2.3)$$

with the speed of light, $c = 299\,792\,458$ m/s, and the refractive index of air, $n \approx 1.00025$, dependent on temperature, pressure and humidity.

Such laser ranging systems are capable of reaching an accuracy in the order of 5 - 10 mm, not including the position and attitude accuracy of the sensor. Furthermore, frequently the intensity of the returned signal is measured, mapping the reflective properties of the object and providing a visual reference, [Vosselman and Maas, 2010].

Although the ground footprint of the laser is much smaller than that of radar based systems, the emitted pulse may still reflect from multiple surfaces, such as the branches of a tree canopy. Most airborne sensors are capable of capturing multiple peaks in the return signal, recording measurements for all surfaces visited by the laser beam. Some sensors are capable of capturing the full waveform, the intensity profile over time of the returned signal, [Vosselman and Maas, 2010].

To measure more than a single point the laser pulse is moved/pointed in a regular pattern using a moving mirror. As the attitude of the mirror is known the direction of the laser pulse can be determined. Combined with the distance, known from the time-of-flight, it is possible to determine the position of the reflecting surface in 3D with respect to the sensor. A small (pointing) error is introduced by the mirror.

To map larger areas the sensor is placed on a mobile platform, for example an airplane, helicopter, car or train. Moving the sensors while measuring. The position of this airborne sensor is determined using a Global Navigation Satellite System (GNSS, e.g. GPS) and the attitude of the sensor (the looking angle) using an Inertial Measurement Unit (IMU), [Vosselman and Maas, 2010]. The errors of the position and attitude propagate do not influence the range measurement itself but propagate into the position estimate of the reflecting surface.

AHN, Actueel Hoogtebestand Nederland, is such an airborne LiDAR survey. As main point cloud data source of this thesis a separate chapter is spend on the properties of this data set.

2.3. Related work

A selection of related work is presented here. Various works exist on the matching of radar data with building models or photographs. Furthermore a plethora of work exists on the analysis of point cloud data and feature extraction. All work on large areas faces high computational and storage requirements, various methods of data organization and querying are discussed.

Fusion of building geometries with InSAR measurements

A combination of optical images and SAR aimed at the texturing, classification or 3D reconstruction of SAR point clouds, has been made before, for example by Tupin [2010] and Schmitt et al. [2017].

Although successful, combinations suffer from poor InSAR geolocation precision, [Wang et al., 2017]. When building models are combined with PS-InSAR data essential building details may get lost in the generalization of the 3D model, [Schunert and Soergel, 2016]. A third approach is chosen by Schunert and Soergel [2016]: attaching PS-InSAR points to the surfaces of a 3D building model. Attachment to relevant surfaces is very successful, but the generalization of the surfaces makes assignment to individual features impossible.

Other attempts combine the radar data with photographs, battling alignment, temporal and radiometric differences, [Schmitt et al., 2017]. Merging optical and radar data can benefit both data products, for example it aids interpretation of radar data by optical texturing and the enhancement of contrast in optical images based on radar, [Tupin, 2010].

Wang et al. [2017] combined aerial photographs and high resolution radar data. First a point cloud is constructed from the aerial images using photogrammetry. Their combination allows for texturing the radar data and the creation of a 'SARptical' point cloud of urban areas. Geolocation can not be improved based on the point cloud, as accuracy and precision do not allow for individual improvement. Instead the radar and optical point clouds are coregistered using a variation of the Iterative Closest Point algorithm (ICP). Facades are poorly presented in this model, as they would require oblique images.

Mou et al. [2017] and Hughes et al. [2018] extended the work of Wang et al. by adding a deep-learning technique that attributes scatters to features in an image, thereby linking optical images to radar scattering behavior.

Opportunities for improvement

Airborne laser scanning outperforms photogrammetry in mapping areas of poor texture and mapping long, narrow features, [Baltsavias, 1999]. Unfortunately acquisition of LiDAR data has its own problems, such as offsets between acquisition strips, [van der Sande et al., 2010]. Fortunately quality descriptions are available [e.g. van der Sande et al., 2010, van der Zon, 2013, van Meijeren, 2017].

Although a combination with the LiDAR point may suffer from the same issues identified by Schmitt et al. [2017] for photogrammetric surveys (alignment, temporal and radiometric differences), the high quality of individual measurements may allow for a combination between the data sets. Furthermore the combination can be used to improve the geolocation of the PS-InSAR, under the assumption that the PS-InSAR reflector is described by the geometry in the point cloud.

Various methods exist to reconstruct buildings or surfaces from point clouds. A benchmark for complex shapes is presented by Berger et al. [2013]. Fortunately most buildings are more regularly shaped than those complex shapes, many buildings contain straight walls only, [Yari et al., 2014]. Others can be approximated by a series of rectangles, [Vinson and Cohen, 2002, Vinson et al., 2001]. Or by a standard model for a house, where necessary replaced by fitting a series of straight planes, [Maas and Vosselman, 1999]. In case of low point cloud densities ground plans could be added to enhance the building reconstruction, [Vosselman et al., 2001]. High quality point clouds may allow reconstruction based on a series of geometric primitives to approximate the original surface, [Baek et al., 2017]. But most generalizations lose small scale information that might be important in the understanding of the radar scattering characteristics, [Schunert and Soergel, 2016].

In 3D building models, in case these are advanced and detailed enough, potential simple dihedral and trihedral reflectors are immediately obvious at the intersections of planes. But as illustrated in figure 2.10, more complex situations might be more complex to detect.

Detecting dihedral and trihedral geometries

When no building model is available corners might be detected directly in the point cloud or on a Digital Surface Model (DSM) [e.g. Kovács and Tevesz, 2013]. Methods that work directly on the point cloud often require the *normal* for each point, a vector perpendicular to the surface the point is a member of. Weber et al. [2010], for example, detect sharp features based on analysis of the variability in the orientation of the normals of neighboring points. Briese [2004] used a similar technique to detect breaklines, the line of intersection between surfaces.

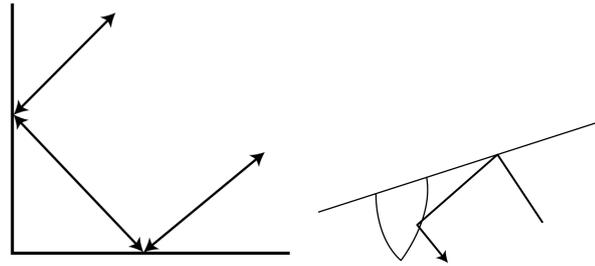


Figure 2.10: Left: a textbook example of a dihedral corner reflector. Right: a less obvious example of a dihedral reflector in a non-standard geometric arrangement. [Richards et al., 2010]

The normal is unavailable in most point clouds as the original surface the point belonged to is unknown. Various methods exist to estimate the normals of an existing point cloud, such as: fitting a analytic function through the point cloud; fitting a plane through or creating a triangular mesh of the points.

Strictly speaking those methods do not give the normal at the point but the normal of the surface formed with the surrounding points, and yield poor results at the intersection of surfaces. More complex methods, such as the deep learning method by Boulch and Marlet [2016] try to enhance the prominence of abrupt changes in the direction of the normal at the cost of computing power.

Alternative methods, that do not depend on normals often depend on finding the surfaces directly. The method by Auer and Hinz [2007] uses a triangulation for region growing rather than normal estimation. It creates generalized planes of (roof) surfaces based on the triangulated mesh between points.

Another approach is estimating the parameters of the surface directly, using the Hough transform, for example. The implementation by Bhattacharya et al. [2000] is mostly applied to depth images rather than point clouds, but the method can be applied on point clouds too. It detects groups of planes with similar properties first before distinguishing between individual planes and their intersection, reducing computational cost. A comparable approach was chosen by Rohr [1992], uses corner detection in images to detect the position of the (possible) corner. A model of a corner is then fitted based on this initial position and orientation. Using Markov Random Fields, Hackel et al. [2016] determined the likelihood of a point belonging to a corner.

Random Sample Consensus (RANSAC), introduced by Fischler and Bolles [1980], can be applied for the detection of planes too. RANSAC is capable of fitting a model to data that contains a significant percentage of noise. Other corners or planes, close to the area of interest, may not necessarily be noise and finding the correct settings might be difficult with the large variation in (point) density and difference in scale of the scattering features. The *pbM*-method, by Subbarao and Meer [2006], is free of predefined parameters and might solve this issue of RANSAC.

None of those methods are able to detect the more complex reflectors such as shown in figure 2.10. Detection of such reflectors would require a ray-tracing model, that simulates the path of the radar signal from and to the radar satellite, [e.g. Yang et al., 2017].

Data organization and storage

Files have shown to be a very reliable form of storage. LiDAR data sets can grow to enormous amounts of data. With an average density of 25 points per square meter for airborne LiDAR and even higher densities for mobile LiDAR efficient storage and access is required, [Boehm, 2014]. Compression algorithms, such as *LASzip* [Isenburg, 2013], exploit the similarity between points to reduce the file size up to 90%. Reduced disk space requirements come at the cost of increased CPU usage.

Querying large data sets for individual points or segments is non trivial. Boehm [2014] suggests keeping the spatial extends of each individual file in a database. Querying the database for the tiles involved first, then querying each candidate tile for the point involved. Van Oosterom et al. [2015] took the integration of points clouds and databases one step further, by testing the performance of various point cloud databases on inserting and querying of individual points and on more complex queries. A tile (file)

oriented approach was fastest for simple queries, while the database environment performed better on complex queries.

Algorithms for Nearest Neighbor search are often based on the Euclidean or the Manhattan distance metric. When the statistical properties of a point cloud are accounted for the distance metric is more complex and above all different for every statistical configuration. Use of a *whitening transform* transforms the data to a subspace compatible with the Euclidean distance metric, allowing use of conventional Nearest Neighbor algorithms, [Stansbury, 2013]. This process is also known as *decorrelation* in geodetic applications, [e.g. Teunissen and Kleusberg, 2012].

This transformation is based on the statistical properties, therefore the change of properties requires a new transformation of the data and rebuilding of any search structure. This problem is solved by Sakurai et al. [2001] by preparing many transformations and accompanying search structures. At query point the most similar transformation is used to find the approximate nearest neighbors after which the distances are checked (and sorted) using the exact transformation. The increased speed at search time comes at great computational cost in preparing the data set.

A completely different solution is the use of *Space Filling Curves*. Where points are indexed or sorted along a single linear index. This index is constructed in such a way spatial proximity is represented by numerical proximity in the index. When this linear index is fine enough, the index could even replace the original coordinates, as the coordinates can be reconstructed based on the index, [Psomadaki, 2016]. Search queries with complex geometries can be solved in iterations by such an index: when no points are found the search area is enlarged up until a predefined maximum.

2.4. Remaining challenges

Although various studies showed successful alignment of the deformation signal with buildings and building geometries, the signal can only be analyzed at the level of detail of facades or planes. Individual features can not be distinguished, nor is the positioning accuracy and precision of the point cloud used to improve the geolocation estimation.

The challenge is to develop a method that is capable of matching the radar measurements to the LiDAR geometries while respecting the statistical properties of both: a fusion method at a high level of detail, capable of data fusion at a national level.

3

Actueel Hoogtebestand Nederland

Actueel Hoogtebestand Nederland (AHN) is a nationwide LiDAR elevation model, covering the whole territory of the European Netherlands. AHN was first recorded in 1996 and is licensed as open data since March 2014. This chapter will take an in depth look at the properties of AHN as an information resource. It will discuss the properties of the data, including the error model and show the steps taken to adapt AHN for the task at hand.

AHN was acquired from an airborne platform, from which laser pulses were fired at the ground below. Given the known propagation velocity of light in air, the time interval between transmission and receiving the reflected signal (*echo*) is proportional to the distance from the aircraft to the ground. Multiple returns are possible, for example in vegetated areas, where parts of the pulse reflect on different surfaces in the scene. The position and orientation of the aircraft are recorded simultaneously using GNSS and inertial motion sensors to record the position from where the measurement was acquired and in which direction [Vosselman and Maas, 2010].

The first release of AHN (1996 - 2003) was at the forefront of technology. As technology advanced, density requirements were increased from 1 point per 16m² to 1 point per square meter. Classification of points was only applied outside the built environment. Due to the low point density the quality of the ground estimation was limited in vegetated areas, as not enough laser pulses penetrated through the canopy, [van der Zon, 2013].

New iterations are acquired approximately once every ten years, as can be seen in table 3.2. Acquisition of both AHN1 and AHN2 is finished: AHN2 supersedes AHN1. The most recent version, AHN3 is yet only partially available (figure 3.1), with new regions added yearly till 2019. Contracts for AHN3 were published as six tenders between 2013 and 2017 for acquisition over the period 2014 - 2019, [e.g. Rijkswaterstaat, 2013]. The raw point cloud data (of each iteration) is published via *Publieke Dienstverlening Op de Kaart* (PDOK) [PDOK, 2018].

As of AHN2 the technical specifications are no longer defined for the acquisition but on the final products. The acquisition strategy is left to the surveying company. For example, there is no requirement on the point density, given that these requirements are met, [N.N., 2013, van der Zon, 2013]. The final product should comply with the following requirements:

- Elevation accuracy/precision should be less than 5 cm systematic and 5 cm stochastic (standard deviation), although no definition of 'systematic errors' is provided.
- The accuracy, precision, density and distribution of points is such that a 2 by 2 meter object has a position error of maximum 50 cm.
- Classification of the points. For AHN2 this classes are as ground and non-ground points, for AHN3 the following classes are defined: ground, buildings, water, civil structure and unclassified.
- The final product should include a Digital Elevation Model (DEM) of ground points, on an equidistant grid of 0.5 by 0.5 meter.

AHN is defined in *Rijksdriehoekscoördinaten* with height relative to NAP, 'RDNAP', the Dutch national coordinate system (EPSG:7415). This Cartesian coordinate system is used as the basis for this project.

For AHN4, which is currently in the planning phase, the conditions are reconsidered and where necessary adapted to the current possibilities and requirements, [Leusink, 2018].

3.1. Available data

A nationwide coverage of AHN2 was finished in 2012, and full coverage for AHN3 is expected mid 2019. A summary of the currently available data and volume is given in table 3.1. The spatial availability of AHN3 can be found in figure 3.1. The file size increase of AHN3 compared to AHN2 is due to the addition of extra attributes, such as classification, intensity and acquisition time. In total 1447 tiles are defined, as some only cover water, a full data set contains less tiles. Tiles can be subdivided in 25 equal size sub-tiles to lower computational requirements when working with tiles¹. These tiles are made from a *combined* product of AHN2 and AHN3, and will be elaborated on in section 3.3.

	Area (km ²)	Points	Size	Tiles
AHN2	35 997	640 billion	988 GiB	1370
AHN3	16 249	252 billion	1.1 TiB	765
Combined			1.7 TiB	30137 (sub-tiles)

Table 3.1: Coverage, point count and data set size of AHN2, AHN3 and their combination. Although not yet fully available, storage requirements for AHN3 will double with respect to AHN2 even though the point density is approximately equal.

	Recording	Avg. Density ($\frac{\text{pts}}{\text{m}^2}$)
AHN1	1996 - 2003	0.06 - 1
AHN2	2008 - 2012	16.8
AHN3	2014 - 2019	18.2

Table 3.2: Acquisition years and average point density of the three available iterations of AHN. Actual point densities for AHN2 and AHN3 were estimated over the study area in Delft (9 tiles).

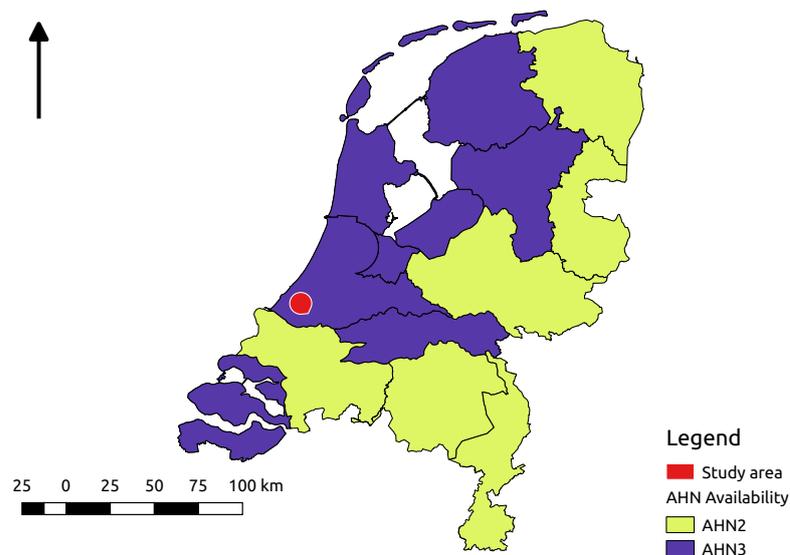


Figure 3.1: Map of the Netherlands, showing the availability of AHN3 (purple) at the time of writing, AHN2 is available in all regions. Shown in red is the extent of the TerraSAR-X data available to this study.

¹An example of the sub-tile structure is shown in figure A.1, the main tiling schema at national level is shown in figure 3.12.

To ease navigation, I colored the point cloud based on the publicly available aerial photograph of 2016 [PDOK, 2018]. This photo may differ from the point cloud, as it was not recorded simultaneously. For AHN4 simultaneous acquisition of LiDAR and imagery data is planned, [Leusink, 2018].

Vertical density

As airborne LiDAR product acquisition is focused on the nadir direction (looking down). Horizontal surfaces are covered with high density, while vertical surfaces suffer from low sampling densities. An example is shown in figure 3.2, which shows the vertical point density on the facade of the EEMCS faculty building at Delft University of Technology. The roof is densely sampled while the facades are sparsely sampled. In point cloud viewers the surface density difference is less obvious, as can be seen in figure 3.3. The other (rear) facade of the building is not present in the data at all. Which part of a building is sampled, and which is in the *shadow*, is dependent on the flight line and the equipment used. For AHN4 experiments are conducted with equipment better capable of sampling walls and facades, [Leusink, 2018].

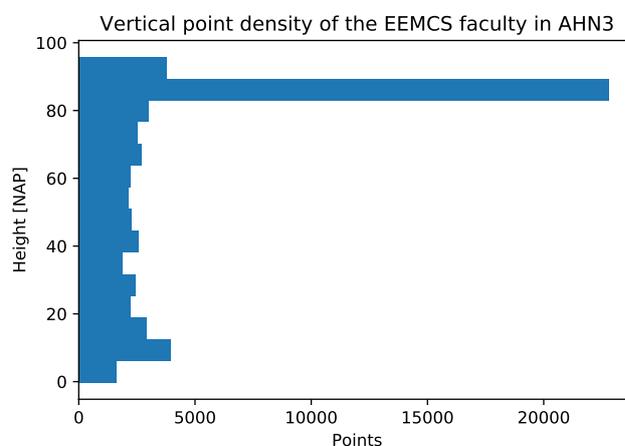


Figure 3.2: Vertical point density profile along the height of the EEMCS building. Only points classified as 'building' within the building outline are shown. Eight height meters of facade cover approximately the same surface area as the roof. The (horizontal) roof is frequently sampled, while the point density on the facades is only a fraction of that.

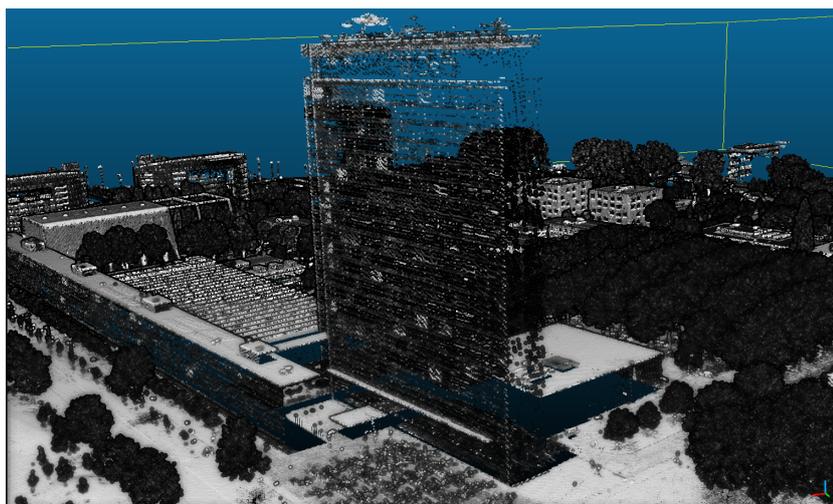


Figure 3.3: EEMCS faculty building in AHN3, visualized in CloudCompare, looking to the south-west. The low point density on the facades results in visible gaps of the building outline. The non-uniform point distribution on the facade appears as horizontal lines, corresponding to small ridges between the exterior plating, and patches of higher point density related to sunblinds.

Known errors

Various tiles, as published on PDOK [2018], suffer from corrupt files or missing segments². The governing body of AHN (Waterschapshuis) and Rijkswaterstaat were contacted about these problems and new tiles were provided to PDOK in July 2018. Only AHN2 ground tile 16DZ1 is unrecoverable. With AHN3, some tiles³ that overlap with multiple concessions do not contain points from both concessions (figure 3.1). This results in gaps in the data if not replaced by older data from AHN2. As of July 2018 new tiles were provided for part of the data⁴.

3.2. Error model

For AHN (iteration 2 and 3) the accuracy is defined as maximum 5 cm systematic (1σ) and precision as a 5 cm stochastic error (1σ) in the vertical direction. Requirements for horizontal accuracy are 50 cm (1σ , both x and y) for objects larger than 2×2 m. In reality this is often outperformed [van Meijeren, 2017], but differences between acquisitions might be larger [Soudarissanane et al., 2010, van der Sande et al., 2010]. Quality reports, including accuracy assessment, were commissioned for the individual tenders/concessions. Unfortunately these reports are not publicly available. The tender states that re-acquisition of the data is necessary if the requirements are not met, [Rijkswaterstaat, 2013], therefore one may assume that the data meets at least the requirements set in the tender.

To assess the applicability of the error model the surfaces of the runways 06/24, 18L/36R and 04/22 of Schiphol Airport were analyzed in tile 25DZ1_09.

Assessment of the stochastic error

The vertical stochastic error has been assessed by measuring the surface roughness of a known (locally) flat surface: the runways. For this purpose the roughness is defined as the distance from a plane fitted through neighboring points. For each point a flat surface is fitted through all neighboring points within 1 meter distance, the distance of the point to this surface is the roughness of this point, [Girardeau-Montaut et al., 2018].

In this tile the requirements set in the tender are outperformed, as can be seen in figure 3.4. The roughness is half-normal distributed with $\sigma = 0.031$ m and zero mean. The spatial distribution of the roughness is related to the flight lines, with higher roughness where data of multiple flight lines meet. When only a single flight line is analyzed, roughness is no longer half-normal distributed. This is either due to imperfections in the runway used as reference or due to non-normal distributed perturbations dominating the error of the LiDAR measurements. Furthermore correlation between the surface reflectance, roughness and precision can be seen (compare figure 3.7 with figures 3.5 and 3.6).

Assessment of systematic errors

To assess the systematic error the height difference between the various flight lines the distance between flight line 54048 and 54049 (see figure 3.7) is calculated. To mitigate the noise of individual points local planes are fitted with all points within 1 meter. The resulting average distance (μ) is 0.05 meter over the whole runway ($\sigma = 0.02$ meter), where spatial variants correspond to differences in the pavement structure. As flights were performed in a short time span, temporal changes (e.g. subsidence) can be neglected. Therefore this is an indicator of the systematic error between flights.

As the roughness is measured as an absolute number a zero-mean offset might be hidden in the data. An Iterative Closest Point algorithm was run to determine the distance between the point clouds. The vertical distance between the two point clouds was determined to be 7 cm. Above the expected zero-mean with a standard deviation of 5 cm.

²This includes the following tiles of AHN2, that contain only ground points: 01CZ2, 07CN2, 11EZ2, 16HN2, 16DZ1 and 16DZ2. The tile containing only non-ground points for 57EN2 is corrupt too.

³Especially: 16DN1, 16DN2, 16EN2, 16EZ1, 16EZ2, 16FN1, 16FN2, 25EZ1, 25EZ2, 25GN2, 25HN2, 26CZ1, 26CZ2 and 26DZ1.

⁴Only 25EZ1, 25EZ2, 25GN2, 25HN2, 26CZ1, 26CZ2 and 26DZ1.

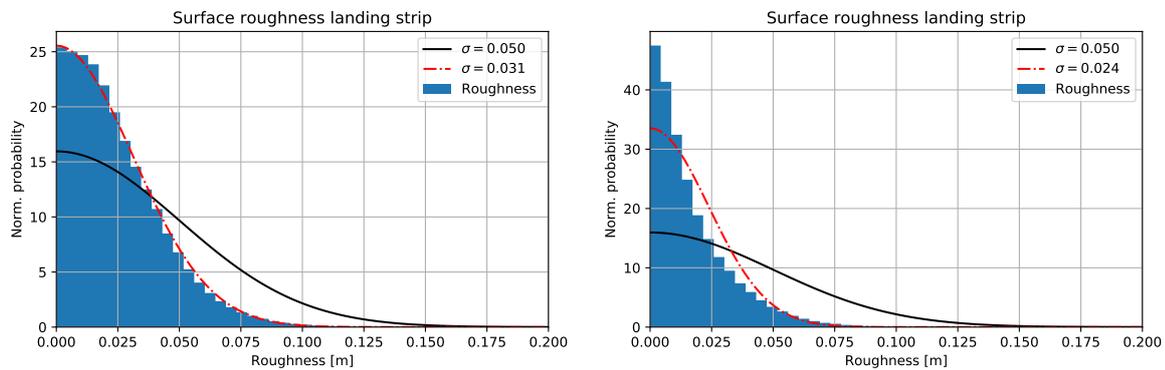


Figure 3.4: Normalized histogram of the surface roughness at the runways. The density function as required in the AHN tender is shown in black and the fitted, empirical, density function in orange (dashed). Left: a combination of all flight lines. Right: only roughness values from flight line 54048. The requirements on roughness are easily met on the runway, while for a single flight line the roughness is no longer a half-normal distribution, suggesting that the noise is no longer normally distributed.

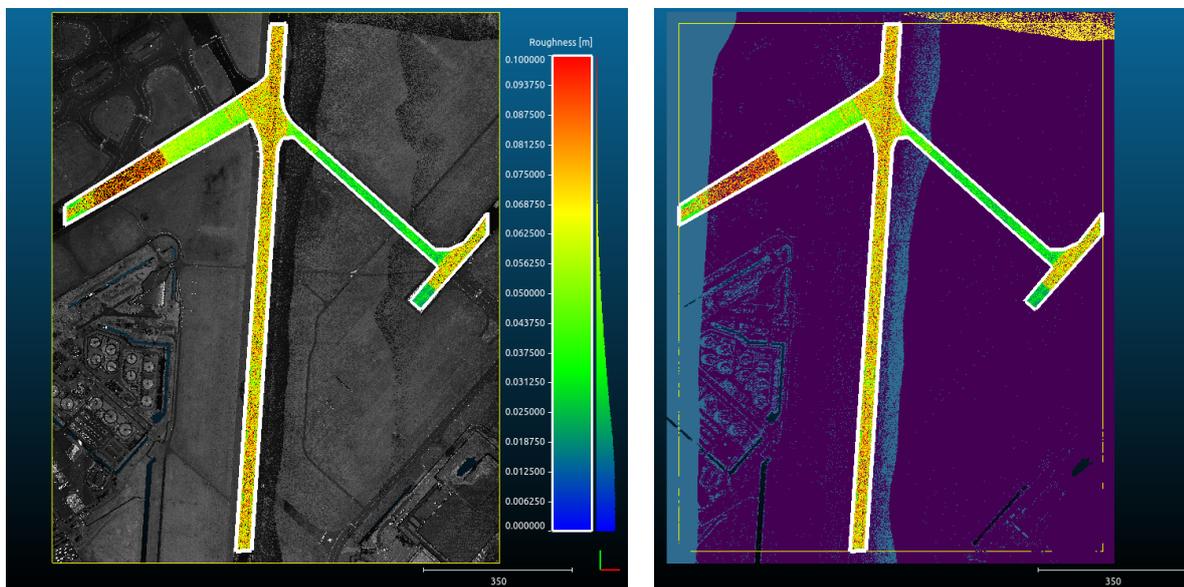


Figure 3.5: Top view of tile 25DZ1_09, which contains runway 18L/36R of Schiphol Airport. The white outline indicates the tarmac runway area. For this area the surface roughness is shown (in meters), including a histogram, which is approximately half-normal distributed with $\sigma = 0.03$ m. The background image on the left: intensity from LiDAR data. On the right the flight line the data originated from (dark blue, light blue or yellow), a breakdown of flight lines is shown in figure 3.7. Variations in roughness show correspondence to flight line boundaries.

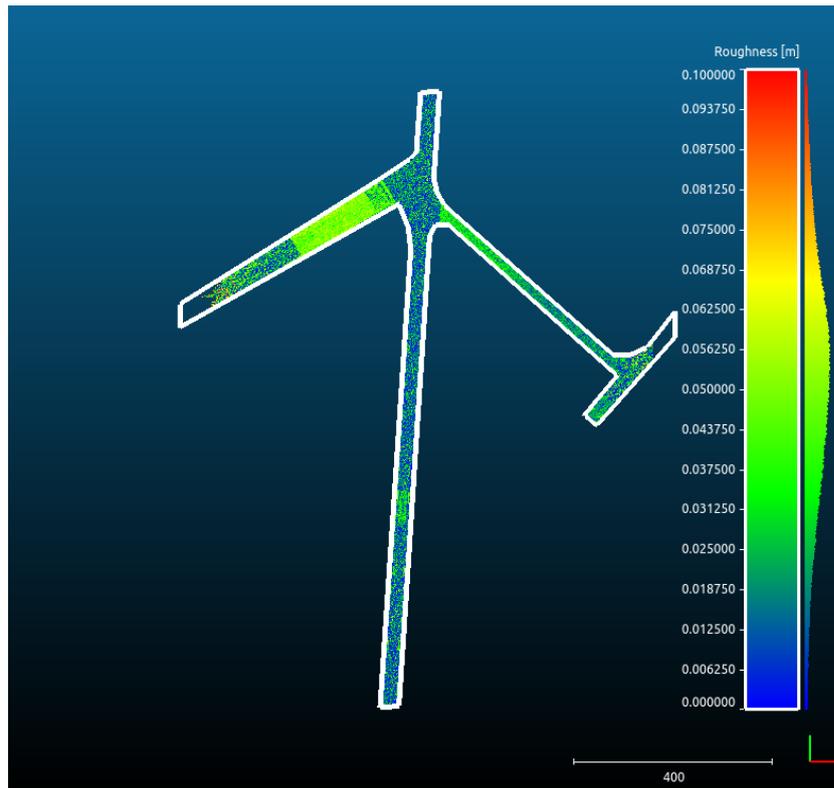


Figure 3.6: Distance between flight line 54048 and 54049, approximated by local planes to reduce the effect of noise. The non-zero mean indicates a systematic difference between the acquisitions. The offset is different for different patches of tarmac, possibly related to differences in surface reflectance.

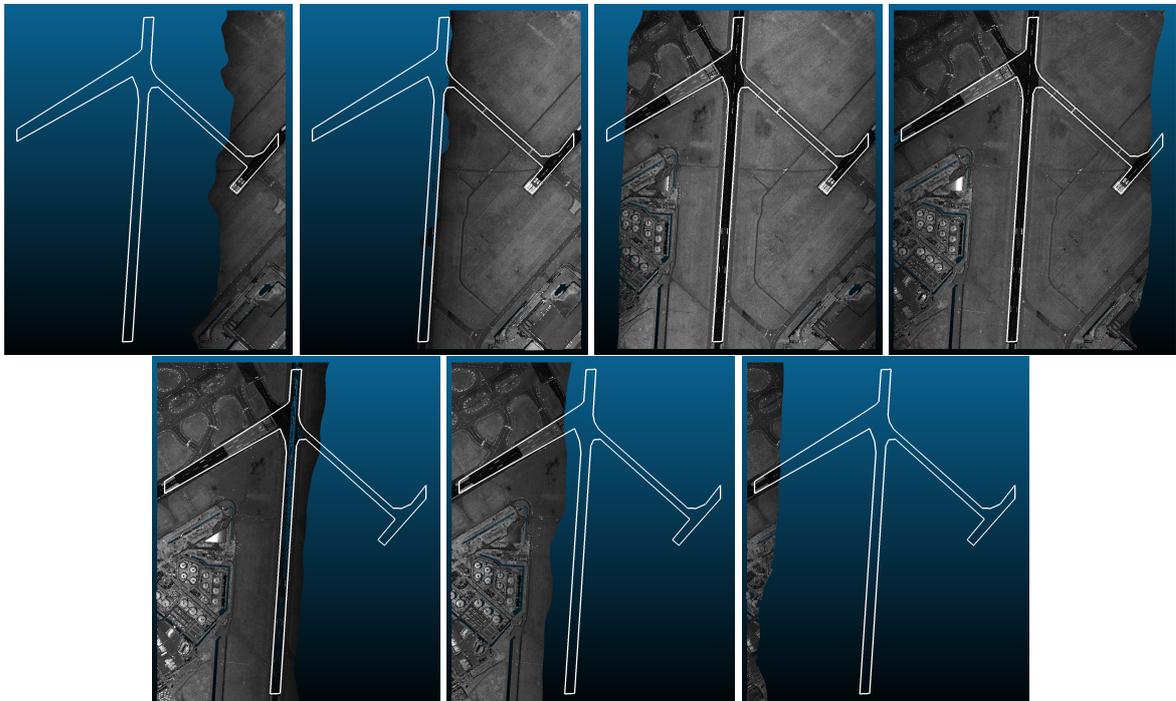


Figure 3.7: Intensity variations over the runway for seven different flight lines (first row: 54046, 54047, 54048, 54049; second row: 54050, 54251, 54052), related to the incidence angle. No, or insufficient, intensity correction is applied.

As the definition of the vertical error is unclear, given the requirements are not met, the pessimistic scenario of the addition of both the stochastic and the systematic error is assumed for the error model. This leads to the vertical error being defined as $\sqrt{0.05^2 + 0.05^2} \approx 0.7$ meter standard deviation for calculations in this thesis.

Matching earlier observations by van Meijeren [2017], the large standard deviation and the infrequent acquisition make AHN itself unsuitable for deformation monitoring at the millimeter level that is obtained by InSAR monitoring. Furthermore not necessarily the same point is measured in each epoch, resulting in apparent differences (false positives) or missed differences (false negatives).

Other errors

Some errors do not follow the defined error model. For example at the EEMCS faculty the bicycle parking and porch in front of the building is projected into the building. As can be seen in figure 3.8 and 3.9 the projected interior of the building does not match the true interior. The resemblance with the exterior objects is striking: these points are likely due to reflection on the glass facade (i.e. multipath). Although no matching with the interior of a building is intended, false points could provide a (false) match, introducing errors.

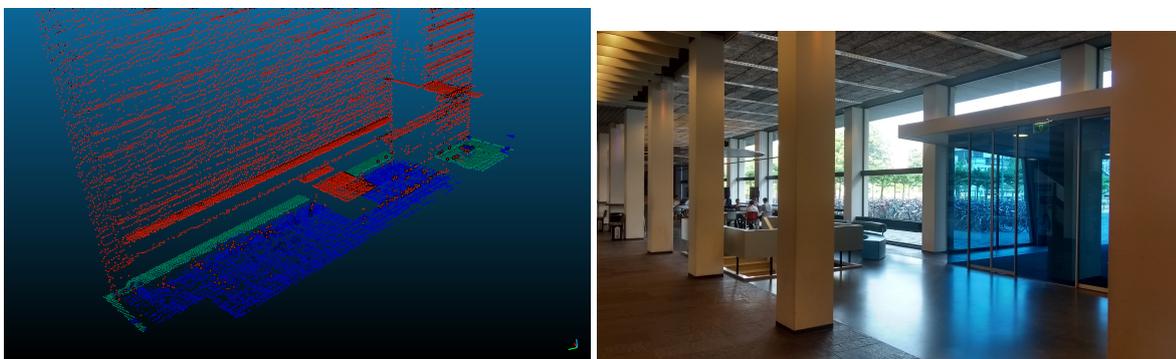


Figure 3.8: Interior entrance of the EEMCS building, as sampled in AHN3 and on a photo. Left: looking south east, right, interior: looking north east. The large roof, shown in red at ground level in the point cloud, does not exist in reality, and is a reflection of the exterior roof (compare figure 3.9).

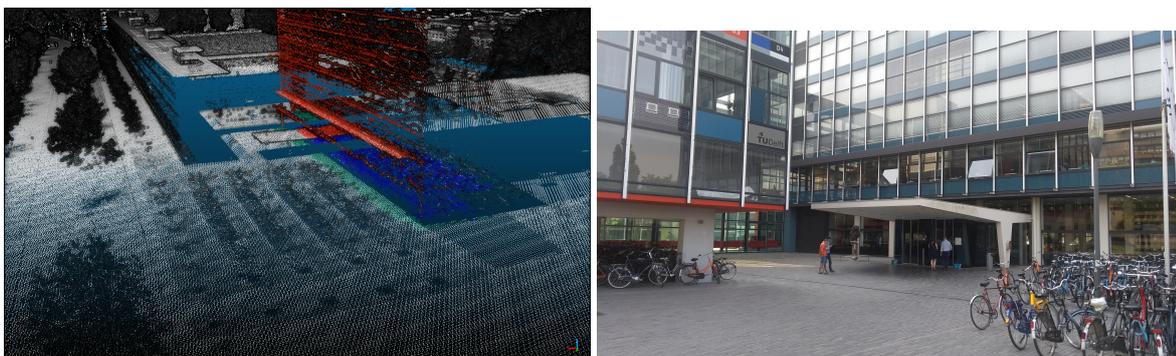


Figure 3.9: Entrance of the EEMCS building, as sampled in AHN3 and on a photo. Both images looking approximately south west, showing the roof erroneously duplicated inside the building (compare figure 3.8).

3.3. Data preparation

Two 'final' products are created from the raw AHN point clouds: a colored point cloud for web visualization and a tiled point cloud for data fusion. Data preparation is built around `LASTools` [Isenburg, 2018], `PDAL` [Bell et al., 2018] and `GDAL` [GDAL Development Team, 2018] and relies heavily on `xargs` for parallelization. An accompanying series of commands and hints on data preparation can be found in appendix B.

Processing can be subdivided in the following steps, visualized in figure 3.10:

1. Merging ground and non-ground points in AHN2.
2. Coloring the point cloud with the areal photograph.
3. Creation of a spatial index.
4. Concession crop, cropping tiles by concession boundaries.
5. Conversion to the Potree format (for the web viewer).
6. Tiling into smaller sub-tiles.
7. Estimation of normals.
8. Combining AHN2 and AHN3, using AHN3 where available.
9. Creation of a Digital Elevation Model, including statistics.

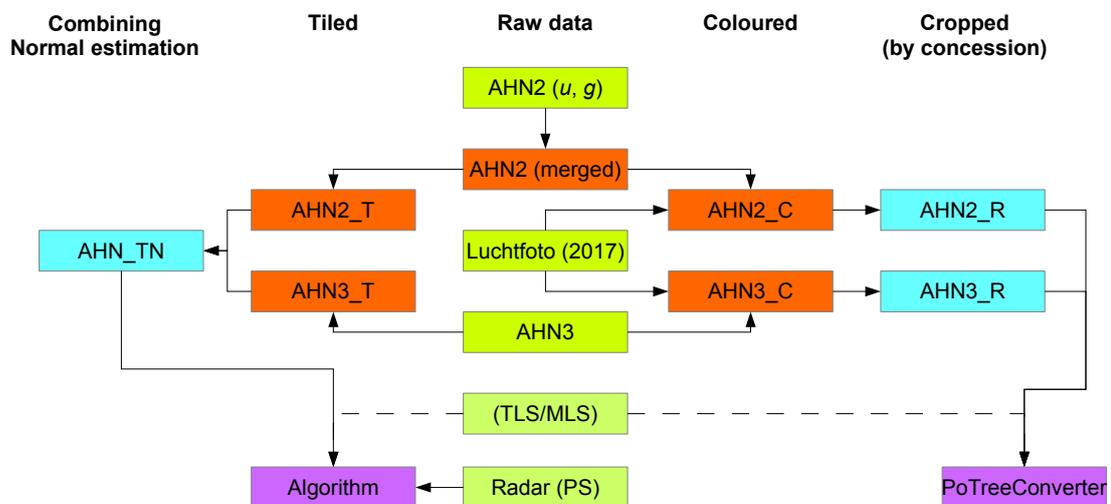


Figure 3.10: Flowchart of AHN data preparation. Input data (green), intermediate products (orange), final products (cyan) and software involved (purple). The raw point cloud data is prepared in various steps for visual presentation (merged, colored and cropped) and for the combination with the PS-InSAR data (merged and tiled), for which the algorithm will be outlined in chapter 4.

AHN data is delivered on PDOK in LAZ-tiles of 5×6.25 km, based on the standard national tiling scheme. An excerpt of this tiling schema and associated sub-tiles can be found in figure A.1. Availability boundaries do not follow this tiling scheme but follow water board (Dutch: *Waterschappen*) boundaries (figure 3.12).

Merging AHN2

Tiles for AHN2 are not delivered as classified LAZ-files but as two files: one file containing ground points and one file containing all other points. These tiles are merged where all non-ground points are marked as *unclassified*, rather than *never classified*. (Classification in accordance with ASPRS [2013].) Thanks to this merging future operations can read a single tile (instead of two) and use the ground classification in their process.

Coloring

In AHN2 data no intensity is available, as of AHN3 intensity attributes are available, but intensity values vary greatly by incidence angle (this effect can be seen next to the runway in figure 3.7). As the scan angle is provided with the scanning data, an intensity correction could be attempted. But to further ease navigation of the point cloud, with or without intensity values, an aerial photograph is overlaid on the points. This photo, provided by PDOK [2018], is downloaded using GDAL `en` overlaid on the point cloud

using `PDAL`. For the particularities on downloading the aerial photograph, please refer to Van Natijne et al. [2017]. `PDAL` is preferred over the previously used `lascolor` (provided with `LASTools`), as `PDAL` is open source and providing a free alternative to the paid version of `LASTools`. This step results in a copy of the raw (input) point cloud with added RGB values for each pixel.

Creation of a spatial index

A spatial index is used to speed up queries on the `LAZ`-files. Thanks to this index a full scan is no longer required to extract points within a bounding box. The `LASindex` tool by Isenburg [2012] does not reorder the points but lists where points within certain quadtree cells (a spatial indexing structure, illustrated in figure 3.11) can be found in the file.

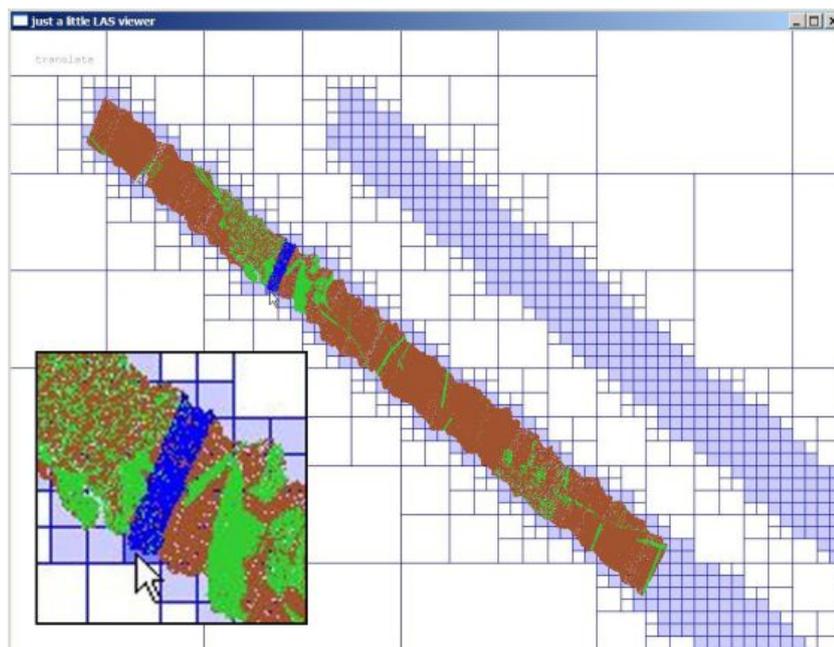


Figure 3.11: Example of a `LASindex` quadtree on top of a point cloud. The locations in the original source of the points are stored at each cell, [Isenburg, 2012].

Concession crop

Tiling of the visualization is built around the border of the water boards. This allows the replacement of all tiles of a water board once an update becomes available rather than updating tiles which overlap neighboring regions. Updating tiles would require re-rendering of the full Potree structure, while independent concessions can be replaced converting only this concession and pointing the webviewer to another resource.

First the intersection of tiles and concessions is calculated. Tiles at the exterior boundaries of concessions are split. This process is illustrated in figure 3.12.

Potree conversion

The concession, that includes the northern part of the province of Zuid-Holland, parts of Utrecht and the region under control by air-traffic control from Schiphol Airport, was prepared for use in Potree (figure 3.13). A total of 57 389 000 153 points was converted in 69 hours. Van Natijne et al. [2017] showed that the processing time required by `PotreeConverter` scales linearly with the amount of points processed ($\mathcal{O}(n)$). This process was done on a consumer grade Intel Core i5-2400 with just 8 GB of RAM and two Seagate 4TB consumer (USB 3) expansion drives for reading and writing.

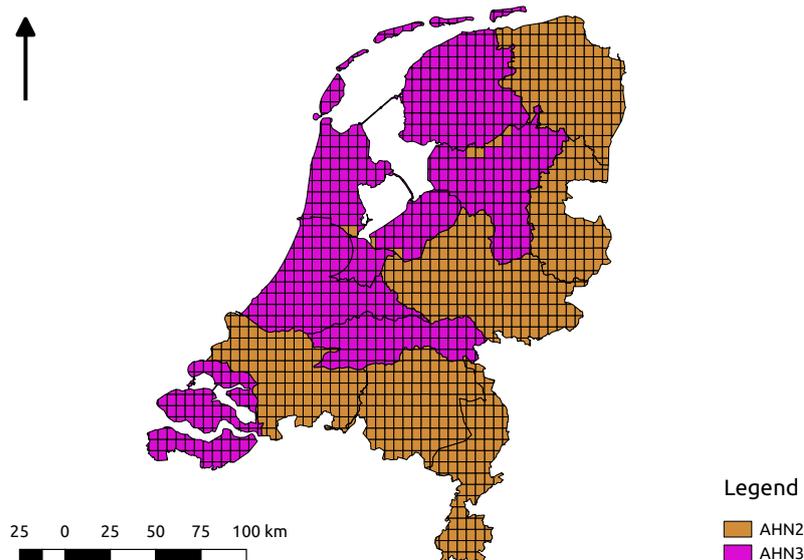


Figure 3.12: AHN is distributed in tiles. Tiles may exist both for AHN2 and AHN3 and might overlap. The availability of AHN2/AHN3 (figure 3.1) is shown here in the discretized tile pattern. Due to errors in the data some AHN3 tiles, overlapping AHN2, were unavailable at the time of writing.

Given there are thirteen concessions of comparable size, processing the whole of the Netherlands would take approximately 40 days. As each concession is completely independent, processing can be parallelized based on concessions rather than the internal structure of the generated octree, as proposed by Martinez-Rubi et al. [2015] (*Massive-PotreeConverter*). Although a maximum of (only) 13 parallel threads is possible when the concessions are split based on the boundaries of the (thirteen) water boards.

Disk requirements are large and have to be tailored to the storage of large quantities of small files. For just this concession the generated structure contained 2 332 999 files and had a size of 1022 GiB.

Tiling

For algorithmic purposes both data sets are tiled in tiles of approximately 1 km^2 ($1 \times 1.25 \text{ km}$, 4% of the original tiles). These tiles are small enough to be processed in memory, and large enough for regular file storage. For each LiDAR tile a buffer of 25 meters on all sides is included, this is to allow radar measurements on the border of two tiles to match. Given the radar error model, such a buffer is large enough to accommodate points on the border between tiles.

Tiling queries on the LAZ-files are elementary (axes aligned bounding boxes). This process is done using `las2las` of LAStools (tiling) [Isenburg, 2018], greatly accelerated by the previously generated spatial index. It created 55076 LAZ-files (tiles, 34777 AHN2, 20299 AHN3), with an average file size of 58 MiB and 21 million points (average point density of 15 points per square meter).

Due to the 25 meter overlap between tiles, the algorithm may be run for each tile independently, allowing *horizontal scaling* of the algorithm. That is, each tile can be processed independently, on a separate CPU or even separate node. This enables us to combine the PS-InSAR points with *massive* numbers of LiDAR measurements in a distributed manner, reducing the execution time required.

Normals

Normals are estimated for each point using `PDAL` [Bell et al., 2018], based on the eigenvectors of the point set consisting of the eight point neighborhood of each point together with the point itself.



Figure 3.13: Northern part of the province of Zuid-Holland, parts of Utrecht and the area under control of Schiphol Airport, together form a single acquisition concession. Shown here using Potree, as a colored point cloud (AHN3). The 57 billion points can be viewed in a regular web browser, on consumer hardware.

The ASPRS definition of the `LAS`-file does not allocate space for the storage of normals, [ASPRS, 2013]. As conversion to another (less compressed) format is undesirable, the normals are converted to eight bit integers (`int8`) to maximize compression benefits and are added as 'Extra Bytes' to the points in a new `LAZ`-file.

Combining AHN2 and AHN3

A combined data set of AHN3 and AHN2 is created, AHN3 is used whenever available (figure 3.1), otherwise AHN2 is used. This combination is made for computational purposes and therefore the sub-tiles, including their overlap are used. Tiles that are fully within a concession or do not cross any other concession (coastal tiles and tiles at the border with Germany and Belgium, for example) can be used directly and a `symlink` (link to the original file) is used instead of making a full copy.

Regional statistics

To assess the differences in AHN on a national scale a Digital Elevation Model was created at 5 meter resolution. Such a Digital Elevation Model (DEM) or Digital Terrain Model (DTM) is available from PDOK. This product, created from the tiled products, is primarily used to get an overview of the regional properties of AHN.

Furthermore the point density of each cell is calculated. These density figures, for example, show different acquisition strategies and density differences over the various concessions. See figures 3.14 (AHN2), A.2 (AHN3), A.3 (AHN 2/3). From these images it is possible to detect the lack of data in certain tiles in the north of the Netherlands, due to data errors.

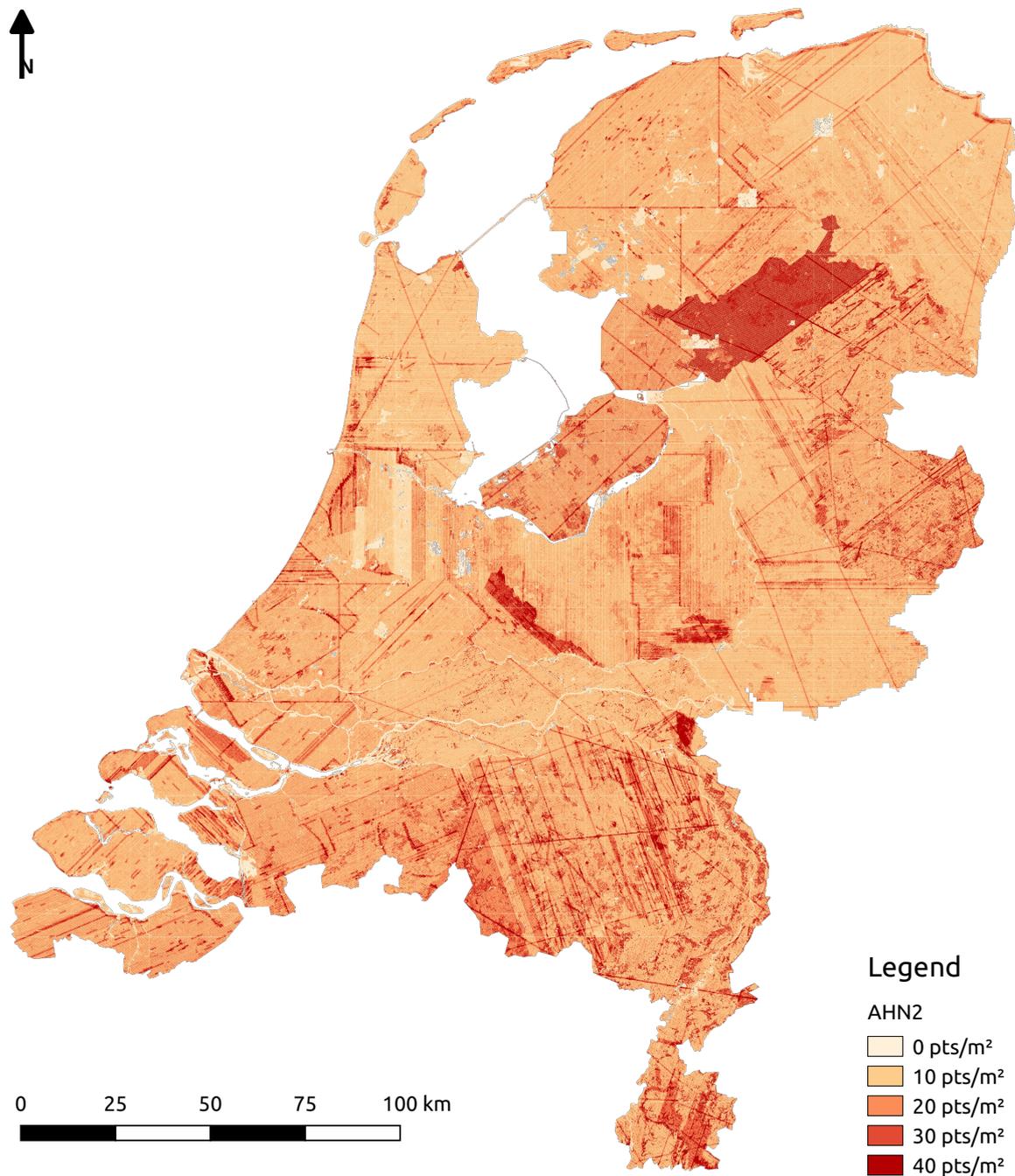


Figure 3.14: Point density of AHN2, sampled over a 35 meter radius (for 25×25 meter pixels). Large differences in point density are visible. Very low (less than 10 points per square meter) densities indicate missing or corrupt data, this includes natural effects such as large water bodies.

3.4. Data volume

Although many intermediate products can be removed after completing the data preparation, the total storage requirements are in the order of a few terabytes and can be covered with regular consumer storage solutions. An overview of the storage requirements can be found in table 3.3. The concession tiles are only stored for the most recent iteration of AHN.

	Raw	Colored	Concessions	Tiled	DEM	
AHN2	988 G	1.7 T	408 G	1.1 T	39 G	35 G (combined)
AHN3	1.1 T	1.3 T	340 G	1.2 T	22 G	

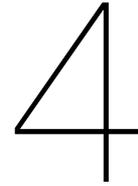
Table 3.3: Storage requirements for raw AHN, final and intermediate products. Due to overlap the tiled products are slightly larger than the original point clouds. Colored versions are larger as each point now contains RGB values too.

Conclusion

The whole of the Netherlands was covered by multiple airborne LiDAR surveys over the last twenty years. Point clouds from the last campaign (AHN2), and the currently running campaign (AHN3), can be used in this study. Average density is approximately 17 points per square meter, with limited coverage of vertical surfaces. Both campaigns have comparable statistical properties and missing data can be mitigated by using data from previous campaigns.

The data over the test site, a runway at Schiphol Airport, meets the quality description for the stochastic error. Furthermore it showed that the offset between flight lines was in the order of 5 centimeters, indicating the flight line is a neglected parameter in the error model. No assessment of planimetric error was made, due to the lack of a uncorrelated reference of high enough precision. Although the AHN LiDAR surveys themselves do not give deformation estimates, the positioning precision per point is such that it might aid in the improvement of the PS-InSAR geolocation estimates.

Data preparation is an elaborate process, merging the data in a uniform format and creating tiled versions of the point cloud for both visualization and matching purposes. For visualization color from an aerial photograph is added to the point cloud and tiles are cropped to the extends of the concessions to parallelize processing and ease replacement with new versions of the data. For the matching algorithm the point cloud is split in smaller sub-tiles, which include an overlap with neighbouring tiles. Thanks to this overlap, matching can be applied on all tiles independently. Each nationwide AHN point cloud, including intermediate products, requires approximately 1 terabyte of storage.



Methodology

The methodology is split into two parts: the visualization of the data, to aid the visual interpretation of the deformation signal and the fusion of both data sets, attaching the deformation signal to the geometries contained in the LiDAR data. After data preparation as discussed in chapter 3.3, the following five methods to achieve these goals will be discussed here:

1. Common visualization of InSAR and LiDAR data
Interpretation is left to the operator, just as with traditional (online) 2D maps (such as figure 5.6). The geolocation estimation is shown by an error ellipsoid in 3D. This enables improved interpretation of the radar signal and scene geometry over traditional 2D maps.
2. Nearest Neighbor linking
The Nearest Neighbor in the LiDAR point cloud (single point), with respect to the radar geolocation estimate, is assumed to govern the scattering behavior. This method is computationally efficient, but may overestimate the distance on low density surfaces, as illustrated in figure 4.1. If no such point is found within a (predefined) maximum range, matching is stopped, and no new coordinates are determined.
3. Linking to a single surface
After Nearest Neighbor search, either up to a predefined number of points or all points up to a maximum radius, a single plane is fitted to the points found. This plane approximation of e.g. a facade makes the matching algorithm more robust in areas of low point density and exploits the redundancy in the points for noise reduction.
4. Linking to multiple surfaces
For complex geometries the previous approach can be extended. Multiple locally linear ('flat') surfaces may exist in the neighborhood of the scatterer. This method was abandoned in favor of:
5. Linking to dihedral or trihedral geometries
Dihedral and trihedral surface configurations are known to act as good radar reflectors. These geometric configurations might be extracted from the scene.

The difference between the methods is sketched in figure 4.1. A local linear surface approximation, adds robustness in case of point density differences. On rough surfaces, or non flat surfaces this approximation may not hold. Rough surfaces are approximated by an 'average' surface and it might be possible to approximate non-flat surfaces as linearly close to the point of intersection.

Reconstruction of the full (building) geometry is not part of this research. Although matching to a 3D model would require less effort during matching, constructing the 3D model is no trivial task. Furthermore, similar attempts at matching between PS-InSAR and LiDAR point clouds showed that the level of abstraction of a building model influences the quality of matches. By working on the raw point cloud, reconstructing the surface area only locally, a high level of detail is achieved at low initial computational cost: only areas with permanent scatterers are reconstructed. Furthermore errors are reduced

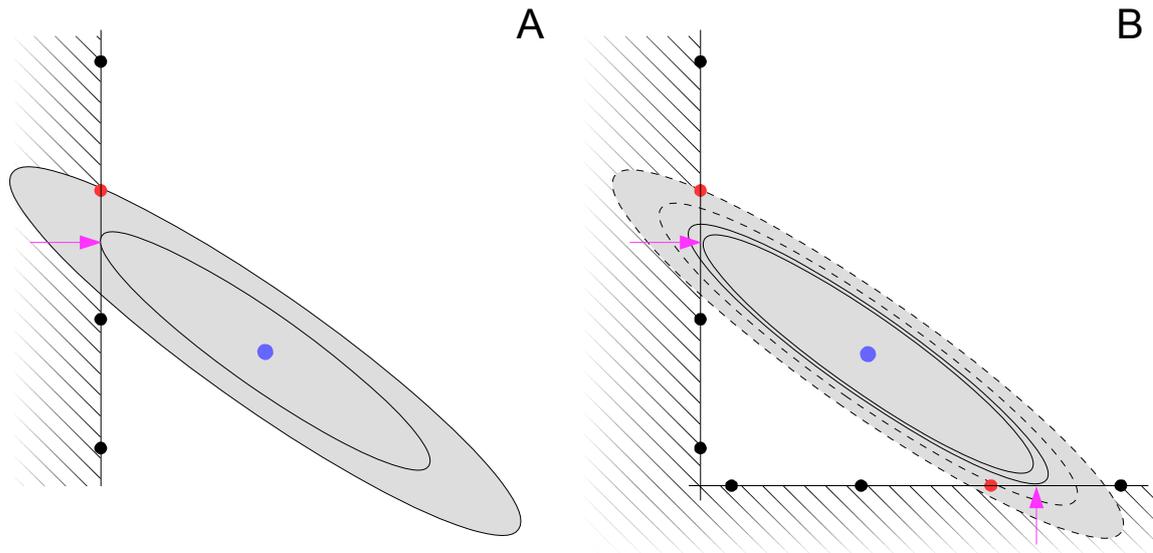


Figure 4.1: Sketch of the effect of surface reconstruction on matching between a LiDAR point cloud (black/red dots) and a PS-InSAR geolocation estimation (blue dot, including the error ellipsoid in gray). The match with the Nearest Neighbor in the point cloud, shown in red, is further away than the actual surface (thin black line, point of intersection indicated by the magenta arrow). Shown in a *simple* situation A; and a *complex* situation B. A: the distance, in terms of the size of the error ellipsoid (σ) is indicated as thin lines for both the closest point and the facade. The surface is closer, and therefore more *likely* than the closest point. B: the distance to the closest points is indicated by a dashed line, the intersection with the surfaces with thin lines. Again the surfaces are closer and more likely than the individual points. And although the horizontal surface is less likely than the vertical surface, it could be considered an alternative match.

to a single point only, where errors in a building model influence all scatterers in the vicinity.

After matching, the resulting matching distances may be used to analyze the geolocation quality, for example detection of biases and trends in the geolocation.

4.1. Common visualization of InSAR and LiDAR data

Web based visualization is built around Potree [Schütz, 2016, 2018] and Three.js [Cabello et al., 2018]. Potree is a WebGL based renderer for large point clouds in the web browser, built on top of the Three.js 3D library. WebGL is a 3D visualization technique for web browsers and online use. Previously the full (nationwide) AHN2 data set has been successfully converted to be used in the Potree viewer [Martinez-Rubi et al., 2015]. Other visual aids (such as the error ellipsoids and plane estimates) can be implemented using Three.js.

Potree ensures a smooth viewing experience by loading the point cloud from a preprocessed octree structure. Only the points in view at the client are downloaded and never more than a user-defined maximum. Due to the $2\frac{1}{2}$ -D nature of the radar data set this data is distributed using a quadtree tiling scheme, loading only the tiles in view and removing those no longer in view from memory. This process is elaborated on by Van Natijne et al. [2017].

The viewer might be extended with synthesized point clouds, simulating the effects of the deformation. These visual indicators aid the interpretation of the deformation signal, as the location of a subsidence bowl is not only colored but visualized in 3D. Furthermore the visualization can be true to the line of sight of the radar, without assumptions on the deformation behavior.

4.2. First Nearest Neighbor linking

Nearest Neighbor search should take the covariance matrix of each radar measurement into account. Without, the nearest points are possibly not the most likely points. Use of the *whitening transform* will al-

low any (conventional) Nearest Neighbor algorithm to be used on this problem [Stansbury, 2013].

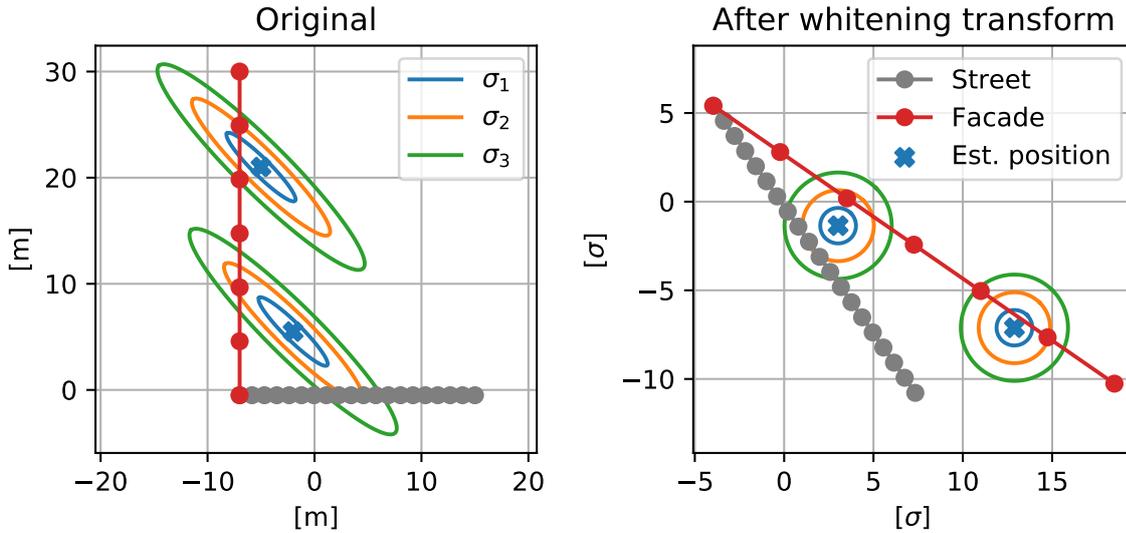


Figure 4.2: Effect of the whitening transform on the error ellipses of the PS-InSAR geolocation estimate and the point cloud. The original situation is shown on the left, on the right the whitening transform is applied. The geometry is deformed such that the error ellipses of the PS-InSAR geolocation estimates are reduced to circles. The unit of the axes is transformed from meters to standard deviations, σ , of the PS-InSAR geolocation estimate. As distance is measured in σ , the closest point or surface is the most likely candidate too.

Using the whitening transform all points (LiDAR and radar) are projected on the eigenvectors of the covariance matrix of the radar point (\mathbf{Q}_{ENU} , see section 2.1 and section 2.1) and scaled by the eigenvalues. This creates a new coordinate system where the Euclidean metric represents distances in σ rather than meters. All radar errors are now standard normal distributed, as can be seen in figure 4.2.

This transformation works for a single, constant, error model only. As a consequence the transformation has to be calculated and applied for each unique viewing geometry (i.e. orbit). This includes construction of a new search structure for each viewing geometry and/or error model. The Multiple Spatial Transformation Technique by [Sakurai et al., 2001], based on preprocessed search structures and approximate transformations may be used to speed up this process if required, but was not applied in this work. The alternative, a weighted distance metric, does not allow for indexing of the point cloud using traditional methods. The lack of an index reduces the search speed, to $\mathcal{O}(n)$ in the worst case, compared to $\mathcal{O}(\log n)$ (on average) for a kD-tree, for n number of points.

This transformation, \mathbf{W} , [Kessy et al., 2018, Stansbury, 2013], is based on the eigenvalues (\mathbf{E}) and a diagonal matrix of the eigenvectors (\mathbf{D}) of the covariance matrix of the radar measurements (\mathbf{Q}_{ENU}) can be found using the equation:

$$\mathbf{W} = \mathbf{E}^{-1} \mathbf{D}^{-\frac{1}{2}} \mathbf{E}^T. \quad (4.1)$$

The effectiveness of this transformation, \mathbf{W} , can be validated by transforming the covariance matrix of the radar measurements (\mathbf{Q}_{ENU}) using the whitening transform to the identity matrix (\mathbf{I}_3):

$$\mathbf{W} \mathbf{Q}_{ENU} \mathbf{W}^T = \mathbf{I}_3. \quad (4.2)$$

In this coordinate system, Nearest Neighbors are nearest in a statistical sense. This search is optimal with respect to the radar, considering AHN as deterministic ground truth, without any statistical variability. This assumption is justified due to the (relatively) small error of the LiDAR point cloud.

This search is implemented using `pykdtree`, a kD-tree implementation in Python [Nielsen et al., 2017, Python Development Team, 2017].

4.3. Linking to a single surface

The surface of AHN points in the neighborhood of a PS-InSAR location is locally approximated as a single, three dimensional, plane of the equation:

$$a\hat{x} + b\hat{y} + c\hat{z} = d, \quad (4.3)$$

where coefficients (a, b, c) represent a normal vector to the plane on the axes \hat{x} , \hat{y} and \hat{z} , and d is a constant.

Two approaches were chosen to approximate the surface:

- Using three LiDAR points, the coefficients are defined by the cross-product of the coordinates of these points.
- Using all Nearest Neighbors found, employing Principal Component Analysis (PCA) to find the coefficients of the plane.

The first method (cross-product) is computationally light but the plane is based on three points only and does not exploit the redundancy in the LiDAR point cloud and does not provide a quality metric for the fit. PCA requires more computational effort but exploits the redundancy in the data and provide a quality metric for the fit.

The error model of both data sets is taken as the starting point of the fusion. To estimate the plane, first the covariance matrix, \mathbf{Q}_{plane} , of the point coordinates \mathbf{x} is calculated for the LiDAR points found,

$$\mathbf{Q}_{plane} = \frac{(\mathbf{x} - \mu_x)^T \cdot (\mathbf{x} - \mu_x)}{n - 1}, \quad (4.4)$$

with μ_x the mean of the coordinates (\mathbf{x}) and n the number of points (≥ 3). The whitening transform (\mathbf{W}_{AHN}) based on the covariance matrix of AHN (\mathbf{Q}_{AHN}) is calculated based on equation 4.1. The covariance matrix (\mathbf{Q}_{plane}) is then transformed using the whitening transform found to \mathbf{Q}_W ,

$$\mathbf{Q}_W = \mathbf{W}_{AHN} \mathbf{Q}_{plane} \mathbf{W}_{AHN}^T. \quad (4.5)$$

Subsequently the eigenvectors and eigenvalues of the covariance matrix (\mathbf{Q}_W) are determined. The eigenvector corresponding to the smallest eigenvalue of the covariance matrix represents the normal vector (a, b, c) . The ratio between the eigenvalues is an indicator for the planarity of the surface, for flat surfaces the smallest eigenvalue is much smaller than the other two, [Hackel et al., 2016]. This results in a ratio, γ , between the eigenvalues ($\lambda_{1,2,3}$) close to 1,

$$\gamma = \frac{\lambda_2 - \lambda_3}{\lambda_1}, \text{ with } \lambda_1 \geq \lambda_2 \geq \lambda_3 \geq 0. \quad (4.6)$$

The constant d of the plane equation (4.3) is found by solving the equation for the mean of the coordinates of the points (μ_x , μ_y and μ_z), with constants a , b and c from the normal vector,

$$d = a\mu_x + b\mu_y + c\mu_z. \quad (4.7)$$

To determine the point of intersection, the PS-InSAR whitening transform (\mathbf{W}) is applied to the PS-InSAR point (\mathbf{y}) and the surface normal vector ($\mathbf{v} = [a \ b \ c]^T$) found,

$$\mathbf{y}_W = \mathbf{y} \mathbf{W}^T, \quad (4.8)$$

$$\mathbf{v}_W = \frac{\mathbf{v} \mathbf{W}^{-1}}{\|\mathbf{v} \mathbf{W}^{-1}\|}. \quad (4.9)$$

The \mathbf{y}_W and \mathbf{v}_W found represent the PS-InSAR measurement and the normal vector of the plane in the whitened space.

The PS-InSAR point, y_W , is projected orthogonal on the surface, the distance found, $d_\sigma(y, \mathbf{v})$, is the distance in σ of the radar error model,

$$d_\sigma(y, \mathbf{v}) = y_W \mathbf{v}_W - d, \quad (4.10)$$

with d the plane constant, determined in equation 4.7.

The point to plane projection, y_{P_W} , is then found based on the distance, $d_\sigma(y, \mathbf{v})$, and the normal vector of the plane, \mathbf{v}_W :

$$y_{P_W} = y_W - d_\sigma(y, \mathbf{v}) \mathbf{v}_W, \quad (4.11)$$

where both point and plane are still in the whitened space. Only after application of the inverse of the whitening transform, \mathbf{W}^{-1} , the point of intersection in world coordinates, y_P , is found:

$$y_P = y_{P_W} \mathbf{W}^{-1T}. \quad (4.12)$$

4.4. Linking to multiple surfaces; dihedral and trihedral geometries

Detection of multiple surfaces can generate a *top-3* list of candidate intersections, such as the horizontal surface slightly further away in figure 4.1B. Detection of multiple surfaces was planned to be based on RANSAC (*random sample consensus*) and only to be applied if the surface estimated using the single surface estimating technique indicated a non-flat area. Initial testing showed that RANSAC was unsuitable for the varying point densities of the point cloud. Successful matches could only be made on two (or more) horizontal surfaces, such as the small height difference between a street and a pedestrian walkway. The vertical surface in between would not lead to a plane estimate.

In practice ensembles of multiple surfaces exhibit joint scattering behavior, with each surface contributing to the final scattering pattern. For dihedral and trihedral configurations this behavior is well defined and consistent, even with slightly varying satellite positions, and therefore constant over time. The dihedral and trihedral configurations of surfaces can be found in the point cloud. To respect the radar scattering behavior of dihedral and trihedral structures the source of the deformation signal should be placed at the intersection of the surfaces rather than at the surfaces themselves [Richards et al., 2010].

Due to the low vertical coverage of AHN, structures have to be fitted based on a limited amount of points in either of the planes. Methods like RANSAC would suffer from the low point density on facades, making the chances of finding the vertical surface 'at random' low. Attempts at using a Gauss map (a projection of the normals on a half-sphere) were unsuccessful due to the smoothing effect of the low point density during the calculation of the normals, creating non distinctive clusters. One such example can be seen in figure 5.15.

As a compromise a method for the detection of right angles was constructed based on an iterative solution, where the density difference between the planes is exploited. The (horizontal) ground plane will match many points and only few points are necessary to match the vertical surface(s). If an orthogonal structure is present, the eigenvectors of the covariance of the points will align this structure. Those eigenvectors are used to simplify this problem to 2D, and improve redundancy of the data.

The fitting procedure is as follows:

1. For each geolocation estimate the closest 100 points within 2 meters are found. This is done in world coordinates, without taking the covariance properties of either point cloud in consideration. As this procedure is on the reconstruction of the feature, rather than statistical proximity.
2. The eigenvectors of the covariance of the point coordinates are determined and the data is projected on the two last eigenvectors (smallest eigenvalues).
3. Two orthogonal lines are initialized, intersecting at the mean coordinates of the cloud. Initialization with the original geolocation estimate would be possible, but this prevents all points from being closest to a single plane on initialization.

4. Iteratively the planes are moved with the mean offset of the points they are closest too and subsequently rotated to come closest to the neighboring points. These steps repeated 5 times (5 translations and 5 rotations). Stopping the iteration on stabilization is possible, and currently not implemented, as this solution is only to be run if the others were unsuccessfully.
5. The point of intersection resulting after the iterations is transformed back to world coordinates and returned. The distance in sigma is calculated by transforming both the original coordinates and the point of intersection using the whitening transform and calculating the Euclidean distance between the two.

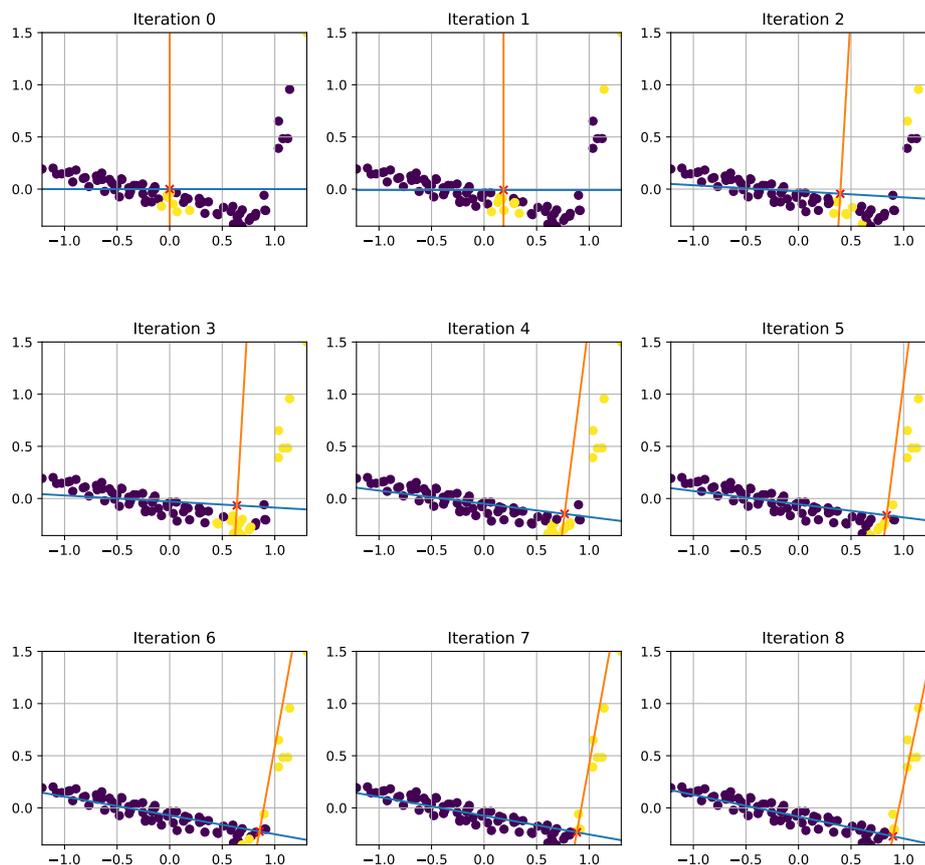


Figure 4.3: Iterative algorithm to find the corner of a dihedral or trihedral reflector. Points are projected on the eigenvectors to reduce the dimensionality of the problem and to mitigate the effect of sampling density differences. A corner is matched in a K-Means like approach: a corner model, with a straight angle, is translated and rotated iteratively to match the inliers (blue or yellow) with the shortest possible distance.

This procedure is illustrated in figure 4.3. The input data can be found in figure 5.15, including the (estimated) normals as vectors and in a Gauss map. Due to the smoothing at the edge (line of intersection) the normals do not show a straight corner. Nevertheless the iterative method is able to find this line.

4.5. Quality attributes and indicators

Although automatic quality assessment will filter gross outliers, a quality estimate of the intersection should be shown to the user. Quality indicators include the distance to the point of intersection and the properties of the plane. These indicators can be a combination of Euclidean and covariance metrics. Furthermore derived units, such as the orientation of the plane with respect to the satellite viewing

geometry, indicative of the likelihood of a strong reflection, could be implemented. These metrics might be used in aggregate products to, e.g. the detection of a regional offset or offsets for specific types of scatterers.

The matching process itself might be validated based known reflectors, with known position (e.g. corner reflectors with accurate GNSS measurements). Although new reference reflectors will not be available in AHN3, as acquisitions are finished for most areas. It is easier to validate on facades, at the intersection of the facade with the street forms a natural dihedral reflector. These positions can be validated afterwards or compared with maps from different sources.

This method works best when the facade faces the radar (figure 2.1, RCS as function of aspect). Cross-facade alignment might prove difficult with this method, as the dihedral reflector shows similar behavior over the full length. To validate this last dimension trihedral reflectors are preferred.

Assessment of the scattering behavior might be done by the addition of photographs to the 3D point cloud. Such combination shows the facade as an easy to interpret photo rather than a full point cloud, with high vertical density. As an alternative high resolution scans from terrestrial or mobile platforms might be added as comparison or included in the input data to improve performance on small features. The classification present in AHN, might be used to assess the type of object the plane or geolocation matched to. Combinations of ground and building are likely, while a persistent scatterer on the interface between a facade and water is unlikely to be consistent over time.

5

Results & validation

5.1. Web viewer

A screenshot is shown in figure 5.1, a zoomed in version in figure 5.2. A live demonstration can be found at <http://dev.fwrite.org/radar/>. The determined intersection is shown as the intersection of the ellipsoid with the estimated (single) plane, as can be seen in figure 5.2.

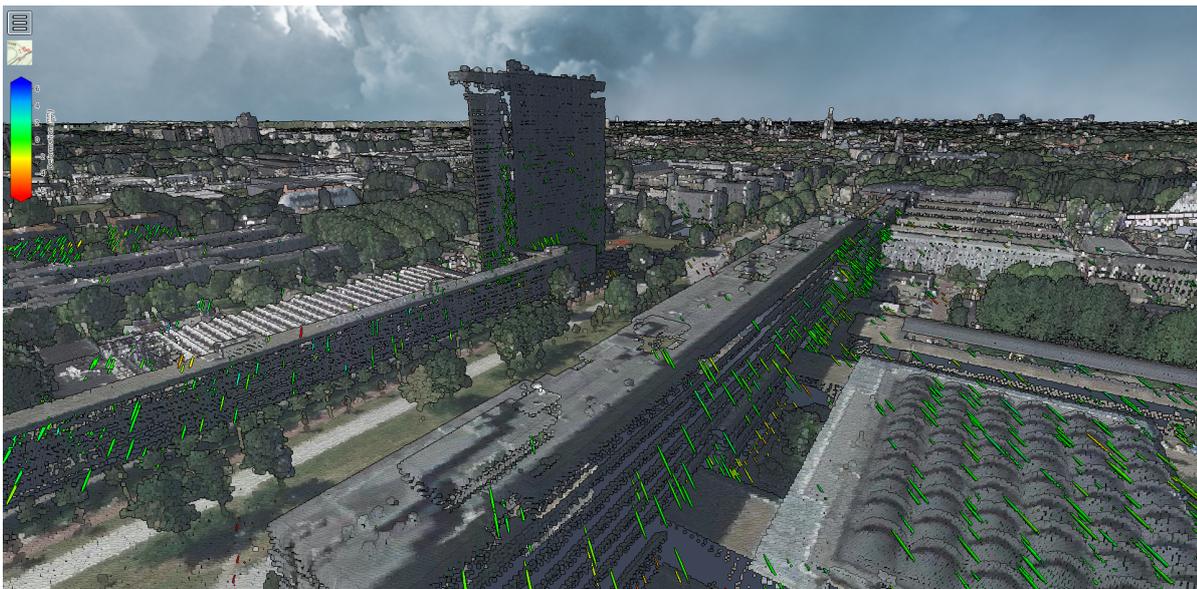


Figure 5.1: Delft University of Technology campus, shown in an adapted version of the Potree viewer, as a colored point cloud (AHN3) overlaid with TerraSAR-X PS-InSAR trend estimates shown as 1σ error ellipsoids around the expected position, colored by their deformation signal.

Figure 5.2 and the image on the cover can be compared to a regular photo in figure 5.9. As only orthophotos were used instead of a combination with oblique images facades are colored as either roofs or surrounding ground surfaces. Facades may be added to the viewer as photographs. An example can be seen in figure 5.3. Photos were taken by the author and perspective corrected. No correction for lens distortions was applied.

Another example of the combined visualization of the point cloud with deformation data is the visualization of a dike in the Eemshaven, as can be seen in figure 5.4. In this image the PS-InSAR deformation signal from the dike is interpolated, exaggerated and applied to the existing AHN2 point cloud. The true line of sight deformation can be shown in a 3D point cloud, unlike in 2D maps.

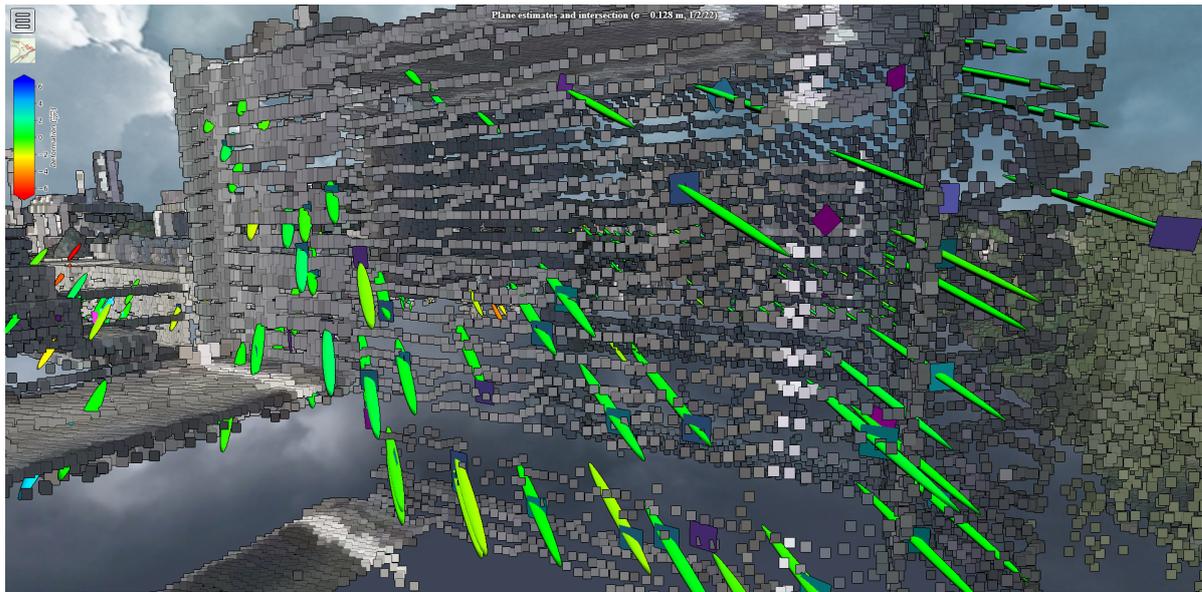


Figure 5.2: Part of the 3ME faculty on the Delft University of Technology campus, shown in an adapted version of Potree, as a colored point cloud (AHN3) overlaid with TerraSAR-X PS-InSAR trend estimates. Indicated are the 1σ error ellipsoids around the expected position, colored by their deformation signal and the planes of intersection estimated from the point cloud.

5.2. Matching

To improve the geolocation estimate of PS-InSAR measurements the original geolocation estimates are matched to the point cloud. Various steps in this integration were introduced in chapter 4. To assess the differences between them all methods were applied to all points, therefore the results of all methods can be compared.

Comparison between matching techniques

A comparison between the various matching techniques of the distance from the original geolocation estimate to the point or intersection found in the point cloud is given in the histograms of figure 5.5. Local reconstruction of the geometry, by surface approximation, leads to lower distances between the original geolocation estimation and the surface found, as sketched in figure 4.1. The intersection with the surface is on average $\frac{1}{2}\sigma$ closer than the first Nearest Neighbor. Due to the anisotropy of the problem this distance can not be uniformly expressed in meters.

The improvement in location can be seen in figure 5.7. Compared to figure 5.6 stable points are now attributed to the facade while subsiding points remain on the street. This subdivision is to be expected given the stable foundations of the building but had to be made manually by a skilled operator on traditional maps. Another example can be found in figure 5.8.

Of the total of 3.1 million PS-InSAR points, less than 20% of the points did not match a Nearest Neighbor within $2\frac{1}{2}\sigma$. For surfaces the results are slightly better: 85% of the points were linked to a nearby surface within $2\frac{1}{2}\sigma$. These results are summarized in tables 5.1 and 5.2. Results further away are very unlikely, given the validity of the error model. Missing links are generally due to occlusions in the point cloud, for example on facades and in narrow streets, resulting from the different viewing geometries between the sensors. Some of them are due to faulty interpretation of the geometry, leading to plane estimates that do not provide a realistic point of intersection.

Corner reflector results

Matching aimed at dihedral and trihedral structures showed poor results, with unrealistic error estimates. A full comparison between the methods can be seen in figure 5.10, which shows planes outper-

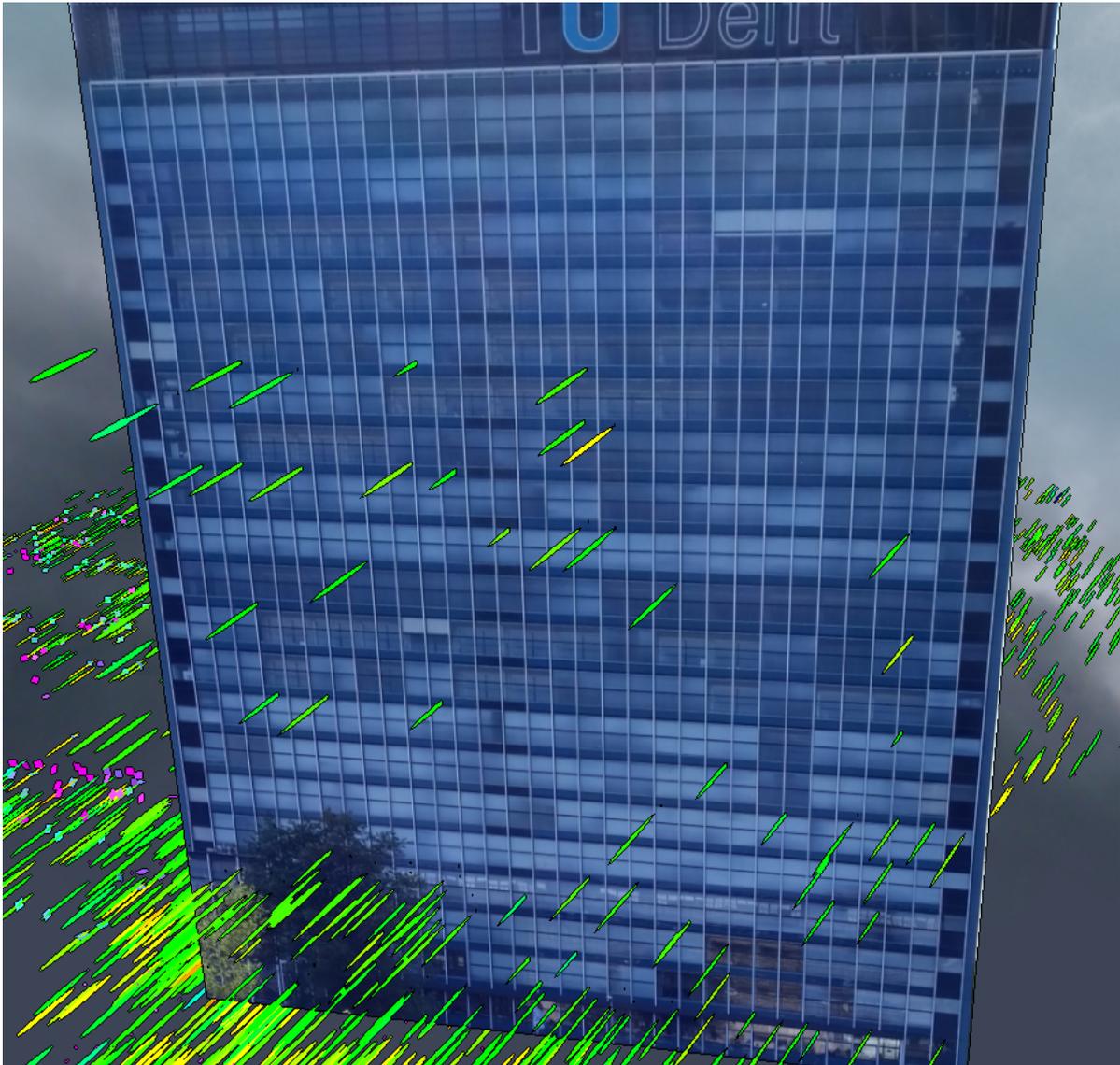


Figure 5.3: Faculty of Electrical Engineering, Mathematics and Computer Science. Shown as a (regular) photograph in 3D space in Potree, together with the TerraSAR-X PS-InSAR geolocation estimates. A regular scattering pattern can be distinguished at the intersection of the floors and the vertical profiles. Such visualization aids in the interpretation of the physical properties of radar scattering.

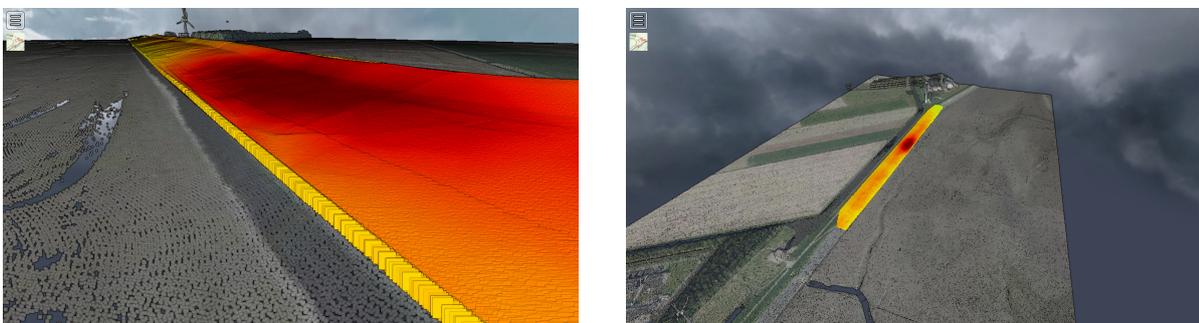


Figure 5.4: Deformation of a dike near Eemshaven, interpolated and exaggerated deformation of the AHN2 point cloud, based on the PS-InSAR deformation trend estimates. Left: close-up of the dike section showing the strongest deformation signal. Right: overview of the surrounding fields and part of the dike.

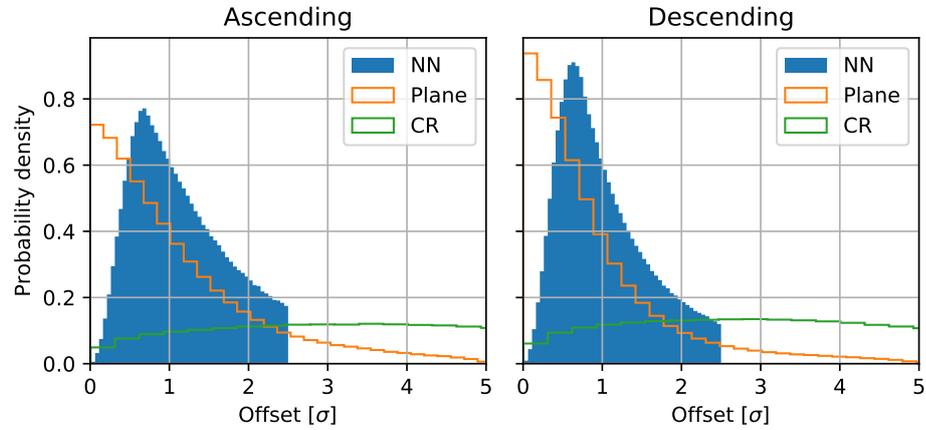


Figure 5.5: Distance to the closest object with various matching techniques. Plane estimation outperforms the Nearest Neighbor (NN) search, which shows a bias due to the sampling distance of the points. The Nearest Neighbor is on average at least half the sample spacing away from the initial estimate. Corner matching (CR) performs the worst of all methods in terms of distance and bias. Nearest Neighbor search is stopped at $2\frac{1}{2}\sigma$. No points further away than 2 m are included in the search for a corner reflector.

Descending	Nearest Neighbor	Plane	Dihedral
μ_σ	1.01	0.89	4.75
%	80 ($\leq 2\frac{1}{2}\sigma$)	91 ($\leq 2\frac{1}{2}\sigma$)	84 ($\leq 2m$)

Table 5.1: Comparison of matching results, for the descending orbit PS-InSAR data. The average distance (μ_σ , measured in σ of the PS-InSAR geolocation estimates) is shown together with the percentage of points that could be matched. Planes outperform the other methods in number of matched points (within cut-off) and the distance to the nearest feature.

Ascending	Nearest Neighbor	Plane	Dihedral
μ_σ	1.13	1.11	5.28
%	75 ($\leq 2\frac{1}{2}\sigma$)	89 ($\leq 2\frac{1}{2}\sigma$)	76 ($\leq 2m$)

Table 5.2: Comparison of matching results, similar to table 5.1, but for the ascending orbit. Again planes outperform the other methods in number of points matched (within cut-off) and the distance to the nearest feature, although the difference is less than with the descending orbit.

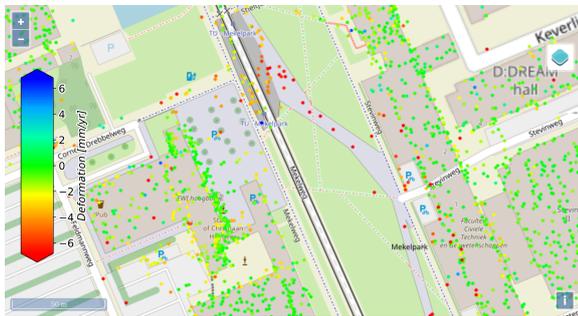


Figure 5.6: Traditional PS-InSAR deformation map of the TU Delft campus, colored by deformation velocity. Misalignment occurs at the building facades. (Background: OpenStreetMap.)

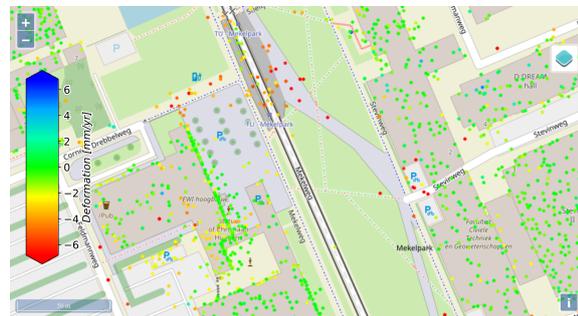


Figure 5.7: PS-InSAR geolocation estimates and velocities after matching between the data sets. Improvement is mainly visible at facades. (Background: OpenStreetMap)

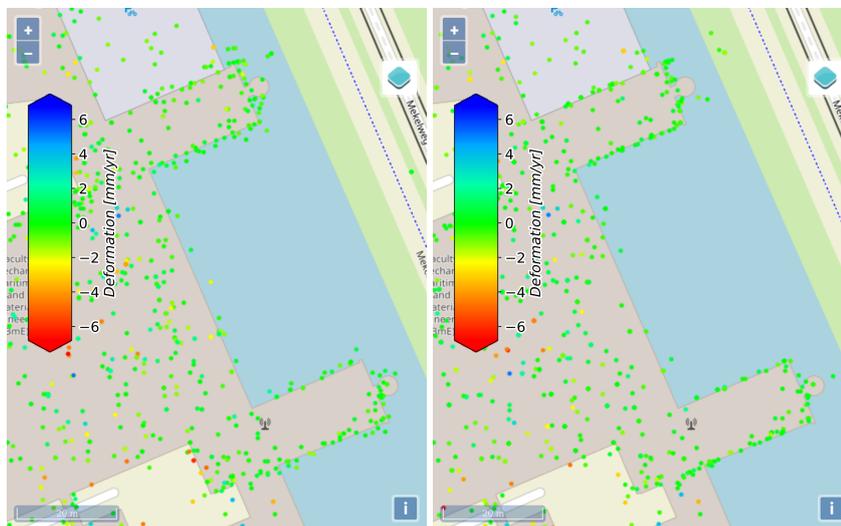


Figure 5.8: Geolocation estimates, mapped in 2D, on the 3ME faculty (compare also figure 5.2), before (left) and after (right) matching between the data sets. Improvement is visible at the facades, changes on the roof are related to the roofing structure.



Figure 5.9: Regular photos of the Mekelpark (left) and the 3ME faculty (right). The same scenes are featured in the PS-InSAR deformation velocity maps in figures 5.6, 5.7 and 5.8.

forming the nearest neighbor (NN) in proximity when measured in standard deviations of the original geolocation estimate. Corner matching (CR) never outperforms plane estimation, not even for very non-flat surfaces.

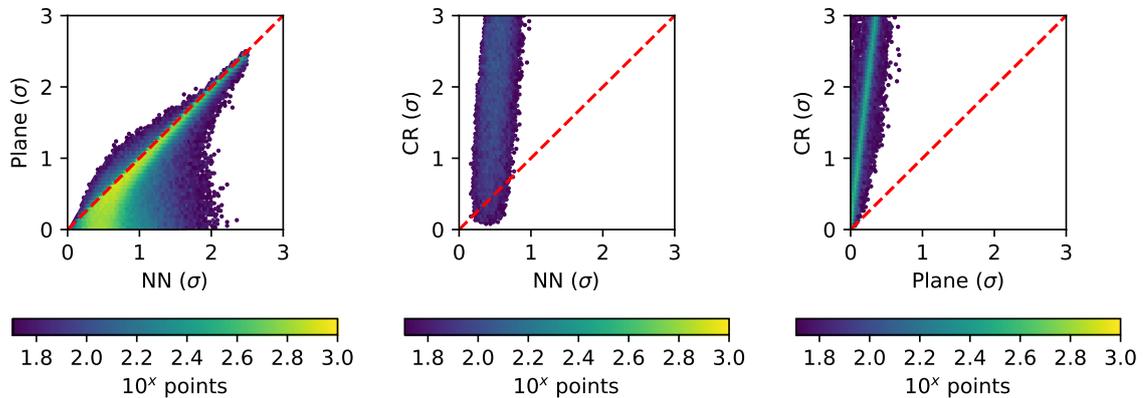


Figure 5.10: Distance (in σ) between the original geolocation estimate and the nearest neighbor (NN), plane or modelled corner reflector (CR). Shown in red is a 1:1 relation, points under the line are closer to the method listed on the y-axis, points over the line are closer to the method on the x-axis. Left: plane matching outperforms the Nearest Neighbor (NN) mostly at close range, where the sampling distance of the point cloud is of influence. Middle and right: the corner matching (CR) may outperform the Nearest Neighbor at short distance, but will never outperform the plane estimate.

Two natural corner reflectors can be seen in the staircases highlighted in figure 5.11. Although (very) similar their representation in the point cloud is different, as shown in figure 5.14 and figure 5.15. Both are poorly present in the point cloud, with only some points on the vertical walls, illustrating the problems encountered in corner matching.



Figure 5.11: An example of large natural dihedral and trihedral reflectors on the Stevin 3 laboratory of the CEG faculty, formed by the stairwell and attached floors. (Image: Google Maps) Their representation in the AHN3 point cloud is shown in figures 5.14 and 5.15.



Figure 5.12: Roof of the Aula, with non straight corners that may falsely be identified as dihedral reflectors. (Image: Google Maps)

Another example is the roof in figure 5.12, that contains many persistent scatterers but does not consist of dihedral reflectors. The roof segments (figure 5.13) are not at straight angles. The matching algorithm is not capable of matching those structures.

Comparison with error estimation

Over the whole data set biases are in the order of decimeters, with standard deviations of multiple meters, as can be seen in figure 5.16. When expressed in radar coordinates the uncertainty in matching corresponds to the expected geolocation error, estimated by Dheenathayalan et al. [2016]. The expected geolocation standard deviation was 0.128/0.256/2.816 meter in range/azimuth/cross-range (section 2.1). As can be seen in figure 5.17, the standard deviations are of the same order of magnitude as the original estimations. In range and azimuth the error is overestimated, while the cross-range estimate is of the same order as the original estimate. Covariance between the components is one order of magnitude lower than the variance of the range, azimuth and cross-range. The initial assumption

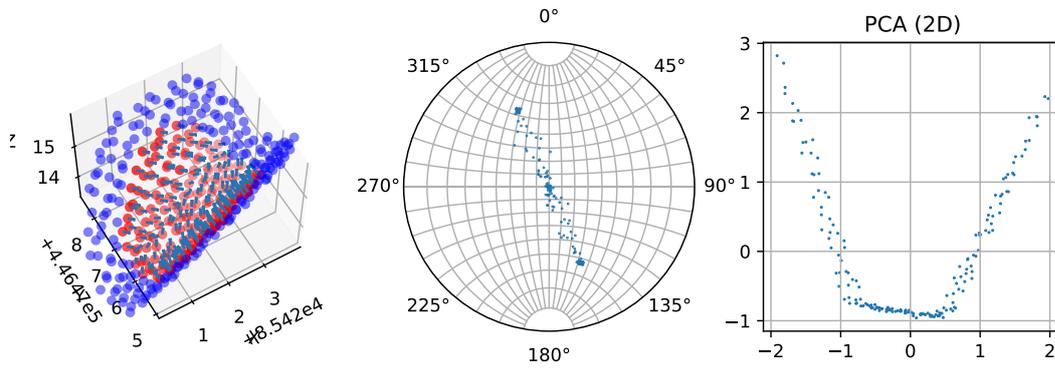


Figure 5.13: An example of a non straight angle, on the roof of the Aula (see figure 5.12), analyzed using a Gauss Map and projected on the eigenvectors. Left: the points used as input to the angle estimation (red) and some surroundings (blue). Thanks to the high point density on the surface, representative normals could be estimated. Middle: a Gauss map of the direction of the normals, showing, first, that the surfaces are not at 90 degrees angles and, second, the *clutter* introduced by the normal estimation at the corners. Right: a projection along the eigenvectors, clearly showing the angle.

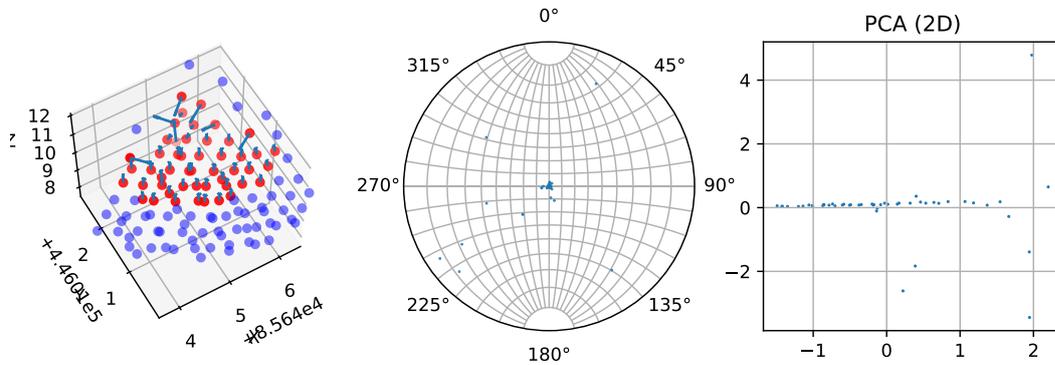


Figure 5.14: Natural trihedral reflectors on the Stevin 3 laboratory of the CEG faculty (see figure 5.11), southern staircase. Analyzed using a Gauss Map and projected on the eigenvectors. Left: the points used as input to the corner estimation (red) and some surroundings (blue), due to the low point density on the facades the normals are strongly influenced by the horizontal surface. Middle: a Gauss map, showing that very few normals represent a (near) vertical surface. Right: the corner as projected on the eigenvectors. Unfortunately projecting on the eigenvectors did not provide a better analysis of this angle.

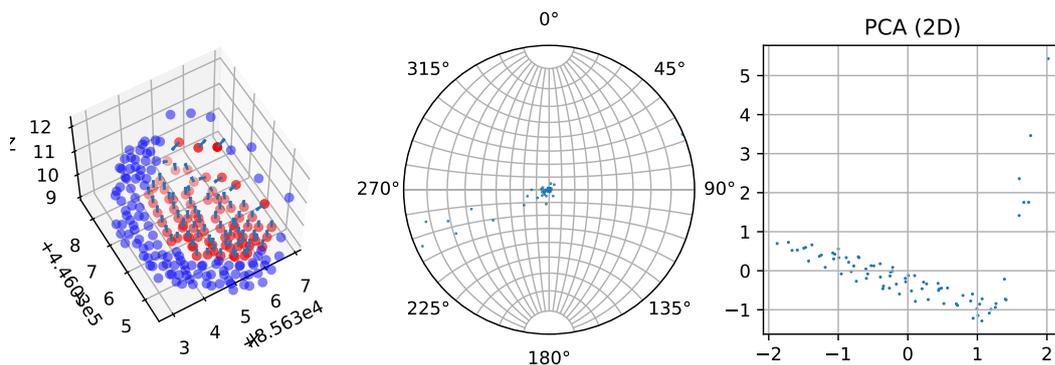


Figure 5.15: A view of the northern staircase of the Stevin 3 laboratory. Even with poor normals (left and middle), the projection on the eigenvectors is successful in detecting the corner (right). The dihedral matching algorithm (section 4.4, figure 4.3) was demonstrated on this data.

that the covariance can be neglected, is therefore a reasonable simplification. This is all under the assumption that the first Nearest Neighbor is the origin of the backscattered signal.

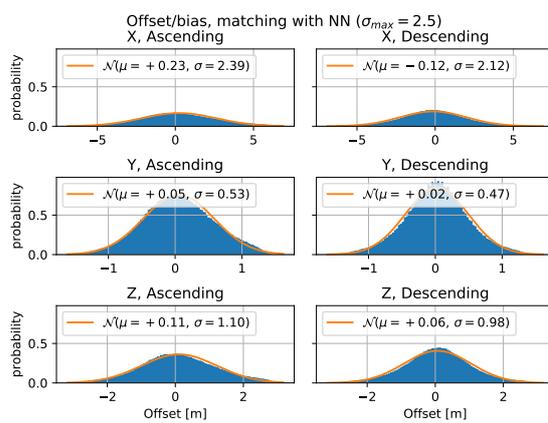


Figure 5.16: Offsets (biases) in matching, in x, east; y, north and z, up, for both ascending and descending orbits, based on the first Nearest Neighbor, if found within $2\frac{1}{2}\sigma$. Shown on top is a fitted normal distribution.

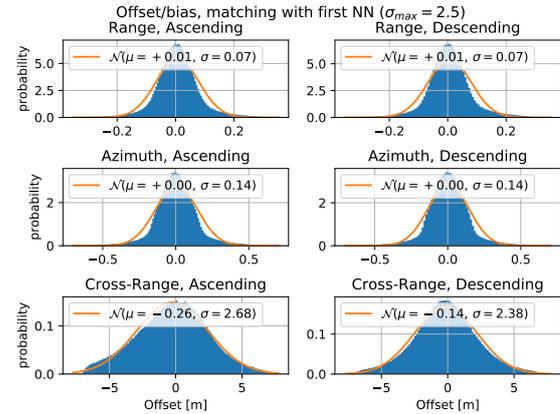


Figure 5.17: Same as figure 5.16, but in radar coordinates: range, azimuth and cross-range. Matching performs better than the fitted normal distribution, which is in turn better than expected from the original estimates used as input for the matching.

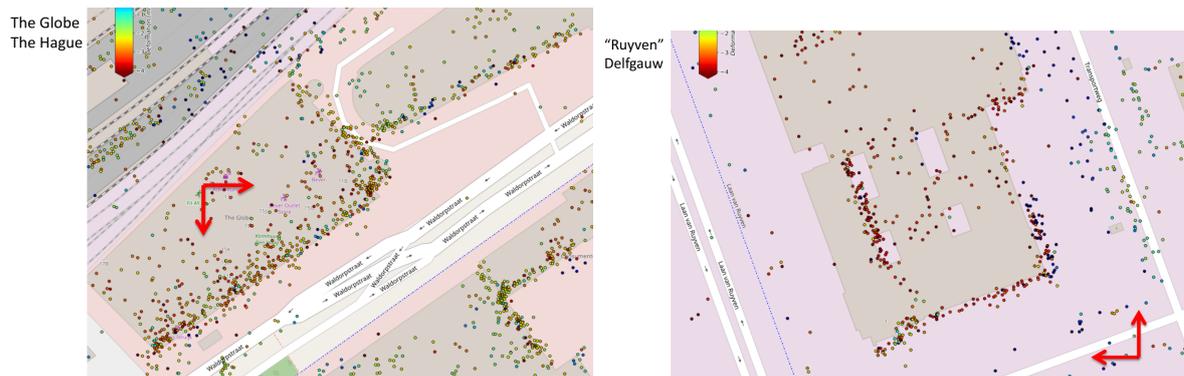


Figure 5.18: Examples of the regional offset visible in the radar data, shown over The Globe in The Hague (north) and on the industrial estate 'Ruyven' in Delfgauw (south). (Background: OpenStreetMap.) Due to the trend in the data, points on the street are attributed to the building in The Hague, while points on the building are projected on the street in Delfgauw.

Quality estimation

Various quality estimates result from the matching procedure. For each point the distance from the original estimation is known, both measured in standard deviation (σ) and in world coordinates (meters). Furthermore for the surface estimation quality metrics are available, such as the flatness of surface (quality of the approximation), that indicate the quality of the fit for each individual surface.

The solution can be compared visually in the point cloud, possibly aided by the addition of a high resolution photograph. In figure 5.3 a regular scattering pattern can be seen on the facade, most persistent scatters seem to reside where the floors intersect with the glass facade.

Processing

Matching 1.4 million SAR points to 3 billion LiDAR points (122 tiles) using the Nearest Neighbor approach takes 15 minutes with three threads on a quad-core Intel i7-3630QM 2.4 GHz laptop computer with 24 GB of RAM. Approximately 95% of this time is spend on opening the compressed LAZ-files, transforming the points (using the whitening transform) and building the KD-tree. As the current script

is written in python 3.5, higher performance is expected to be achieved by using more optimized programming languages. The addition of more complex methods will slow the process down. The iterative detection of corners took 45 minutes to complete on all points. Plane estimation took almost 1:15 hour, due to the necessity of a double search in two kD-trees (transformed and original coordinates). As no intermediate results are stored, processing of the other orbit (ascending/descending) will take approximately as long, and effectively double the processing time. Typically more complex methods are only necessary after their faster counterparts had insufficient results, integrating the steps would require reading and indexing the data only once, speeding up the process.

With the research area spanning 122 tiles of a total of 30137 tiles, processing the nationwide data set will take around sixty hours on this laptop. Given that the program can be scaled over many nodes the computation time can be reduced to the duration desired by the addition of more computing power.

5.3. Regional trends

Although the average offset is almost zero (figure 5.16), subsets of the data may be biased. By comparing the difference between the point of intersection or Nearest Neighbor and the original geolocation estimate with respect to location, trends become visible.

In figure 5.19 the median offsets between the geolocation estimation and Nearest Neighbor are shown (NN - estimation), binned in 100×100 meter bins. The same offsets are shown in radar coordinates in figure 5.20. A clear north-south trend can be distinguished in the descending orbit, while in the geolocation of the ascending track the offset increases in the southern part. Due to this trend geolocation estimates are off by 2 to 3 meters at the borders of the radar image. In figure 5.18 two examples at a local (building) scale can be seen.

These offsets and trends, that can be converted to radar coordinates too, may help to improve the radar processing.

5.4. Discussion

This research is focused on the geometric alignment of the PS-InSAR measurements with a point cloud, physical processes governing the radar behavior were not studied in detail. Another approach to the improvement of PS-InSAR geolocation estimates would be to take the physical behaviour as a starting point of the alignment. Nevertheless the visualization and large scale geometric alignment of data aid the interpretation by radar professionals of both deformation behaviour and physical processes influencing the radar signal.

The following other topics, related to the geometric approach, were addressed in this thesis, but require some elaboration.

Error model

Numerical work in this thesis is based on two assumptions:

1. There is only one error model for the radar data;
2. The error introduced by AHN can be neglected.

These assumptions might oversimplify the problem, the consequences will be discussed.

Single error model

In this thesis the geolocation error model is taken as constant over the whole region of the radar image and for all scatterers. This assumption is incorrect, as indicated by Dheenathayalan et al. [2016]. In reality each point has its own geolocation precision estimate. The incidence angle ranges from 20 to 45 degrees over the image and reflectors vary in type: from (round) lamp posts to trihedral reflectors, each with different properties and associated (different) geolocation error estimations. Unfortunately

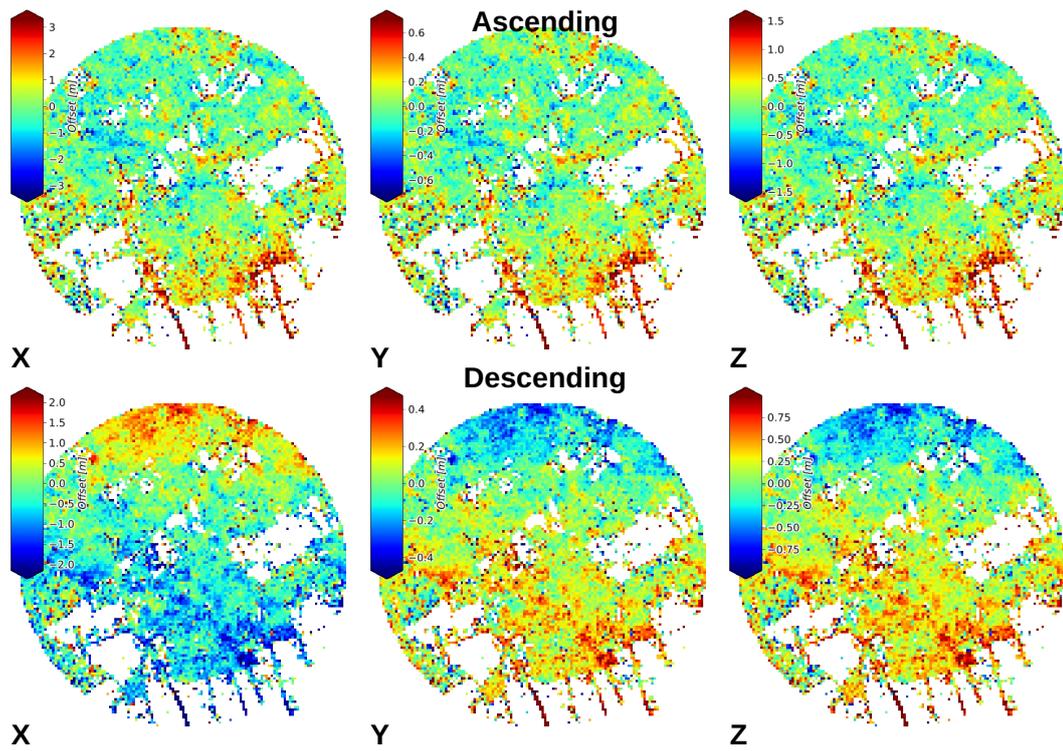


Figure 5.19: Regional trends (100 × 100 meter median) in the offset between the geolocation estimation and the Nearest Neighbor (NN - estimated) for both ascending and descending orbits. Offsets in X (North) direction are largest, showing ± 2 to 3 meters offset at the boundaries.

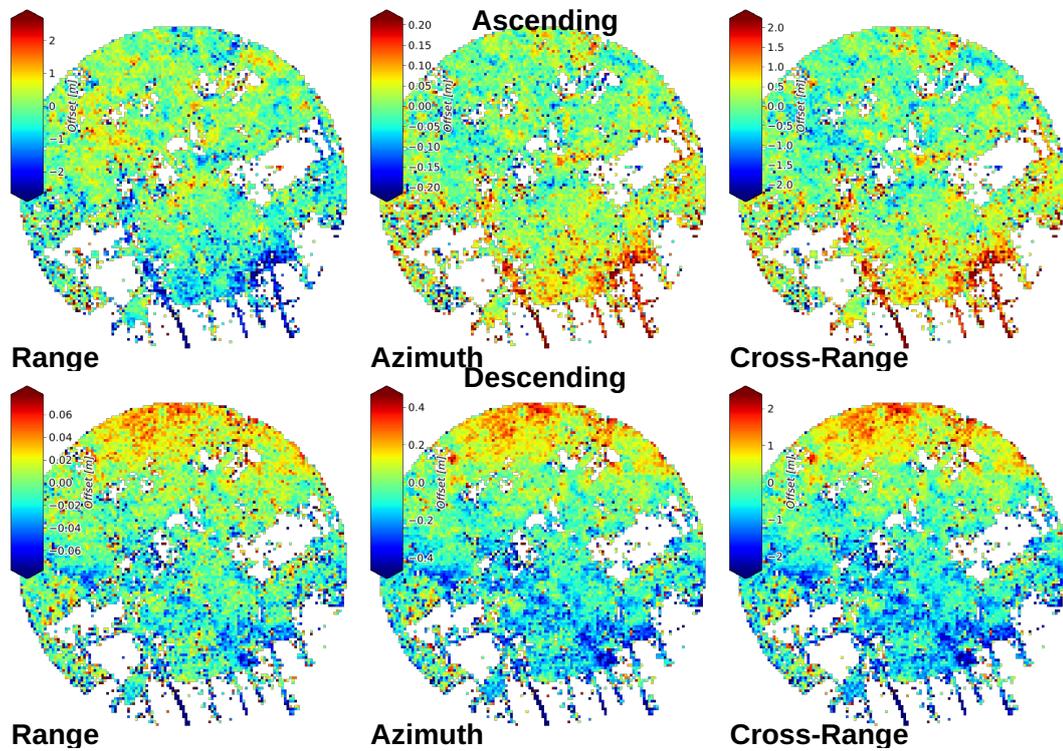


Figure 5.20: Similar to figure 5.19, but offsets shown in radar coordinates. Although the matching is unbiased, a trend with correlation between the axes can be seen.

this information was not available in this study. Therefore a single error model of a lamp post was assumed for all scatterers.

Applicability of the whitening transform

The decorrelation method, chosen for matching LiDAR and PS-InSAR, the *whitening transform*, is only suitable for a single covariance matrix. Different scattering regimes, with different statistical regimes, would each require the transformation of the original point cloud data and the subsequent creation of a search structure.

For regional variations, such as the incidence angle, the transformation can be parameterized. Although no longer linear, it is still be approximately valid given that variations that only occur at a regional scale do not influence close range approximations, a traditional nearest neighbor search could still be applied then.

A solution would be to use a search structure independent of the covariance structure. Either brute force, using all points in a sphere around the estimated position, or by querying small parts of the point cloud, approximating the (ellipsoidal) shape of the covariance structure. For this application a search structure such as a space filling curve could be used to query the point cloud with irregular and changing patterns.

Formal error propagation

Currently no overall error model is implemented. The quality of the surface fit and the quality of the intersection are defined and estimated independently as two separate metrics. An integrated metric could provide, for example, a new quality metric for the point of intersection found. The complexity of such metric is illustrated in figure 5.21, where there are areas of equal, maximum, probability, rather than single points. The same holds for the probability of the intersection with surfaces, where the surface is not uniquely defined and crosses the error ellipse of the PS-InSAR geolocation estimate at various points.

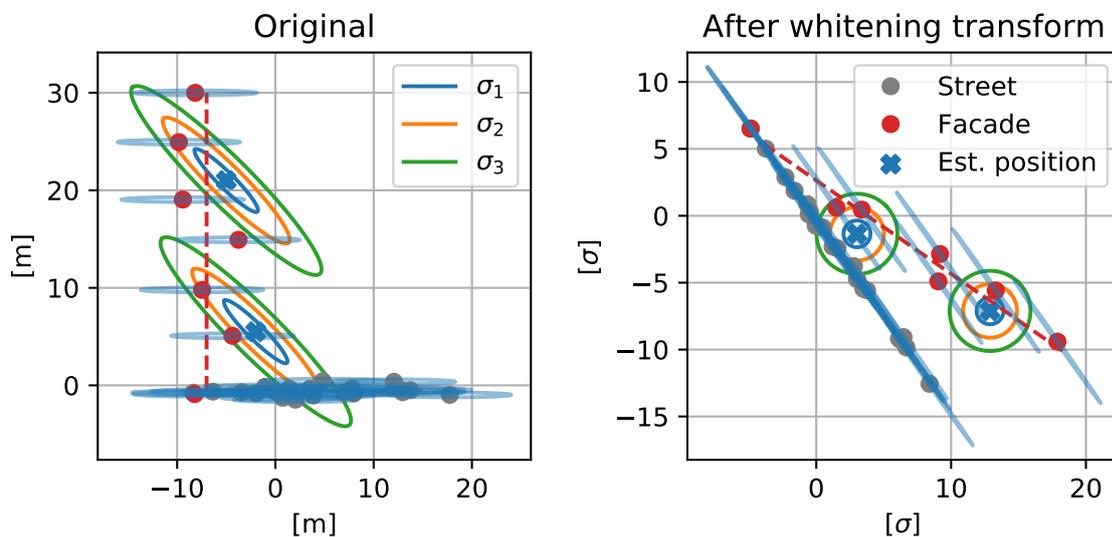


Figure 5.21: Adapted figure 4.2, to include the effects of noise on the point cloud. Single points have an area of equal probability rather than a single point. There is no longer a single, unique, surface estimate, creating a complex *shape* of equal probability. The image is not to scale with the error description of AHN or the PS-InSAR geolocation estimates.

A more formal approach to error estimation would provide more information on the quality of the match. Given the different matching techniques, there might not be one single error model, and the error models are unlikely to be normal distributed. This makes it difficult, if not impossible, to compare between matching techniques.

Iterative bias correction

The radar ellipsoidal confidence interval is a simplification, based on the assumption that the processing parameters were correct. In reality this confidence interval has a more complex shape. Assessment of alignment could be integrated to improve the assumptions in the radar processing too.

To improve the matching results the algorithm could be re-run with the biases obtained from the first run after initial matching. After correcting for the bias or trend in the initial geolocation the results of the second geolocation matching algorithm may find a different (possibly better) point of intersection. Quality assessment for this iterative process will be difficult, as the new geolocation will be the result of multiple transformations of the original data set.

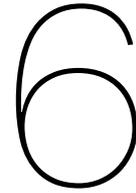
Size of persistent scatterers

In this thesis persistent scatterers are linked to single points or coordinates, this implies the assumption of an infinitely small scattering object. In practice this assumption holds for the apex of a trihedral reflector and to a lesser extent for the corner of a dihedral reflector. Other reflectors, such as planes, do not have such a point.

The minimum length of the sides of a trihedral reflector is only 50 cm for X-band radar (such as TerraSAR-X) to be strong enough to be used in deformation measurements, [Garthwaite et al., 2015]. Many natural scatterers are formed by larger features and can be detected in the point cloud, but this high level of detail may not be conveyed in the AHN point cloud, as points on all three surfaces are necessary to reconstruct the trihedral shape. Addition of other, high resolution, point cloud data, such as from terrestrial or mobile scanners might improve this detection at the cost of increased processing requirements associated with matching such small features.

Matching applicability estimation

'Normal maps' could be created from the point cloud, showing the orientation of the surfaces represented. This data could help in the assessment of the usefulness of this matching technique before radar images are commissioned or to determine where an extra LiDAR survey would be most efficient. Novellino et al. [2017], for example, used a DEM, land cover and other constraints to predict the effectiveness of Sentinel 1 for deformation monitoring purposes. In the built environment maps could provide a resource on the (expected) orientations of facades. An example of such analysis is shown in figure A.4, that shows the direction distribution of streets (and likely parallel facades) is strongly dependent on the city.



Conclusions

This thesis was written around the main question:

How to improve geolocation of InSAR point scatterers using detailed 3D point clouds?

Various options exist for the improvement of the geolocation estimation using point clouds, using geometric linking of the two data sets. The implemented methods are based on geometry only and may not represent the underlying (physical) radar processes. Nevertheless they improve the geolocation estimate, aid in the attribution of the radar signal to real world objects, support in the assessment of the source of the deformation signal and the validation of the radar processing. This process can be applied at a national scale, as the computations are fast and scaleable across machines.

6.1. Conclusions

This main question was subdivided into the following sub-questions:

1. What aspects of PS-InSAR deformation assessment can be improved by linking to 3D surface geometry?

The techniques introduced enable the efficient attribution of the InSAR deformation signal to real-world objects and features, allowing a spatial join at a higher scale: linking individual signals to objects. The 3D visualization will allow for better communication with the greater public, as less interpretation is required with respect to the traditional deformation maps. For professionals it allows better interpretation of the radar signal, and the processing of artifacts contained therein.

For 85% of the persistent scatterers could be attributed to a nearby surface in the point cloud. This improves positioning, classification and quality assessment ('trustworthiness') of the PS-InSAR data.

2. What statistical model applies to the data, and how was it defined?

The statistical model for the PS-InSAR measurements is defined in radar coordinates, as the precision of range and azimuth follow directly from the sensor and processing properties. Formal error propagation is used to estimate the precision along those axes. The estimated precision of the cross-range follows from the geolocation estimate and is dependant on the time-series and properties of the scatterer. Therefore statistical properties vary per sensor and per scatterer and may change when new radar images are added. No covariance was assumed.

For AHN the error model used is based on the requirements set for the survey, not on values found empirically by analysis of the data. The precision is defined along the three axes of the RD coordinate system as unbiased and normal distributed. Furthermore the vertical precision is simplified by combining the stochastic and systematic errors. This step simplified processing and

accounted for errors between overlapping flight lines. No covariance is assumed between axes or measurements (time).

- Is the statistical model valid?

The PS-InSAR model was validated in literature using corner-reflectors and a lamp post measured using the same sensor (TerraSAR-X) as used in this study. After alignment the model is validated against the offset from the initial alignment. The distribution found is unbiased and of the same order of magnitude as the initial estimate, but contains a regional trend is not described in the model. Covariance of the matching offset is limited and an order of magnitude lower than variance of the principal axes.

For AHN the error estimate was validated on a runway of Schiphol Airport. Over the flat area the data outperformed the error model, although the offset between flight lines is larger than expected. All data was validated before publication by third companies, and were necessary remeasured. Although notable, this result might not be characteristic for the whole data set. Unfortunately the validation reports are not publicly available and the validation methods used unknown.

- Is the data suitable for the planned application?

Yes, given the high precision of the initial PS-InSAR geolocation estimate most (> 85%) scatterers can be attributed to a distinct object in the LiDAR point cloud. However, some particularities are not covered in the error model. The unexpected trend in the PS-InSAR measurements may lead to assignment errors as the distance between the point clouds is no longer described by the error models. Furthermore the point density is not covered in the statistical model, especially on vertical surfaces the density of AHN might be so low that no candidate matches are present. *Suitable data is not necessarily successful data.*

3. How are PS-InSAR and LiDAR point clouds best spatially joined?

- How to represent the point cloud in segments of similar features?

When the point cloud describes a single, flat, surface it can be considered *locally 2D*. More complex geometries, such as corners, are represented as multiple *locally 2D* environments, i.e. planes. This plane approximation circumvents problems related to point density.

Geometries important to radar scattering, for example dihedral and trihedral configurations, can be matched as objects too, although point cloud density differences have to be accounted for in the matching algorithm.

- How to join a persistent scatterer to the surface outlined by LiDAR point cloud data?

The intersection between the plane and the smallest possible error ellipsoid is considered the point of maximum likelihood. Projection is done in the whitened space of the radar sensor, so that the projection is a generic point-to-plane problem. The (orthogonal) projection of the point on the plane is the closest point as measured in the standard deviation of the radar geolocation estimate.

For dihedral and trihedral reflectors the new geolocation estimate is in the corner, or apex, of the reflector. Again the distance from the original solution can be expressed in the standard deviation of the original estimate.

- How to represent the uncertainties of this match?

Multiple sources of uncertainty have to be represented: uncertainty in the PS-InSAR location, uncertainty in the LiDAR point cloud and uncertainty in the approximation of the geometry. Uncertainty is only expressed with respect to the PS-InSAR location, as the uncertainty in especially the cross-range component is much higher than the other uncertainties. Uncertainties are represented in standard deviations from the estimated position of the scatterer, ergo they can be neglected.

4. What is the quality of the alignment between the data sets and can the alignment be improved?

- Is there a constant offset or another bias between data sets? Is this offset local or regional?

Initial alignment is unbiased (zero mean), although a regional trend is present of up to ± 3 meters in the horizontal plane (± 2 meters in cross-range). This regional offset is present in both ascending and descending orbits of the case study, and on average zero in the centre of the data set.

- Can this offset be mitigated, based on the spatial join?

Both a constant offset (given small enough not to match other buildings, etc.) and a trend might be modelled. The model found can be used to mitigate the offset. Matching can be incorporated in various stages of the workflow too: as final (refining) step or iterative within the process to check and adjust the assumptions made during the radar processing.

5. How to scale the matching process to regional or national scale?

Initial matching is an independent process that can be performed for each PS-InSAR geolocation estimate individually. Therefore the process can be scaled horizontally, allowing for many parallel search operations. Data preparation is tailored to this, splitting the data in many chunks that can be distributed over the processing nodes.

The current implementation of the whitening transform is very efficient but requires regeneration of the search structure for each iteration. Although the search structure can be saved, it would only be valid for a single radar viewing geometry. The application of a space filling curve, indexing and ordering the data based on location, will allow a uniform search index that can be stored efficiently and be used by all viewing geometries. The current algorithm can be run in approximately 60 hours for the whole of the Netherlands, on a consumer laptop, including generation of the search structure. Therefore the need for a more efficient algorithm is currently limited, while processing time scales approximately linear ($\mathcal{O}(n)$) with the number of PS-InSAR measurements and $\mathcal{O}(\log n)$ with the number of points in the point cloud.

Data prepared for the point cloud viewer is split by concession. This allows for incremental replacement as new areas become available, reducing the computational cost of updates and allowing parallel creation of the structure.

6.2. Recommendations

Apart from the topics discussed in the discussion (section 5.4) further research could benefit from the following recommendations:

- Use of space filling curves, sorting the data instead indexing, such as with kD-trees or similar methods, might be a faster search method than a single index in multiple satellite or error model configurations.
- Current focus is on airborne LiDAR, due to its nation wide availability. Mobile laser scanning might prove to be a valuable asset. Such scans provide better coverage on the sides of buildings, in recesses (e.g. roofed balconies) invisible to an airborne survey with a low incidence angle, and at the intersections between facades and streets.
- Given the classification provided with the LiDAR point cloud and the free availability of digital topographic data in the Netherlands, it is possible to link deformation behaviour to outlines of buildings, enabling the detection of differential deformation and the conversion from coordinate to addresses (geocoding). Furthermore it would be possible to detect differential deformation on buildings. This would open opportunities for automated early warning systems.
- Augmented reality (and virtual reality) could assist in the analysis of the radar signal. As extensions of the pictures in the 3D viewer, these techniques can help reveal the scattering characteristics of objects.
- Cut-off values could be determined for each matching technique, resorting to a more complex technique only when the previous technique was unable to find a satisfactory solution to the

problem. This would save time and resources.

- Currently no combination is made between AHN2 and AHN3. A combination would require a more advanced error model, including possible deformation. But a combination may give better coverage on low density surfaces, such as facades, as both acquisitions will complement each other. An algorithm could be made to detect major changes (demolition, construction) and only combine the point clouds there where differences are small.

Acknowledgements

I would like to thank SkyGeo¹ for providing the two TerraSAR-X time series used in this thesis.

Furthermore I would like to thank: Roderik Lindenbergh, Ramon Hanssen and Peter van Oosterom for the discussion, reviews and support during the writing of this thesis; Martijn Meijers for the discussion on spatial structures; Ingrid Alkemade for her persistence to fill the gaps in the AHN data and Bibi van der Horst for her support.

¹<https://skygeo.com/>

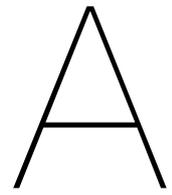
Bibliography

- Airbus Defence and Space GmbH, 2014. TerraSAR-X archive.
- ASPRS, Standards Committee and others, 2013. ASPRS LIDAR Data Exchange Format Standard, version 1.4 r13. Standards document.
- Auer, S. and Hinz, S., 2007. Automatic extraction of salient geometric entities from LiDAR point clouds. In: *Geoscience and Remote Sensing Symposium (IGARSS)*, IEEE, pp. 2507–2510.
- Baek, N., Shin, W.-s. and Kim, K. J., 2017. Geometric primitive extraction from LiDAR-scanned point clouds. *Cluster Computing* 20(1), pp. 741–748.
- Baltsavias, E. P., 1999. A comparison between photogrammetry and laser scanning. *ISPRS Journal of photogrammetry and Remote Sensing* 54(2-3), pp. 83–94.
- Bekendam, R. and Pottgens, J., 1995. Ground movements over the coal mines of southern Limburg, the Netherlands, and their relation to rising mine waters. *IAHS Publications-Series of Proceedings and Reports-Intern Assoc Hydrological Sciences* 234, pp. 3–12.
- Bell, A., Chambers, B., Butler, H., Gerlek, M. et al., 2018. PDAL - point data abstraction library. Hobu inc. <https://www.pdal.io> (March 2018).
- Berger, M., Levine, J. A., Nonato, L. G., Taubin, G. and Silva, C. T., 2013. A benchmark for surface reconstruction. *ACM Transactions on Graphics (TOG)* 32(2), pp. 20.
- Bhattacharya, P., Liu, H., Rosenfeld, A. and Thompson, S., 2000. Hough-transform detection of lines in 3D space. *Pattern Recognition Letters* 21(9), pp. 843–849.
- Boehm, J., 2014. File-centric organization of large LiDAR point clouds in a big data context. In: *Workshop on processing large geospatial data Cardiff, UK*.
- Boersma, C., 2015. The harder we pump, the faster we fall. *Delta Life*, pp. 10-13. Deltares periodical.
- Boulch, A. and Marlet, R., 2016. Deep learning for robust normal estimation in unstructured point clouds. In: *Computer Graphics Forum*, Vol. 35number 5, Wiley Online Library, pp. 281–290.
- Briese, C., 2004. Three-dimensional modelling of breaklines from airborne laser scanner data. ISPRS (details unknown).
- Cabello, R. et al., 2018. Three.js software library. <https://threejs.org/>.
- CBS, 2017. Bestand bodemgebruik. <https://tinyurl.com/yc9mxk66> 2012 (March 2017). Dataset on land use.
- Chang, L. and Hanssen, R. F., 2014. Detection of cavity migration and sinkhole risk using radar interferometric time series. *Remote Sensing of Environment* 147, pp. 56–64.
- Cuenca, M. C., Hanssen, R., Hooper, A. and Arikan, M., 2011. Surface deformation of the whole Netherlands after PSI analysis. In: *Proceedings Fringe 2011 Workshop, Frascati, Italy*, pp. 19–23.
- Dheenathayalan, P. and Hanssen, R., 2013. Radar target type classification and validation. In: *Geoscience and Remote Sensing Symposium (IGARSS), 2013 IEEE International*, IEEE, pp. 923–926.
- Dheenathayalan, P., Small, D. and Hanssen, R. F., in press. 3D positioning and target association for medium resolution SAR sensors. *IEEE Transactions on Geoscience and Remote Sensing*.

- Dheenathayalan, P., Small, D., Schubert, A. and Hanssen, R. F., 2016. High-precision positioning of radar scatterers. *Journal of Geodesy* 90(5), pp. 403–422.
- Fischler, M. A. and Bolles, R. C., 1980. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Technical report, SRI International.
- Garthwaite, M., Nancarrow, S., Hislop, A., Thankappan, M., Dawson, J. and Lawrie, S., 2015. Design of radar corner reflectors for the Australian geophysical observing system. *Geoscience Australia*.
- GDAL Development Team, 2018. GDAL - Geospatial Data Abstraction Library, Version 2.2.3. Open Source Geospatial Foundation.
- Gernhardt, S., Adam, N., Eineder, M. and Bamler, R., 2010. Potential of very high resolution SAR for persistent scatterer interferometry in urban areas. *Annals of GIS* 16(2), pp. 103–111.
- Girardeau-Montaut, D. et al., 2018. Cloudcompare 2.9.
- Hackel, T., Wegner, J. D. and Schindler, K., 2016. Contour detection in unstructured 3D point clouds. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1610–1618.
- Hanssen, R. F., 2001. *Radar interferometry: data interpretation and error analysis*. Vol. 2, Springer Science & Business Media.
- Hughes, L. H., Schmitt, M., Mou, L., Wang, Y. and Zhu, X. X., 2018. Identifying corresponding patches in SAR and optical images with a pseudo-siamese CNN. *IEEE Geoscience and Remote Sensing Letters* 15(5), pp. 784–788.
- Isenburg, M., 2012. LASindex: simple spatial indexing of LiDAR data. In: *European LiDAR Mapping Forum, Salzburg*.
- Isenburg, M., 2013. LASzip: lossless compression of LiDAR data. *Photogrammetric Engineering and Remote Sensing* 79(2), pp. 209–217.
- Isenburg, M., 2018. LASTools. rapidlasso GmbH <https://rapidlasso.com/LAStools/> (March 2018).
- Kadaster and Geonovum, 2018. Publieke Dienstverlening Op de Kaart (PDOK). <https://www.pdok.nl/> (March 2018).
- Kappenburg, B., 2018. Street orientation of dutch cities. <https://tinyurl.com/ycn8dsg7> . Internet forum post.
- Kessy, A., Lewin, A. and Strimmer, K., 2018. Optimal whitening and decorrelation. *The American Statistician* pp. 1–6.
- Ketelaar, G., van Leijen, F., Marinkovic, P. and Hanssen, R., 2006. On the use of point target characteristics in the estimation of low subsidence rates due to gas extraction in Groningen, the Netherlands. In: *Fringe 2005 Workshop*, Vol. 610.
- Ketelaar, V. and Hanssen, R., 2003. Separation of different deformation regimes using PS-InSAR data. In: *Proceedings of FRINGE*, pp. 1–5.
- Kovács, V. and Tevesz, G., 2013. Corner detection and classification of simple objects in low-depth resolution range images. *Periodica Polytechnica. Electrical Engineering and Computer Science* 57(1), pp. 9.
- Leusink, J., 2018. The future of the dutch national height model (AHN). In: *Point Cloud Processing Symposium, Delft*.
- Maas, H.-G. and Vosselman, G., 1999. Two algorithms for extracting building models from raw laser altimetry data. *ISPRS Journal of photogrammetry and remote sensing* 54(2-3), pp. 153–163.

- Martinez-Rubi, O., Verhoeven, S., Van Meersbergen, M., Schütz, M., Van Oosterom, P., Gonçalves, R. and Tijssen, T., 2015. Taming the beast: Free and open-source massive point cloud web visualization. In: *Capturing Reality Forum 2015, 23-25 November 2015, Salzburg, Austria*, The Survey Association.
- Mou, L., Schmitt, M., Wang, Y. and Zhu, X. X., 2017. A CNN for the identification of corresponding patches in SAR and optical imagery of urban scenes. In: *Urban Remote Sensing Event (JURSE), 2017 Joint*, IEEE, pp. 1–4.
- Nielsen, E., Herb, J. et al., 2017. `pykdtree`: fast kd-tree implementation in Python. <https://github.com/storpipfugl/pykdtree> (March 2018).
- Nomen nescio, 2013. Bestekvoorwaarden landsdekkende dataset AHN 2014-2019. Technical report.
- Novellino, A., Cigna, F., Brahmi, M., Sowter, A., Bateson, L. and Marsh, S., 2017. Assessing the feasibility of a national InSAR ground deformation map of Great Britain with Sentinel-1. *Geosciences* 7(2), pp. 19.
- Perissin, D. and Ferretti, A., 2007. Urban-target recognition by means of repeated spaceborne SAR images. *IEEE Transactions on Geoscience and Remote Sensing* 45(12), pp. 4043–4058.
- Psomadaki, S., 2016. Using a space filling curve for the management of dynamic point cloud data in a relational dbms. Master's thesis.
- Python Development Team, 2017. Python programming language. Python Software Foundation <https://www.python.org> (September 2017).
- Rees, W., 2001. *Physical Principles of Remote Sensing*. Cambridge University Press.
- Richards, M. A., Scheer, J. A., Holm, W. A. et al., 2010. *Principles of modern radar*. Vol. 1. Basic principles, SciTech Publishing.
- Rijkswaterstaat Centrale Informatievoorziening, 2013. Inwinning en controle AHN 2014-2015. Tender.
- Rohr, K., 1992. Recognizing corners by fitting parametric models. *International journal of computer vision* 9(3), pp. 213–230.
- Sakurai, Y., Yoshikawa, M., Kataoka, R. and Uemura, S., 2001. Similarity search for adaptive ellipsoid queries using spatial transformation. In: *VLDB*, pp. 231–240.
- Samiei Esfahany, S., 2017. Exploitation of distributed scatterers in synthetic aperture radar interferometry. PhD thesis, Delft University of Technology.
- Schmitt, M., Tupin, F. and Zhu, X. X., 2017. Fusion of SAR and optical remote sensing data - challenges and recent trends. In: *Proceedings of IEEE Geoscience and Remote Sensing Symposium*.
- Schubert, A., Small, D., Miranda, N., Geudtner, D. and Meier, E., 2015. Sentinel-1A product geolocation accuracy: commissioning phase results. *Remote Sensing* 7(7), pp. 9431–9449.
- Schunert, A. and Soergel, U., 2016. Assignment of persistent scatterers to buildings. *IEEE Transactions on Geoscience and Remote Sensing* 54(6), pp. 3116–3127.
- Schütz, M., 2016. Potree: Rendering large point clouds in web browsers. PhD thesis, Technische Universität Wien.
- Schütz, M., 2018. Potree. <http://potree.org/> (January 2018).
- Soudarissanane, S., Van der Sande, C. and Khoshelham, K., 2010. Accuracy assessment of airborne laser scanning strips using planar features. In: *Proceedings of the International Calibration and Orientation Workshop EuroCOW 2010*.
- Stansbury, D., 2013. The statistical whitening transform. <https://tinyurl.com/j2f93lz> (March 2013).

- Subbarao, R. and Meer, P., 2006. Beyond RANSAC: user independent robust regression. In: *Computer Vision and Pattern Recognition Workshop, 2006. CVPRW'06. Conference on*, IEEE, pp. 101–101.
- Teunissen, P. J. and Kleusberg, A., 2012. *GPS for Geodesy*. Springer Science & Business Media.
- Tupin, F., 2010. Fusion of optical and SAR images. In: *Radar Remote Sensing of Urban Areas*, Springer, pp. 133–159.
- van Aalst, J. W., 2018. OpenTopo.nl.
- van de Kerkhof, B., Pankratius, V., Chang, L., van Swol, R. and Hanssen, R., 2017. Information extraction from dynamic PS-InSAR time series using machine learning. In: *AGU Fall Meeting Abstracts*.
- van der Sande, C., Soudarissanane, S. and Khoshelham, K., 2010. Assessment of relative accuracy of AHN2 laser scanning data using planar features. *Sensors* 10(9), pp. 8198–8214.
- van der Zon, N., 2013. Kwaliteitsdocument AHN2.
- van Leijen, F. J., 2014. Persistent scatterer interferometry based on geodetic estimation theory. PhD thesis, Delft University of Technology.
- van Meijeren, H., 2017. Assessing the differences between Dutch elevation datasets AHN2 and AHN3. MSc thesis.
- van Natijne, A., Lindenbergh, R. and Hanssen, R., 2017. Web based visualisation of 3D radar and LiDAR data.
- van Natijne, A., Lindenbergh, R. and Hanssen, R., 2018. Massive linking of PS-InSAR deformations to a national airborne laser point cloud. *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*.
- van Oosterom, P., Martinez-Rubi, O., Ivanova, M., Horhammer, M., Geringer, D., Ravada, S., Tijssen, T., Kodde, M. and Gonçalves, R., 2015. Massive point cloud data management: Design, implementation and execution of a point cloud benchmark. *Computers & Graphics* 49, pp. 92–125.
- Vinson, S. and Cohen, L. D., 2002. Multiple rectangle model for buildings segmentation and 3D scene reconstruction. In: *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, Vol. 3, IEEE, pp. 623–626.
- Vinson, S., Cohen, L. D. and Perlant, F., 2001. Extraction of rectangular buildings in aerial images. In: *Proceedings Of The Scandinavian Conference On Image Analysis*, pp. 431–438.
- Vosselman, G. and Maas, H.-G., 2010. *Airborne and Terrestrial Laser Scanning*. Whittles Publishing.
- Vosselman, G., Dijkman, S. et al., 2001. 3D building model reconstruction from point clouds and ground plans. *International archives of photogrammetry remote sensing and spatial information sciences* 34(3/W4), pp. 37–44.
- Wang, Y., Zhu, X. X., Zeisl, B. and Pollefeys, M., 2017. Fusing meter-resolution 4-D InSAR point clouds and optical images for semantic urban infrastructure monitoring. *IEEE Transactions on Geoscience and Remote Sensing* 55(1), pp. 14–26.
- Weber, C., Hahmann, S. and Hagen, H., 2010. Sharp feature detection in point clouds. In: *Shape Modeling International Conference (SMI), 2010*, IEEE, pp. 175–186.
- Werninghaus, R. and Buckreuss, S., 2010. The TerraSAR-X mission and system design. *IEEE Transactions on Geoscience and Remote Sensing* 48(2), pp. 606–614.
- Yang, M., Dheenathayalan, P., Biljecki, F. and Hanssen, R. F., 2017. On the predictability of PS occurrence and location based on 3D ray-tracing models. In: *Fringe 2017 Workshop*.
- Yari, D., Mokhtarzade, M., Ebadi, H. and Ahmadi, S., 2014. Automatic reconstruction of regular buildings using a shape-based balloon snake model. *The Photogrammetric Record* 29(146), pp. 187–205.
- Yarovoy, A., Krasnov, O., Uysal, F. and Unal, C., 2017. Microwaves, radar and remote sensing (ET4169). Lecture slides.



Additional figures

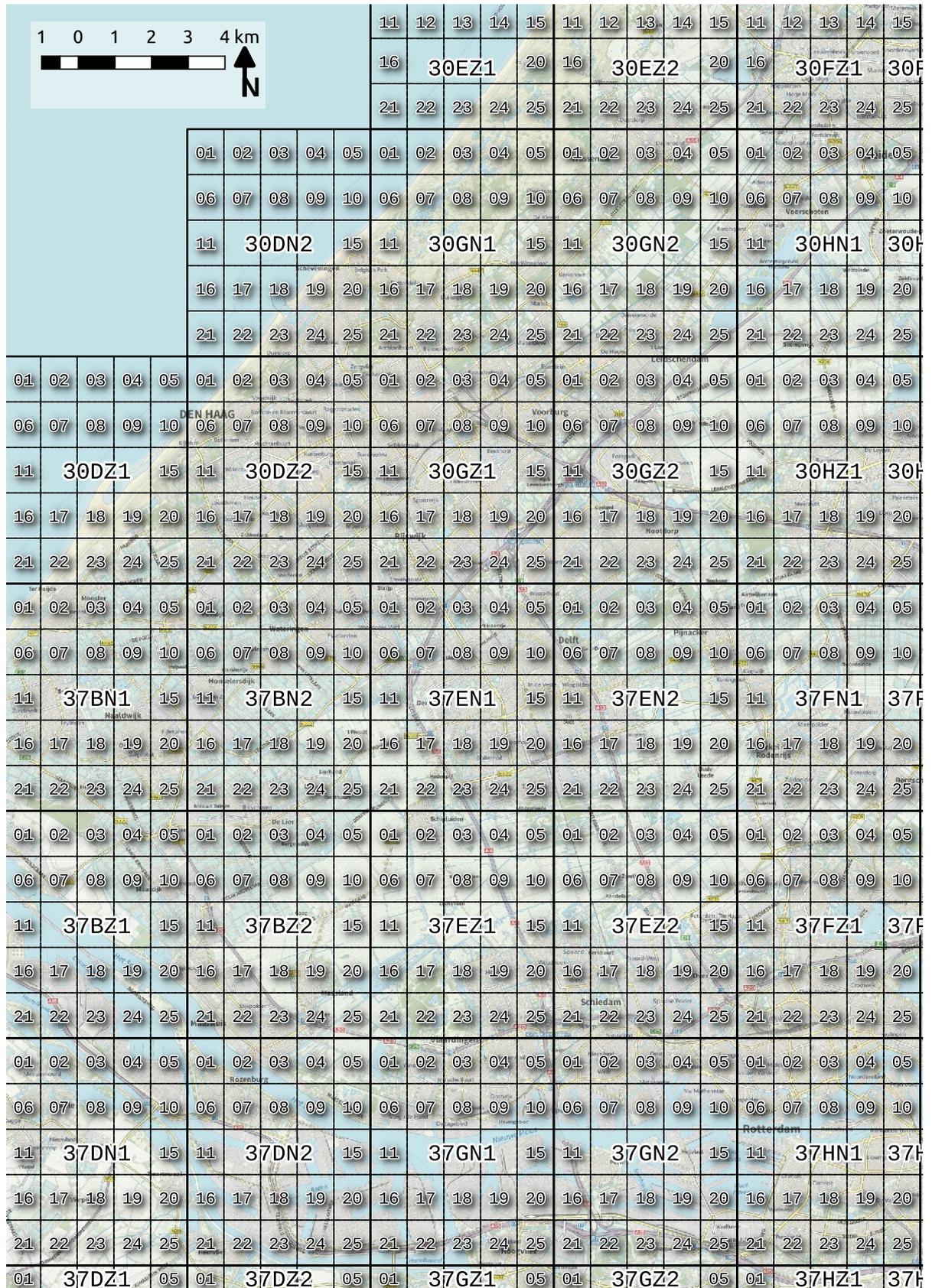


Figure A.1: Tile index over Delft and surroundings, including full tile names and numbers of sub-tiles. (Background: Van Aalst [2018].)

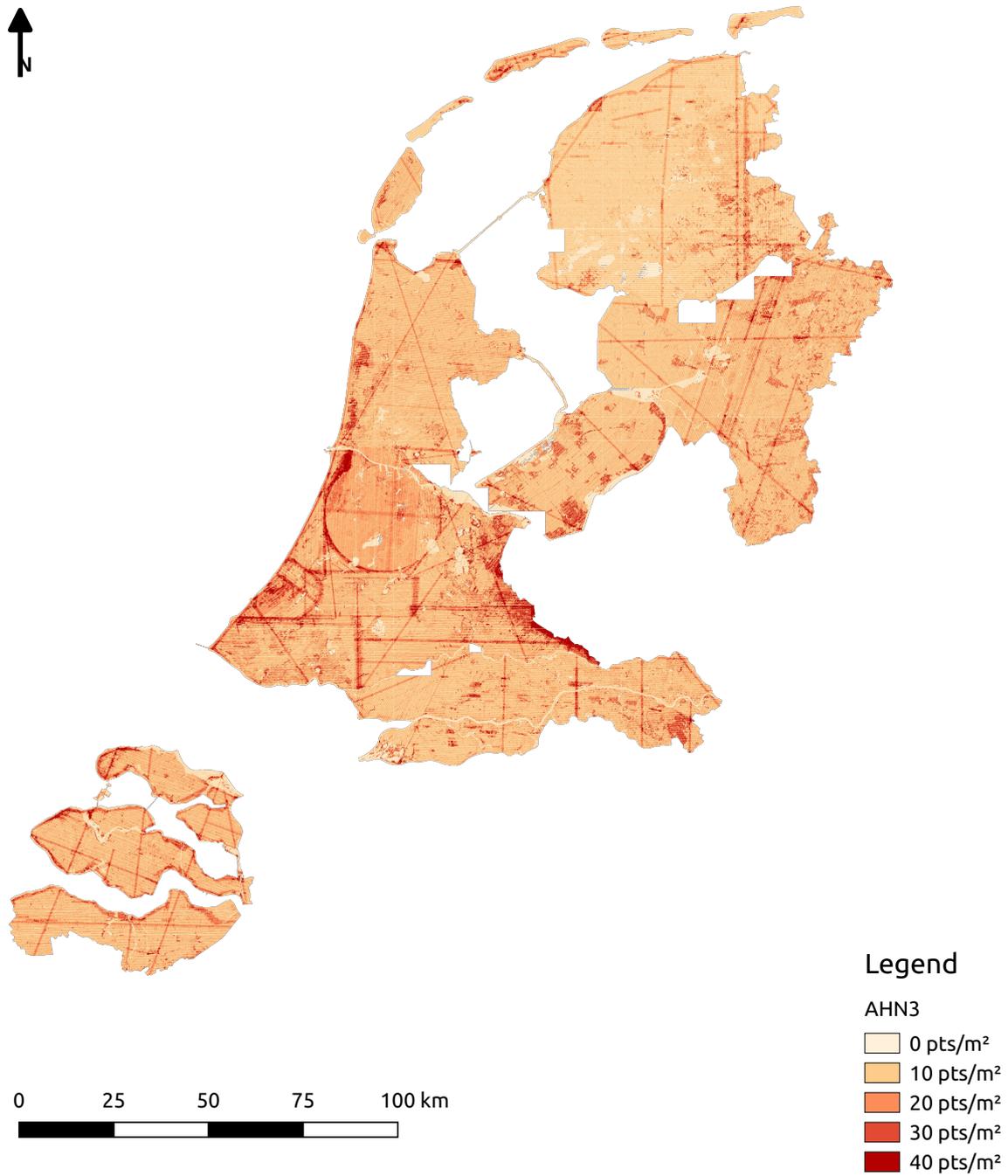


Figure A.2: Point density of AHN3, sampled over a 35 meter radius (for 25×25 meter pixels).

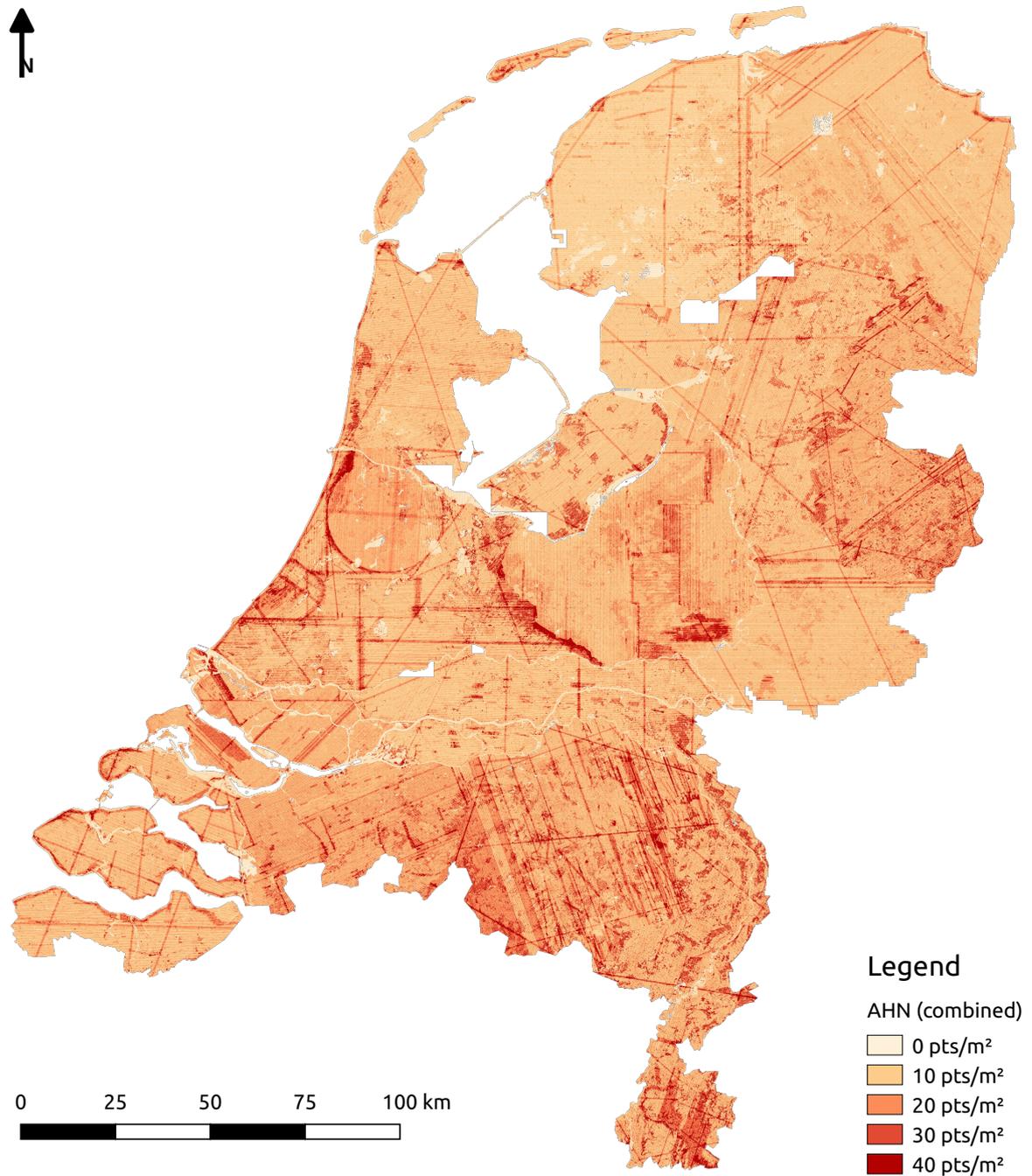


Figure A.3: Point density of a combined AHN2 and AHN3 product, sampled over a 35 meter radius (for 25×25 meter pixels). The gaps in the data correspond to the missing data in AHN3, these could be filled with AHN2.

Oriëntatie van straten in Nederlandse (hoofd)steden

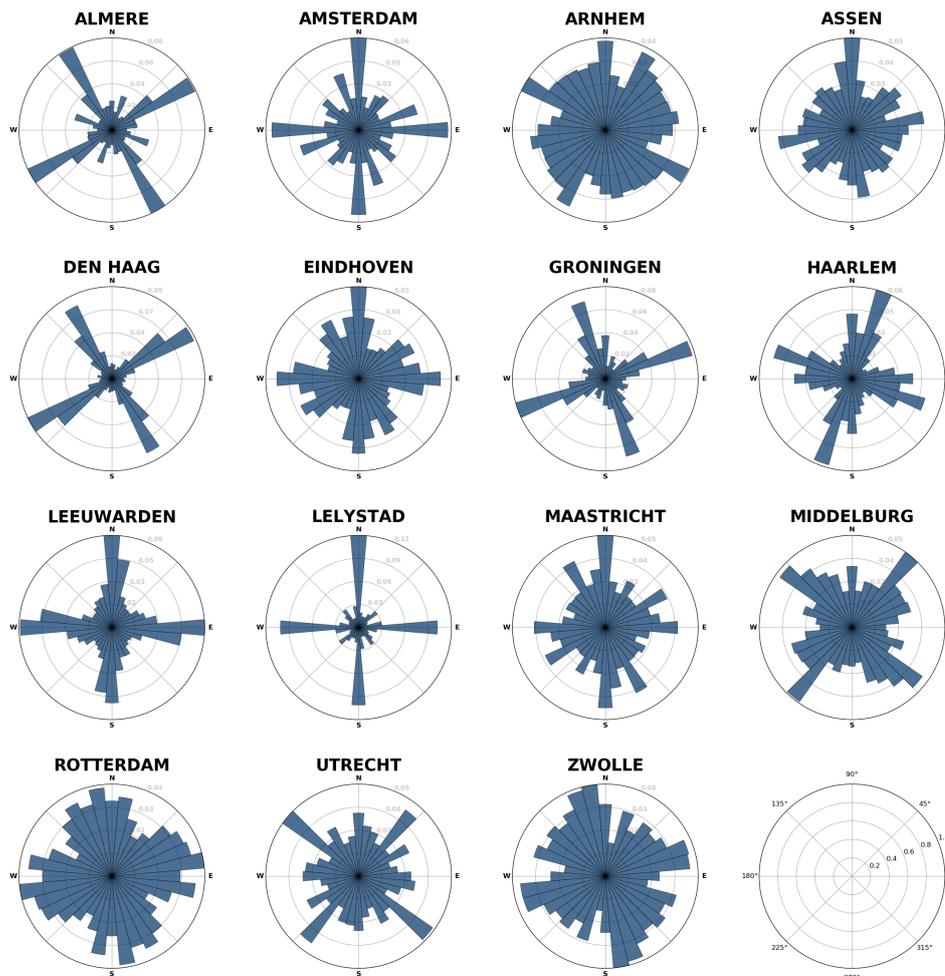


Figure A.4: Orientations of Dutch streets, as analyzed from OpenStreetMap by Kappenburg [2018]. With the assumption that facades are parallel to the street and AHN acquisitions are perpendicular to the flight direction, the facade point density can be predicted. The orientation towards the radar satellite may give information on the radar signal to be expected. (Caption: "Orientations of streets in Dutch (provincial) capitals.")



Data preparation commands

Actueel Hoogtebestand Nederland

Each step discussed in section 3.3 is represented here by a section of the same name that lists some commands or hints on the use of `LASTools` [Isenburg, 2018], `PDAL` [Bell et al., 2018] and `GDAL` [GDAL Development Team, 2018] for data preparation.

`PDAL` works with *pipelines*, series of commands that are subsequently executed, including reading, filtering and writing of various file types, like raster representations of the point cloud. `LASTools` is a set of more traditional command line tools, accepting arguments to process from and to (mostly) `LAS`- or `LAZ`-files. `LASTools` is renowned for being highly efficient, Van Oosterom et al. [2015].

Merging AHN2

This process is two fold, with a single intermediate file. First all non-ground points are set from *never classified* to *unclassified*, using the `las2las` [Isenburg, 2018] (examples for tile `01CZ1`):

```
las2las -i u01cz1.laz -set_classification 1 -olas -o u01cz1_1.laz
```

The result should be stored in temporary storage, preferably on a fast SSD or in memory. The second step is merging the file with the ground-points, for which the classification will be set from *never classified* to *ground*.

```
las2las -i u01cz1_1.laz -i g01cz1.laz -merged  
  -change_classification_from_to 0 2 -olaz -o 01cz1.laz
```

Coloring

The following pipeline can be used to instruct `PDAL` to combine `input.laz` and aerial photograph `img.tiff` into `out.laz`.

```
{  
  "pipeline": [  
    {  
      "type": "readers.las",  
      "filename": "input.laz"  
    },  
    {  
      "type": "filters.colorization",  
      "raster": "img.tiff"  
    },  
  ]  
}
```

```

        "type": "writers.las",
        "forward": "scale,offset",
        "pdal_metadata": true,
        "a_srs": "EPSG:28992",
        "dataformat_id": 3,
        "filename": "out.laz",
        "compression": true,
    }
]
}

```

The LAS-format requires adjustment to add the RGB values to each record in the file. For AHN2 `dataformat_id` has to be set to 2, as no time attribute is available. A `dataformat_id` of 3 indicates that for each point color and time should be stored, 2 indicates only color, [ASPRS, 2013].

Creation of a spatial index

A spatial index of the LAS or LAZ file can be generated with the following command:

```
lasindex -i C_16EZ2.LAZ
```

Although only the paid option of `LASindex` offer multithreading support, multiple instances of the open source version can be run simultaneously. For example, split the processing of a full directory into four instances using `xargs`:

```
find . -iname '*.LAZ' -print0 | xargs -P 4 -0 lasindex -i
```

Concession crop

A simple PDAL pipeline can be used to crop the tiles to the concession boundaries found:

```

{
  "pipeline": [
    "in.las",
    {
      "type": "filters.crop",
      "polygon": "WKT"
    },
    "out.las"
  ]
}

```

For every iteration the input (`in.las`), output (`out.las`) and bounding polygon (defined in the Well-Known-Text, `WKT`, format) are replaced.

Potree conversion

The following command is used to start the conversion process, a detailed description of the options is available in Van Natijne et al. [2017].

```

PotreeConverter \
~/AHN3_R/AHN2014_RegioZuidHolland_inclSchiphol/ \
-o ~/AHN3_PoTree \
--material RGB \
-p RegioZuidHolland_inclSchiphol \
--show-skybox \
--projection "+proj=sterea +lat_0=52.15616055555555
  ↳ +lon_0=5.387638888888889 +k=0.9999079 +x_0=155000 +y_0=463000
  ↳ +ellps=bessel +units=m +no_defs" \

```

```
--edl-enabled \
-a RGB CLASSIFICATION INTENSITY \
-d 1024 \
--format LAZ
```

Tiling

Depending on the position of the sub-tile it might be necessary to include neighboring tiles in the tiling process to provide the required 25 meter buffer (see figure A.1). Together with the new bounding box a `las2las` command is constructed to crop the input tiles and merge the resulting points into a single new file. Although the process itself is single threaded, multiple operations can be run simultaneously.

To create sub-tile `37EN2_16` from tiles `37EN1` and `37EN2` at the TU Delft campus (see figure A.1) the following command is used:

```
las2las -i ~/AHN3/C_37EN1.LAZ -i ~/AHN3/C_37EN2.LAZ -merged -o
  ↳ ~/AHN3_T/T_37EN2_16.LAZ -inside 84975 444975 86025 446275
```

This process benefits greatly from the previously generated spatial index.

Normals

First the normals are estimated based on the eight closest points, the output is then mapped to a small python program that scales the values to the range available in eight bits (-127 to +127). This last step is only necessary to benefit from the integer compression capabilities of the LAZ-format. This is accomplished with the following PDAL pipeline:

```
{
  "pipeline": [
    {
      "filename": "/home/adriaan/AHN3_T/T_37EN2_10.LAZ",
      "type": "readers.las"
    },
    {
      "type": "filters.normal"
    },
    {
      "type": "filters.python",
      "script": "normal.py",
      "function": "scale",
      "module": "anything"
    },
    {
      "filename": "/tmp/normal.laz",
      "type": "writers.las",
      "pdal_metadata": true,
      "compression": true,
      "forward": "scale,offset",
      "dataformat_id": 0,
      "a_srs": "EPSG:28992",
      "extra_dims": "NormalX=int8_t,
NormalY=int8_t,NormalZ=int8_t,Curvature=int8_t"
    }
  ]
}
```

This pipeline is used in conjunction with the following small python function (`normal.py`), which sole purpose is to scale the values ([-1, 1]) to the maximum range of `int8` ([-127, 127]):

```
import numpy as _np;

def scale(ins, outs):
    outs['NormalX'] = ins['NormalX'] *127;
    outs['NormalY'] = ins['NormalY'] *127;
    outs['NormalZ'] = ins['NormalZ'] *127;

    outs['Curvature'] = ins['Curvature'] *127;

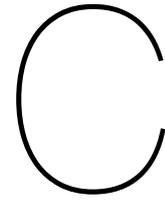
    return True;
```

Combining AHN2 and AHN3

For sub-tiles that are on the boundary of AHN2/AHN3 availability, the AHN2 sub-tile is cropped to the area where AHN3 is not available, based on concession boundaries. The resulting sub-tile is subsequently merged with the AHN3 tiles using the following PDAL pipeline, reading both files, cropping them based on the AHN2/AHN3 boundary and merging them into a single file:

```
{ "pipeline": [
  { "type": "readers.las",
    "filename": "AHN2_f",
    "tag": "AHN2"
  },
  { "type": "readers.las",
    "filename": "AHN3_f",
    "tag": "AHN3"
  },
  { "type": "filters.crop",
    "polygon": "WKT",
    "tag": "AHN2_C",
    "inputs": ["AHN2"]
  },
  { "type": "filters.crop",
    "polygon": "WKT",
    "tag": "AHN3_C",
    "inputs": ["AHN3"]
  },
  { "type": "writers.las",
    "filename": "AHN_f",
    "compression": true,
    "inputs": ["AHN2_C", "AHN3_C"]
  }
]}
```

The `WKT` references are replaced with the boundaries of both AHN2 and AHN3 availability. Both layers are cropped to prevent unexpected overlap if points exist outside of the concession boundary.



ISPRS Conference Paper

The following paper was presented during the ISPRS TC II Mid-term Symposium “Towards Photogrammetry 2020”, 4–7 June 2018, in Riva del Garda, Italy. And is included as such in the The International Archives of the Photogrammetry, Volume XLII-2, 2018.

MASSIVE LINKING OF PS-IN SAR DEFORMATIONS TO A NATIONAL AIRBORNE LASER POINT CLOUD

A. L. van Natijne*, R.C. Lindenbergh, R.F. Hanssen

Delft University of Technology, Department of Geoscience and Remote Sensing,
Stevinweg 1, 2628CN Delft, The Netherlands - A.L.vanNatijne@student.tudelft.nl

Commission II, WG II/10

KEY WORDS: Airborne Laser Scanning, PS-InSAR, geolocation, deformation monitoring, local surface reconstruction, data fusion

ABSTRACT:

Build on soft soil, close to sea level the Netherlands is at high risk for the effects of subsidence and deformation. Interferometric Synthetic Aperture Radar (InSAR) is successfully used to monitor the deformation trends at millimetre level. Unfortunately the InSAR deformation trends suffer from poor geolocation estimates, limiting the ability to link deformation behaviour to objects, such as buildings, streets or bridges. A nationwide, high resolution, airborne LiDAR point cloud is available in the Netherlands. Although the position accuracy of this LiDAR point cloud is too low for deformation estimates, linking the InSAR location to the geometries outlined by the LiDAR point can improve the geolocation estimates of the InSAR trends. To our knowledge no such integration is available as of yet. In this article we outline methods to link deformation estimates to the LiDAR point cloud and give an outlook of possible improvements. As a test we link 3.1 million TerraSAR-X InSAR Persistent Scatterers to 3 billion LiDAR points, covering the city of Delft and surroundings.

1. INTRODUCTION

1.1 Subsidence

Subsidence in the Netherlands, and deformation in general, is threatening building integrity, damaging infrastructure and lowering the land with respect to sea-level. Deformation occurs at all scales, from single pillar failure at the 't Loon shopping mall in 2011 (Chang and Hanssen, 2014) to complete regions suffering from effects such as subsidence relative to the water table (Boersma, 2015).

Deformation processes include: peat compaction in the west and north of the country (Boersma, 2015); induced subsidence (and seismic activity) due to gas extraction in the Groningen area (Ketelaar et al., 2006); land uplift and cavity formation due to the flooding of old mine works in Limburg (Bekendam and Pottgens, 1995).

1.2 PS-InSAR deformation monitoring

Interferometric Synthetic Aperture Radar (InSAR) is used to monitor deformation from satellites. Millimetre per year accuracy can be achieved in deformation trend estimation. Unfortunately the source of the deformation signal is, in general, less accurately known: geolocation estimates of PS-InSAR are known with metres precision at best, depending on the sensor (Dheenathayalan et al., 2016).

Although this will allow for deformation estimates up to street level (Ketelaar et al., 2006), the deformation signal can not be attributed to a single geometric feature. Persistent scatterer InSAR (PS-InSAR) measurements are commonly dominated by a single scatterer. The location of this scatterer is of key importance in the understanding and interpretation of the deformation behaviour: a subsiding garden house or street will require different precautions than a subsiding bridge pillar.

*Corresponding author

1.3 Combination with LiDAR

To find and improve the estimated location of the dominant scatterer it is beneficial to combine radar measurements with a (high resolution) point cloud. This will allow for linking scattering behaviour to a geometric feature in the scene.

An example of a traditional (2D) InSAR deformation map can be seen in figure 2. A 3D visualisation aids this interpretation of the PS-InSAR signal, over the classical 2D mapped interpretation. Geometric 3D linking of PS-InSAR geolocation estimates to 3D LiDAR point cloud data will give a quantifiable improvement of the geolocation.

Geometric (3D) linking of the datasets will provide:

- Assessment of differential deformation, as a deformation signal can be *attached* to a building geometry.
- Linking of the deformation signal to specific parts of the infrastructure, for maintenance planning and early warning.
- Detection and mitigation of (regional) errors and trends in the radar processing.

This article explains how to create this missing link by truly integrating both data sources. Given the (free) availability of a nationwide airborne LiDAR dataset (Actueel Hoogtebestand Nederland), and the available TerraSAR-X InSAR data, the Netherlands form a perfect test bed for this integration of datasets. Furthermore existing online point cloud viewers, such as Potree (Schütz, 2018), can be extended to visualise this link between the laser point cloud and radar data.

Currently no such combination of datasets is known to us. The combination of optical images and SAR is more common and

aimed at the texturing, classification or 3D reconstruction of SAR point clouds (Tupin, 2010, Schmitt et al., 2017). Although successful, (Wang et al., 2017) suffered from poor InSAR geolocation accuracy.

2. DATA

2.1 TerraSAR-X

The German (DLR) TerraSAR-X Synthetic Aperture Radar (SAR) mission, was launched in 2007 and delivers high resolution radar imagery ever since. As a Public-Private Partnership the mission combines scientific and commercial interests of X-band, land oriented monitoring applications. With an 11-day repeat cycle and resolutions up to 3×3 m the mission can provide deformation data of high spatial and temporal resolution (Werninghaus and Buckreuss, 2010).

For this article radar data from two TerraSAR-X orbits is available, descending and ascending orbits over the same region, covering Delft, surrounding fields, Rijswijk and parts of The Hague (the Netherlands), marked in red on figure 3 and shown in more detail in figure 1.

A total of 72 radar images were acquired between 2016-01-06 and 2018-01-01, 36 for each orbit. SAR Interferometry (InSAR) was applied to extract deformation signals by analysing the time series of phase changes. Pixels that can be tracked consistently over multiple acquisitions are Persistent Scatterers (PS). These *coherent* pixels denote the deformation behaviour of the same scatterer over longer periods of time. (Hanssen, 2001) A linear deformation trend (in time, $\frac{\text{mm}}{\text{yr}}$) is estimated for those points.

Data of the descending orbit contains 1.7 million PS-InSAR points, the ascending data contains 1.4 million points. For all points a geolocation estimation is provided in WGS84 coordinates and a height above NAP (*Normaal Amsterdams Peil*). Both datasets span the same area of 123 km², of which 60 km² is over urban (built) terrain, where the highest density of persistent scatterers is to be expected (Hanssen, 2001, CBS, 2017).

2.1.1 Error model The ratio of the error ellipsoid in range (direction of radar signal), azimuth (direction of flight) and cross-range (perpendicular to signal and flight direction) was estimated to be 1/2/22 for typical consistent scatterers (figure 4, not to scale). With an estimated standard deviation of 0.128 m, the axes of the error ellipsoids are 0.128 m (range, σ_r), 0.256 m (azimuth, σ_a), 2.816 m (cross-range, σ_{cr}) (Dheenathayalan et al., 2016).

The direction of flight was defined to be 192° for the descending and 350° for the ascending orbit, with a elevation angle of 65.9° for both orbits.

2.2 Actueel Hoogtebestand Nederland

Actueel Hoogtebestand Nederland (AHN) is a nationwide LiDAR elevation model. AHN was first recorded in 1996 and is licensed as open data since march 2014. The raw point cloud data is published via *Publieke Dienstverlening Op de Kaart* (PDOK) (Kadaster and Geonovum, 2018).

AHN was acquired from an airborne platform, from which laser pulses were fired at the ground below. Given the known propagation velocity of light in air, the time interval between transmission and receiving the reflected signal (*echo*) is proportional to

the distance from the aircraft to the ground. Multiple returns are possible, for example in vegetated areas, where parts of the pulse reflect on different surfaces in the scene. The position and orientation of the aircraft are recorded simultaneously using GNSS and inertial motion sensors to record the position from where the measurement was acquired and in which direction (Vosselman and Maas, 2010).

New iterations are acquired approximately once every ten years, as can be seen in table 1. Acquisition of both AHN1 and AHN2 is finished: AHN2 supersedes AHN1. AHN3 is yet only partially available, and is used where available (figure 3). A combination is made between AHN2 and AHN3 to create a single dataset with the highest possible point density. The higher point density allows for better detection of small objects and improved reconstruction of facades that are badly aligned with respect to the viewing angle of the laser scanner.

A summary of the data volume involved is given in table 2. The file size increase of AHN3 is due to the addition of extra attributes, such as classification, intensity and acquisition time.

	Recording	Density ($\frac{\text{pt}}{\text{m}^2}$)
AHN1	1996 - 2003	0.06 - 1
AHN2	2008 - 2012	16.8
AHN3	2014 - 2018	18.2

Table 1. Acquisition years and average data density of the three available iterations of AHN. Point densities for AHN2 and AHN3 were determined over the study area (9 tiles).

	Area (km ²)	Points	Size
AHN2	35 997	640 billion	988 GiB
AHN3	16 249	252 billion	1.1 TiB
Combined			1.7 TiB

Table 2. Coverage, point count and dataset size of AHN2, AHN3 and a combination of both.

AHN is defined in *Rijksdriehoekskoördinaten* with height relative to NAP, 'RDNAP', the Dutch national coordinate system (EPSG: 7415). This Cartesian coordinate system is used as the basis for this project.

To ease navigation in the web application (section 3.1) the point cloud is coloured based on the publicly available aerial photograph of 2016 (Kadaster and Geonovum, 2018). This photograph shows small differences to the point cloud, as it was not recorded simultaneously.

2.2.1 Error model For AHN (iteration 2 and 3) the accuracy is defined as maximum 5 cm systematic (1σ) and a 5 cm stochastic error (1σ) in the vertical direction. Requirements for horizontal accuracy are 50 cm (1σ , both x and y) for objects larger than 2×2 m. In reality this is often outperformed (van Meijeren, 2017).

The large standard deviation and the infrequent acquisition make AHN itself unsuitable for deformation monitoring at the millimetre level that is obtained by InSAR monitoring (van Meijeren, 2017).

2.3 Data preparation

For the massive visualisation a combined dataset of AHN3 and AHN2 is created, AHN3 is used whenever available (figure 3).

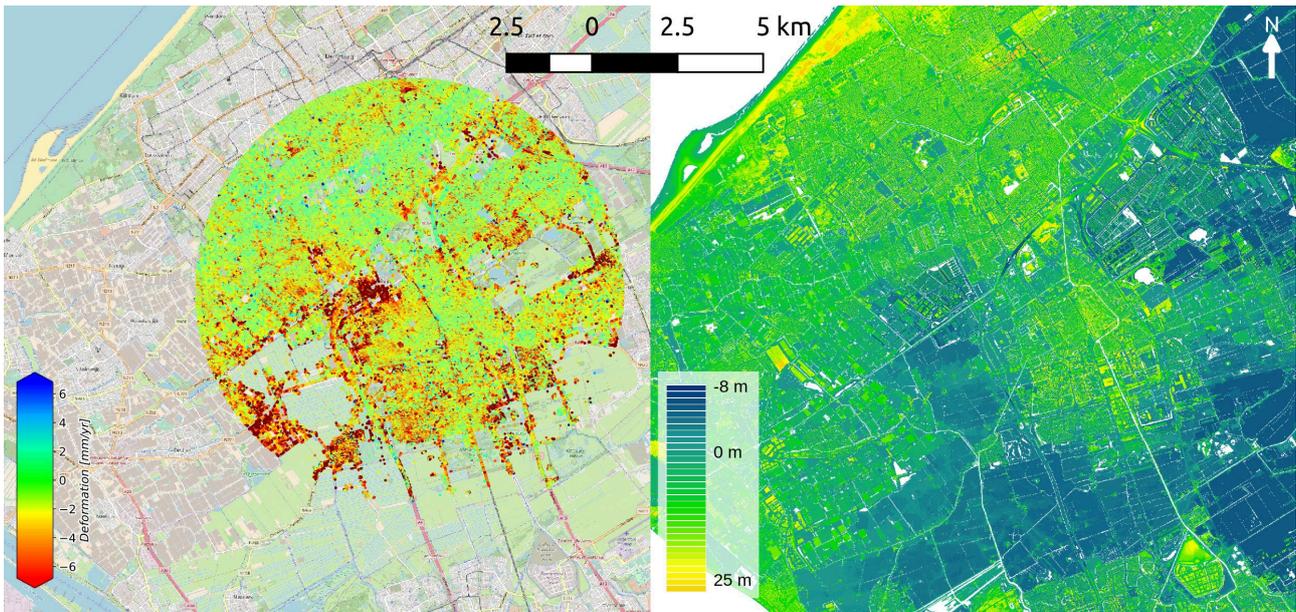


Figure 1. Left: PS-InSAR data from both ascending and descending orbits. The city of Delft is hidden below the points. The Hague can be seen in the north and the northern part of Rotterdam in the south. (Background: OpenStreetMap) Right: digital surface model extracted from AHN3 of the same region.

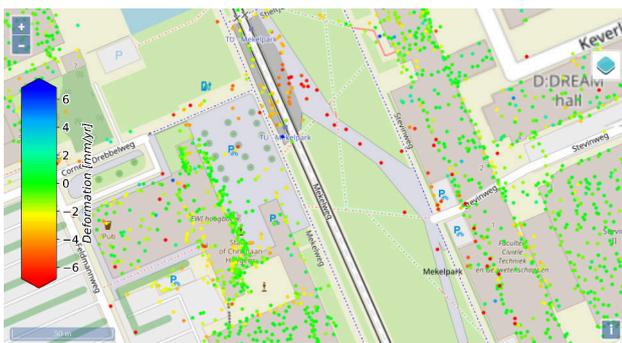


Figure 2. Traditional PS-InSAR deformation map, showing the TU Delft campus. Misalignment can be seen at the building facades. (Background: OpenStreetMap.)

AHN data is delivered in LAZ-tiles of 5×6.25 km, based on the standard national tiling scheme. Both datasets are tiled in tiles of approximately 1 km^2 (1×1.25 km, $\frac{1}{25}$ of the original tiles). For each LiDAR tile a buffer of 25 metres on all sides is included, this is to allow radar measurements on the border of two tiles to match. Given the radar error model, such a buffer is large enough to accommodate points on the border between tiles.

This process is done using `pda1` (colouring and clipping) and `las2las` of LASTools (tiling) (Bell et al., 2018, Isenburg, 2018). It created 30137 LAZ-files (tiles), with an average filesize of 58 MiB and 21 million points (average point density of 15 points per square metre). These tiles are small enough to be processed in memory, and large enough for regular file storage. Queries on the LAZ-files are elementary (axes aligned bounding boxes) and do not require a point cloud database (van Oosterom et al., 2015).

Due to the 25 metre overlap between tiles, the method may be run for each tile independently, *horizontal scaling* of the algorithm is possible. That is, each tile can be processed independently, on

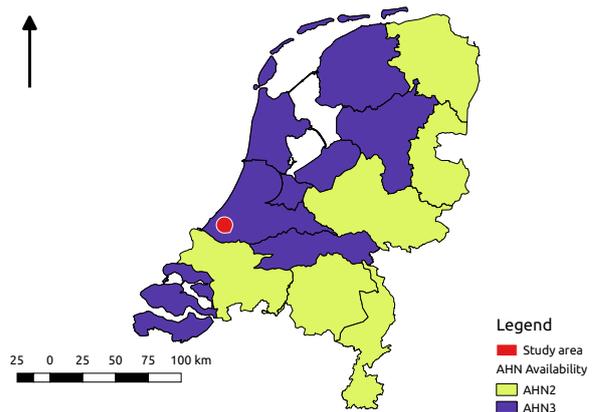


Figure 3. Map of the Netherlands, showing the availability of AHN2 and AHN3 respectively. Shown in red is the extent of the TerraSAR-X data available to this study.

a separate CPU or even separate node. This enables us to combine the PS-InSAR points with *massive* numbers of LiDAR measurements in a distributed manner, reducing the execution time required.

Due to the small size, the radar dataset is not tiled. Small datasets (such as the TerraSAR-X dataset discussed here) are tiled in memory. Larger datasets are converted to HDF5 first, which allows efficient querying during processing.

3. METHOD

To aid the interpretation of the deformation signal contained in the PS-InSAR data we want to visualise the available data and link the data to the geometry known from the LiDAR point cloud. The following five steps to achieve this will be discussed here:

1. Common visualisation of InSAR and LiDAR data
Interpretation is left to the operator, just as with traditional (online) maps (such as figure 2). Unlike traditional 2D maps the geolocation estimation is shown with the error ellipsoid in 3D. This enables improved interpretation of the radar signal and scene geometry.
2. First Nearest Neighbour linking
The Nearest Neighbour in the LiDAR point (single point), with respect to the radar geolocation estimate, is expected to govern the scattering behaviour. This method is computationally efficient, but may overestimate the distance on low density surfaces, as illustrated in figure 4.
3. Linking to a single surface
After Nearest Neighbour search, either up to a predefined number of points or all points up to a maximum radius, a single plane is fitted on the points found. This plane approximation of e.g. a facade makes the matching algorithm more robust in areas of low point density.
4. Linking to multiple surfaces
For complex geometries the previous approach can be extended. Multiple locally linear ('flat') surfaces may exist in the neighbourhood of the scatterer.
5. Linking to dihedral or trihedral geometries
Dihedral and trihedral surface configurations are known to act as better radar reflectors. These geometric configurations might be extracted from the scene. This is currently not implemented.

The difference between the methods is sketched in figure 4. Approximating locally linear surfaces as 2D planes adds robustness in case of point density differences. On rough surfaces, or non flat surfaces this approximation may not hold. Rough surfaces are approximated by an 'average' surface and it might be possible to approximate non-flat surfaces as linearly close to the point of intersection.

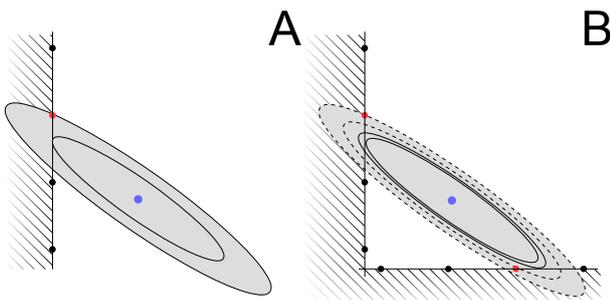


Figure 4. Sketch of the effect of surface reconstruction on matching between a LiDAR point cloud (black/red dots) and a PS-InSAR geolocation estimate (blue dot, including error ellipsoids). The match with Nearest Neighbour in the point cloud, shown in red, is much further away than the actual surface (thin black line). Shown in a *simple* situation A; and a *complex* situation B. Although the horizontal surface is further away, it could be considered a candidate match.

After matching, the resulting matching distances may be used to analyse the geolocation quality, for example detection of biases and trends in the geolocation.

3.1 Common visualisation of InSAR and LiDAR data

Web based visualisation is built around Potree (Schütz, 2016, Schütz, 2018) and Three.js. Potree is a WebGL based renderer for large point clouds in the web browser, built on top of Three.js 3D library. Previously the full (nationwide) AHN2 dataset has been successfully converted to be used in the Potree viewer (Martinez-Rubi et al., 2015). Other visual aids (such as the error ellipsoids and plane estimates) can be implemented using Three.js.

Potree ensures a smooth viewing experience by loading the point cloud from a pre-processed octree structure. Only the points in view at the client are downloaded and never more than a user-defined maximum. Due to the $2\frac{1}{2}$ D nature of the radar dataset this data is distributed using a quadtree tiling scheme, loading only the tiles in view and removing those no longer in view from memory.

3.2 First Nearest Neighbour linking

Nearest Neighbour search should take this covariance into account. Use of the *Whitening Transform* will allow any (conventional) Nearest Neighbour algorithm to be used on this problem (Stansbury, 2013).

The viewing geometry of the radar satellite can be expressed as a rotation matrix relative to the world coordinates (RDNAP). This rotation (\mathbf{R}_{SAR}) can be combined with the quality model of the radar geolocation ($\sigma_r, \sigma_a, \sigma_{cr}$) to a covariance matrix relative to world coordinates:

$$\mathbf{Q}_{SAR} = \mathbf{R}_{SAR} \begin{bmatrix} \sigma_r & & \\ & \sigma_a & \\ & & \sigma_{cr} \end{bmatrix} \mathbf{R}_{SAR}^T \quad (1)$$

Using the Whitening Transform all points (LiDAR and radar) are projected on the eigenvectors of the covariance matrix of the radar point (\mathbf{Q}_{SAR}) and scaled by the eigenvalues. This creates a new coordinate system where the Euclidean metric represents distances in σ rather than metres. All errors are now standard normal distributed, as can be seen in figure 5.

The transformation found works for a single, constant, error model only. As a consequence the transformation has to be calculated and applied for each unique viewing geometry (ie. orbit). This includes construction of a new search structure for each viewing geometry and/or error model. The Multiple Spatial Transformation Technique by (Sakurai et al., 2001), based on pre-processed search structures and approximate transformations may be used to speed up this process if required.

This transformation, based on the eigenvalues (\mathbf{E}) and a diagonal matrix of the eigenvectors (\mathbf{D}) of the covariance matrix of the radar measurements (\mathbf{Q}_{SAR}), can be found using:

$$\mathbf{W} = \mathbf{E}^{-1} \mathbf{D}^{-\frac{1}{2}} \mathbf{E}^T \quad (2)$$

The correctness of this transformation can be checked by transforming the covariance matrix using the Whitening Transform to the identity matrix:

$$\mathbf{W} \mathbf{Q}_{SAR} \mathbf{W}^T = \mathbf{I}_3 \quad (3)$$

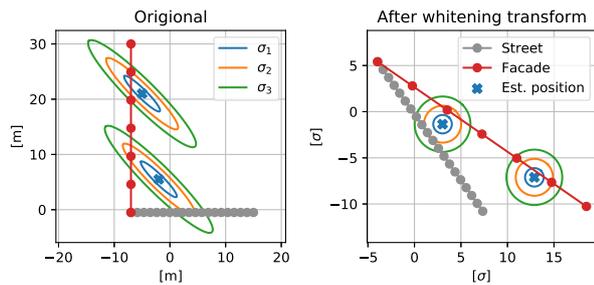


Figure 5. Effect of the Whitening Transform.

In this coordinate system, Nearest Neighbours are nearest in a statistical sense. This search is optimal with respect to the radar, considering AHN the ground truth, without any statistical variability. This assumption is justified due to the (relatively) small error of the LiDAR point cloud.

This search is implemented using `pykdtree`, a kD-tree implementation in Python (Python Development Team, 2017, Nielsen et al., 2017).

3.3 Linking to a single surface

The surface is locally approximated as a single, three dimensional, plane of the equation:

$$a\hat{x} + b\hat{y} + c\hat{z} = d \quad (4)$$

Coefficients (a , b , c) equal a normal vector of the plane and d is a constant.

Two approaches were chosen to approximate the surface:

- Using three LiDAR points, the coefficients are defined by the cross-product of the coordinates of these points.
- Using all Nearest Neighbours found, employing Principal Component Analysis (PCA) to find the coefficients of the plane.

The first method (cross-product) is computationally light but the plane is based on three points only and does not exploit the redundancy in the LiDAR point cloud and does not provide a quality metric for the fit. PCA requires more computational effort but exploit the redundancy in the data and provide a quality metric for the fit.

The error model of both datasets is taken as the starting point of the fusion. To estimate the plane, first the covariance matrix of the coordinates is calculated for the LiDAR points found. This covariance matrix is then transformed using the Whitening Transform based on the covariance matrix of AHN (\mathbf{Q}_{AHN} , formula 2). The eigenvector corresponding to the smallest eigenvalue of the covariance matrix represents the normal vector. The ratio between the eigenvalues is an indicator for the quality of the fit, for flat surfaces the smallest eigenvalue is much smaller than the other two. The constant d of the plane equation (4) is found by solving the equation for the mean point.

To determine the point of intersection, the PS-InSAR Whitening Transform is applied to the PS-InSAR point and the surface normal found. The PS-InSAR point is projected orthogonal on the

surface, the distance found is the distance in σ . After the application of the inverse Whitening Transform the point of intersection in world coordinates is found.

3.4 Linking to multiple surfaces; dihedral and trihedral geometries

Detection of multiple surfaces can generate a *top-3* list of candidate intersections, such as the horizontal surface slightly further away in figure 4B. Detection of surfaces is based on RANSAC (*random sample consensus*) and is only applied if the surface estimated using the single surface estimating technique indicates the area as non-flat.

Furthermore it will allow the detection of more advanced structures, such as dihedral and trihedral configurations of multiple surfaces. To respect the radar scattering behaviour of dihedral and trihedral structures the source of the deformation signal should be placed at the intersection of the surfaces rather than at the surfaces themselves (Richards et al., 2010).

4. RESULTS

4.1 Webviewer

A screenshot is shown in figure 6. A live demonstration can be found at <http://dev.fwrite.org/radar/>.

4.2 Matching

A comparison between the various matching techniques can be seen in the histograms of figure 7. Local reconstruction of the geometry, by surface approximation, leads to lower distances between the original geolocation estimation and the surface found (orange), as sketched in figure 4. The intersection with the surface is on average $\frac{1}{2}\sigma$ closer than the first Nearest Neighbour (blue).

Over the whole dataset biases are in the order of decimetres, with standard deviations of multiple metres, as can be seen in figure 8. When expressed in radar coordinates the uncertainty in matching corresponds to the expected geolocation error. The expected geolocation standard deviation was 0.128/0.256/2.816 metre in range/azimuth/cross-range (section 2.1.1). As can be seen in figure 9, the standard deviations are of the same order of magnitude as the original estimations. In range and azimuth it is overestimated, while the cross-range estimate is of the same order as the original estimate. This is all under the assumption that the first Nearest Neighbour is the origin of the backscattered signal.

The improvement in location can be seen in figure 10. Compared to figure 2 stable points are now attributed to the facade while subsiding points remain on the street. This subdivision is to be expected given the stable foundations of the building but had to be made by a skilled operator on traditional maps.

Of the total of 3.1 million PS-InSAR points, less than 20% of the points did not match a Nearest Neighbour within $2\frac{1}{2}\sigma$. For surfaces the results are slightly better: 85% of the points was linked to a nearby surface within $2\frac{1}{2}\sigma$. Results further away are very unlikely, given the validity of the error model. Missing links are generally due to occlusions in the point cloud, for example on facades and in narrow streets, resulting from the different viewing geometries between the sensors. Some of them are due to faulty interpretation of the geometry, leading to plane estimates that do not provide a realistic point of intersection.

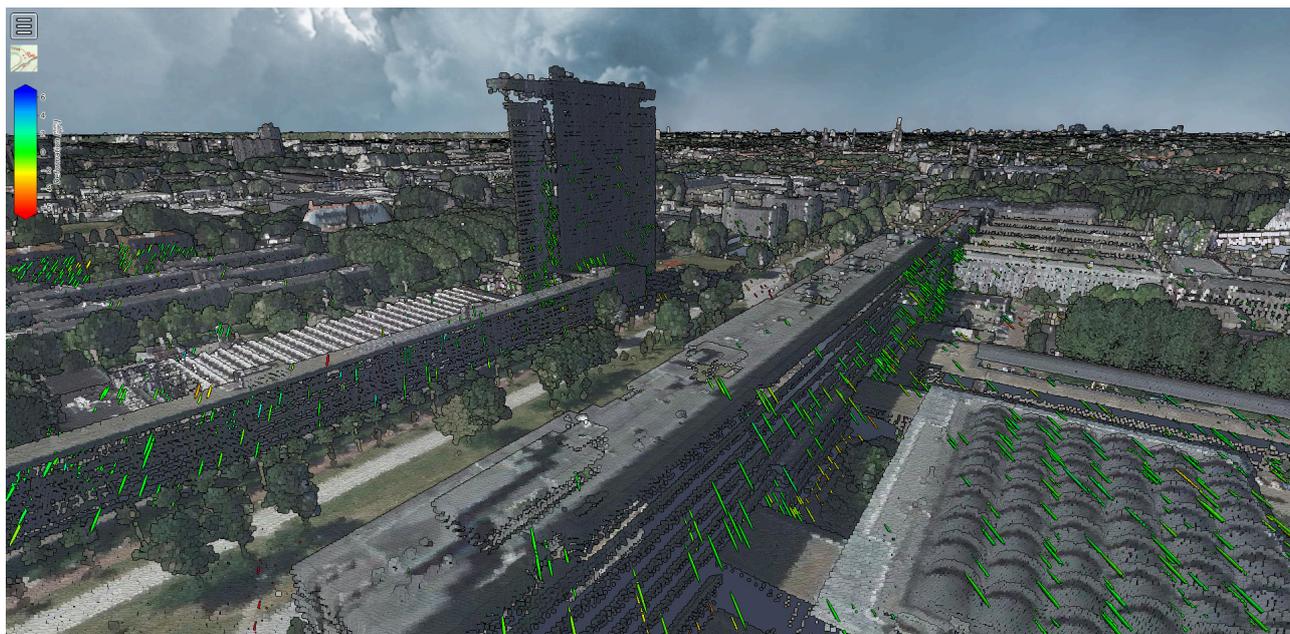


Figure 6. Delft University of Technology campus, shown using an adapted version of Potree, as a coloured point cloud (AHN3) overlaid with TerraSAR-X PS-InSAR trend estimates (shown as 1σ error ellipsoids around the expected position, coloured by their deformation signal).

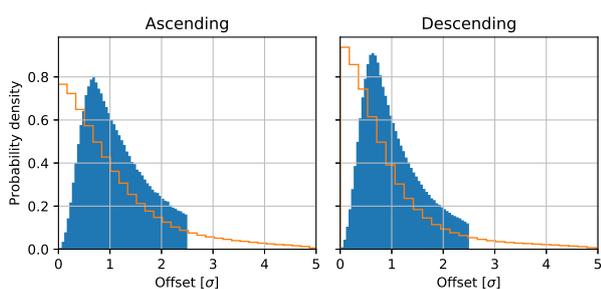


Figure 7. Distance to the closest object using various matching techniques. Blue: first Nearest Neighbour, orange: plane estimation. Nearest Neighbour search is stopped at $2\frac{1}{2}\sigma$.

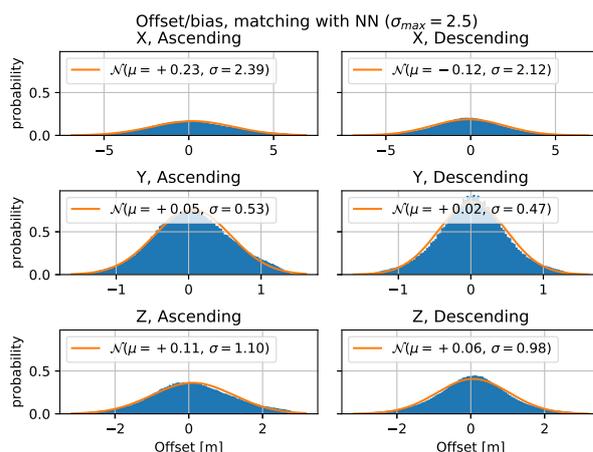


Figure 8. Offsets (biases) in matching, based on the first Nearest Neighbour.

4.2.1 Quality estimation Various quality estimates result from the matching procedure. For each point the distance from the original estimation is known, both measured in standard deviation (σ) and in world coordinates (metres). Furthermore for the surface estimation quality metrics are available, such as the flatness of surface (quality of the approximation). Indicating the quality of the fit for each individual surface.

4.3 Regional trends

Although the average offset is almost zero (figure 8), subsets of the data may be biased. By comparing the difference between the point of intersection or Nearest Neighbour and the original geolocation estimate with respect to location, trends become visible.

In figure 11 the median offsets between the geolocation estimation and Nearest Neighbour are shown (NN - estimation), binned in 100×100 metre bins. A clear north-south trend can be distinguished in the descending orbit, while in the geolocation of the ascending track the offset increases in the southern part. Due to

this trend geolocation estimates are off by 2 to 3 metres at the borders of the radar image.

These offsets and trends, that can be converted to radar coordinates too, may help to improve the radar processing.

4.4 Processing

Matching 1.4 million SAR points to 3 billion LiDAR points (122 tiles) using the Nearest Neighbour approach takes 15 minutes with three threads on a quad-core Intel i7-3630QM 2.4 GHz laptop computer with 24 GB of RAM. This includes opening the compressed LAZ-files and building of the transformed kD-tree. As the current script is written in python 3.5, higher performance is expected to be achieved by using more optimised programming languages.

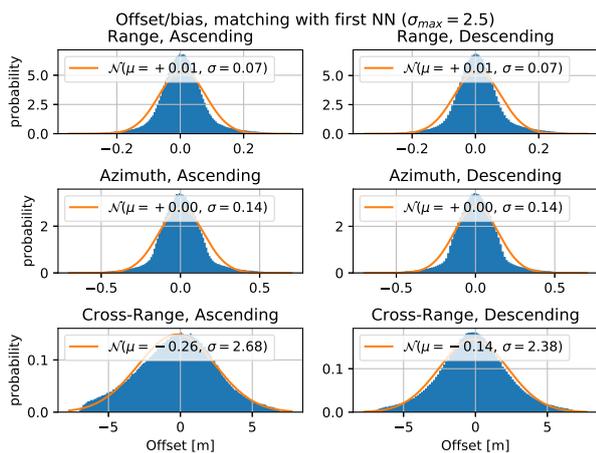


Figure 9. Offsets (biases) in matching, based on the first Nearest Neighbour, in radar coordinates.

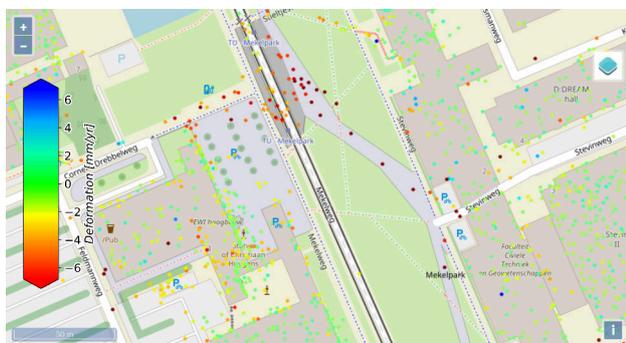


Figure 10. Point locations (mapped in 2D), after matching between the datasets. Compare to figure 2. (Background: OpenStreetMap)

With the research area spanning 122 tiles of a total of 30137 tiles, processing the nationwide dataset will take around sixty hours on this laptop. Given that the program can be scaled over many nodes the computation time can be reduced to the duration desired by adding more computing power.

5. DISCUSSION

In this article the geolocation error model is taken as constant over the whole region of the of the radar image and for all different scatterer types. This assumption is likely incorrect, as indicated by (Dheenathayalan et al., 2016). The incidence angle ranges from 20 to 45 degrees over the image and reflectors vary in type: from (round) lamppost to trihedral reflector. Each with different properties and likely different geolocation error estimations. When parameterised, regional variations in the error model can be included.

To improve the matching results the algorithm could be re-run with the biases obtained from the first run after initial matching. After correcting for the bias or trend in the initial geolocation the results of the second geolocation matching algorithm may find a different point of intersection. Quality assessment for this iterative process will be difficult, as the new geolocation will be the result of multiple transformations of the original dataset.

Given the classification provided with the LiDAR point cloud and the free availability of topographic maps in the Netherlands, it is possible to link deformation behaviour to outlines of buildings. Allowing the detection of differential deformation and the conversion from coordinate to addresses (geocoding).

Currently no single error model is implemented. The quality of the surface fit and the quality of the intersection are defined and estimated independently as two separate metrics. An integrated metric could provide, for example, a new quality metric for the point of intersection found.

6. CONCLUSION

The techniques introduced enable the efficient attribution of the InSAR deformation signal to real-world objects and features, allowing the next spatial join at a higher scale: linking individual signals to objects. The 3D visualisation will allow for better communication with the greater public, as less interpretation is required with respect to the traditional deformation maps.

Various options exist for the geometric linking of the two datasets. Implemented methods are based on the geometry and may not represent the underlying (physical) radar processes. Nevertheless they improve the geolocation estimate and aid in the attribution of the radar signal to real world objects.

ACKNOWLEDGEMENTS

We thank SkyGeo (<https://skygeo.com/>) for providing the processed TerraSAR-X data.

REFERENCES

- Bekendam, R. and Pottgens, J., 1995. Ground movements over the coal mines of southern Limburg, the Netherlands, and their relation to rising mine waters. *IAHS Publications-Series of Proceedings and Reports-Intern Assoc Hydrological Sciences* 234, pp. 3–12.
- Bell, A., Chambers, B., Butler, H., Gerlek, M. et al., 2018. PDAL - point data abstraction library. Hobu inc. <https://www.pdal.io> (March 2018).
- Boersma, C., 2015. The harder we pump, the faster we fall. *Delta Life*, pp. 10-13. Deltares periodical.
- CBS, 2017. Bestand bodemgebruik. <https://www.cbs.nl/nl-nl/dossier/nederland-regionaal/geografische%20data/natuur%20en%20milieu/bestand-bodemgebruik-2012> (March 2017).
- Chang, L. and Hanssen, R. F., 2014. Detection of cavity migration and sinkhole risk using radar interferometric time series. *Remote Sensing of Environment* 147, pp. 56–64.
- Dheenathayalan, P., Small, D., Schubert, A. and Hanssen, R. F., 2016. High-precision positioning of radar scatterers. *Journal of Geodesy* 90(5), pp. 403–422.
- Hanssen, R. F., 2001. *Radar interferometry: data interpretation and error analysis*. Vol. 2, Springer Science & Business Media.
- Isenburg, M., 2018. Lastools. rapidlasso GmbH <https://rapidlasso.com/LAStools/> (March 2018).
- Kadaster and Geonovum, 2018. Publieke Dienstverlening Op de Kaart (PDOK). <https://www.pdok.nl/> (March 2018).

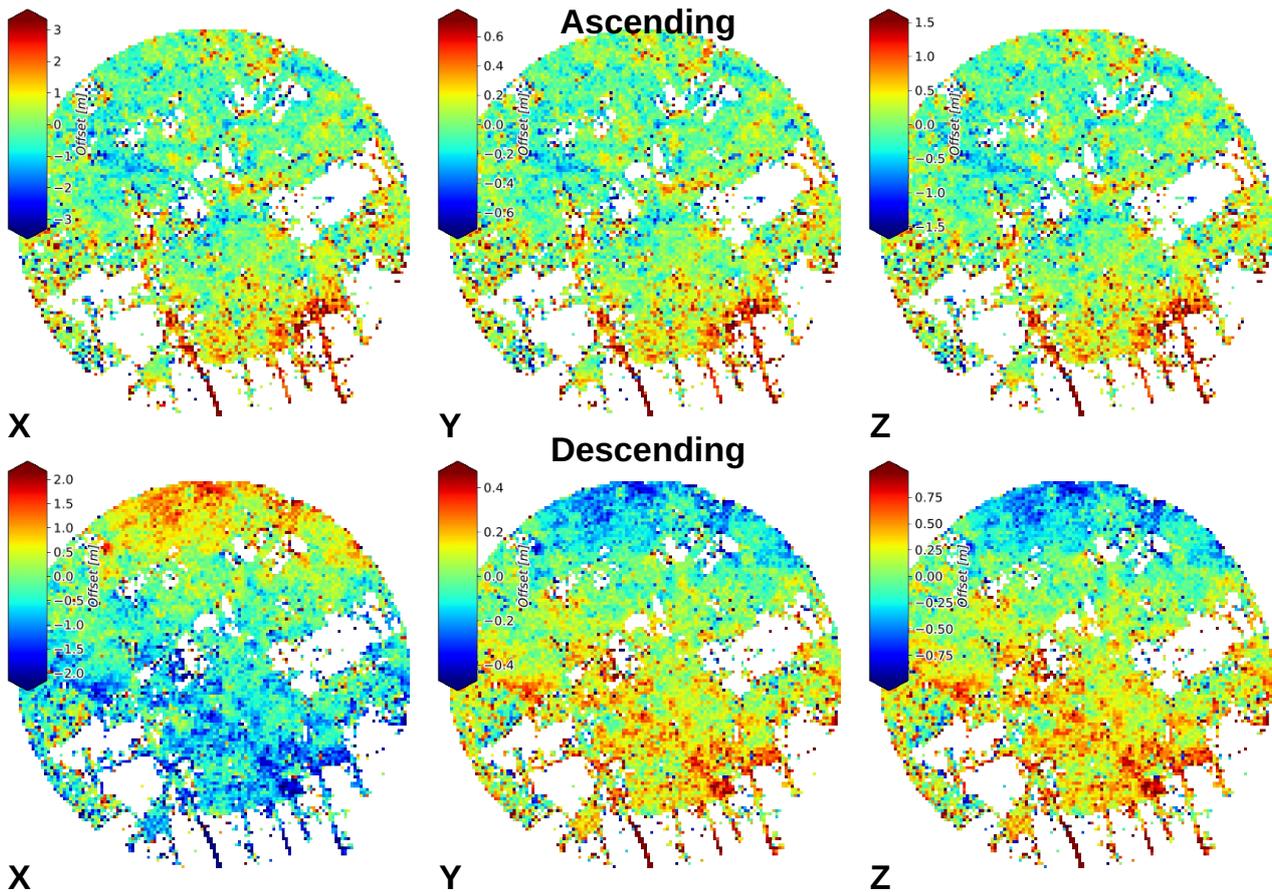


Figure 11. Regional trends (100 × 100 m median) in the offset between the geolocation estimation and the Nearest Neighbour (NN - est.) for both ascending and descending orbits.

Ketelaar, G., van Leijen, F., Marinkovic, P. and Hanssen, R., 2006. On the use of point target characteristics in the estimation of low subsidence rates due to gas extraction in Groningen, the Netherlands. In: *Fringe 2005 Workshop*, Vol. 610.

Martinez-Rubi, O., Verhoeven, S., Van Meersbergen, M., Schütz, M., Van Oosterom, P., Gonçalves, R. and Tijssen, T., 2015. Taming the beast: Free and open-source massive point cloud web visualization. In: *Capturing Reality Forum 2015, 23-25 November 2015, Salzburg, Austria*, The Survey Association.

Nielsen, E., Herb, J. et al., 2017. *pykdtree*: fast kd-tree implementation in Python. <https://github.com/storpipfugl/pykdtree> (March 2018).

Python Development Team, 2017. Python programming language. Python Software Foundation <https://www.python.org> (September 2017).

Richards, M. A., Scheer, J., Holm, W. A., Melvin, W. L. et al., 2010. *Principles of modern radar*. SciTech Publishing.

Sakurai, Y., Yoshikawa, M., Kataoka, R. and Uemura, S., 2001. Similarity search for adaptive ellipsoid queries using spatial transformation. In: *VLDB*, pp. 231–240.

Schmitt, M., Tupin, F. and Zhu, X. X., 2017. Fusion of SAR and optical remote sensing data - challenges and recent trends. In: *Proceedings of IEEE Geoscience and Remote Sensing Symposium*.

Schütz, M., 2016. *Potree: Rendering large point clouds in web browsers*. PhD thesis, Technische Universität Wien, Wien.

Schütz, M., 2018. *Potree*. <http://potree.org/> (January 2018).

Stansbury, D., 2013. The statistical whitening transform. <https://theclevermachine.wordpress.com/tag/eigenvalue-decomposition/> (March 2013).

Tupin, F., 2010. Fusion of optical and SAR images. In: *Radar Remote Sensing of Urban Areas*, Springer, pp. 133–159.

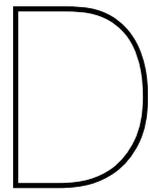
van Meijeren, H., 2017. Assessing the differences between Dutch elevation datasets AHN2 and AHN3. MSc thesis.

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

Vosselman, G. and Maas, H.-G., 2010. *Airborne and Terrestrial Laser Scanning*. Whittles Publishing.

Wang, Y., Zhu, X. X., Zeisl, B. and Pollefeys, M., 2017. Fusing meter-resolution 4-D InSAR point clouds and optical images for semantic urban infrastructure monitoring. *IEEE Transactions on Geoscience and Remote Sensing* 55(1), pp. 14–26.

Werninghaus, R. and Buckreuss, S., 2010. The TerraSAR-X mission and system design. *IEEE Transactions on Geoscience and Remote Sensing* 48(2), pp. 606–614.



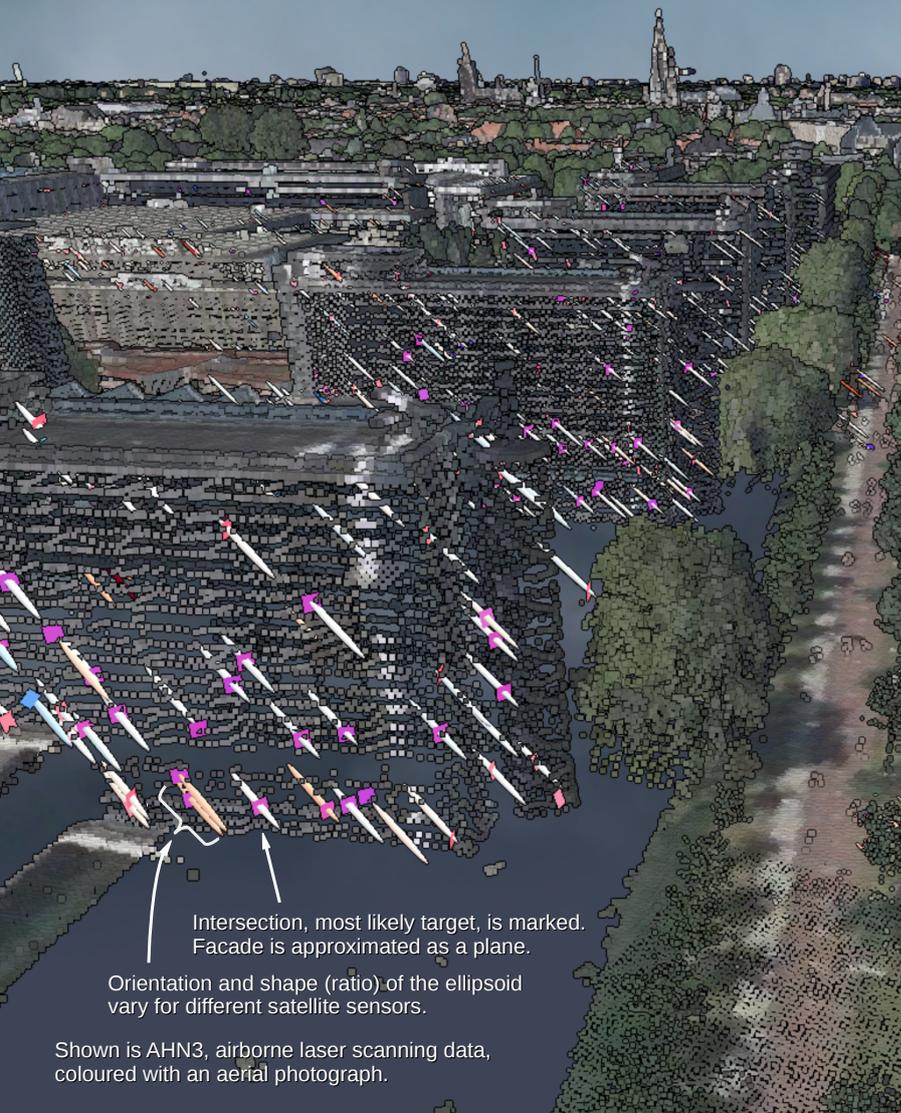
NAC Conference Poster

The following poster was presented during the NAC (Nederlands Aardwetenschappelijk Congres), 15-16 March 2018, in Veldhoven, the Netherlands.

Walking with your head in the (point) cloud

Object recognition using combined InSAR and LiDAR data

A.L. van Natijne, L. Chang, R.C. Lindenbergh, P. Dheenathayalan, R.F. Hanssen
Delft University of Technology



Intersection, most likely target, is marked.
Facade is approximated as a plane.

Orientation and shape (ratio) of the ellipsoid vary for different satellite sensors.

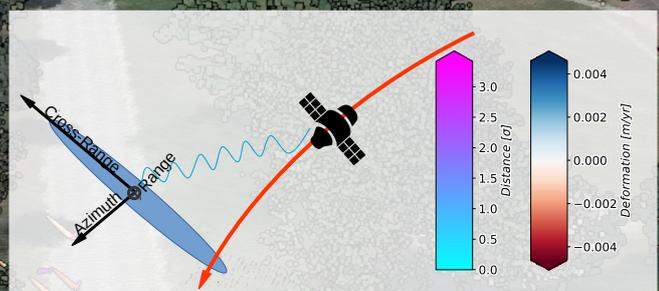
Shown is AHN3, airborne laser scanning data, coloured with an aerial photograph.

InSAR deformation monitoring suffers from relatively **poor geolocation**, making it difficult to attribute the deformation signal to a physical object.

Using the covariance structure of the geolocation (estimation), the dataset can be **joined to a high density LiDAR point cloud**. Results are shown in a web-portal, able to show vast amounts of information.

This approach provides:

- **Improved geolocation;**
- Ability to **link deformation signal to objects**, under the assumption that a single scatterer dominates the signal,
- **Visual feedback** on quality of results, both of the geolocation estimate as of the link between the data sources.



Positioning precision varies strongly in range, azimuth and cross-range and is dependent on the satellite, scatterer and processing. For TerraSAR-X the ratio of the error ellipsoid varies between 1/2/22 ($r/a/cr$) to 1/3/200 for corner reflectors. [Dheenathayalan et al., 2016]

This anisotropy of the error is accounted for in the spatial join. Both deformation behaviour (colour of the ellipsoid) and information on the match (colour of the plane of intersection) can be visualised in the viewer. Distance to the closest surface can be used as a quality indicator for the match.



Coloured AHN3 plus a mobile scan of the train track.

Possible applications

- Deformation measurements
Attributing signal to distinct objects, such as distinction between streets and buildings.
- Railway infrastructure
Early warning system for individual components of the rail infrastructure.
- Improvement of PS-InSAR processing, enabling 'in-situ' analysis.

In short

Goal: Improvement of geolocation of PS-InSAR by combining with LiDAR point clouds.

Which allows:

- Attribution of deformation signal to object(s);
- Visual and numerical detection of biases and other errors.

Reference

Dheenathayalan, P., Small, D., Schubert, A. et al. "High-precision positioning of radar scatterers" Journal of Geodesy (2016) 90:403

