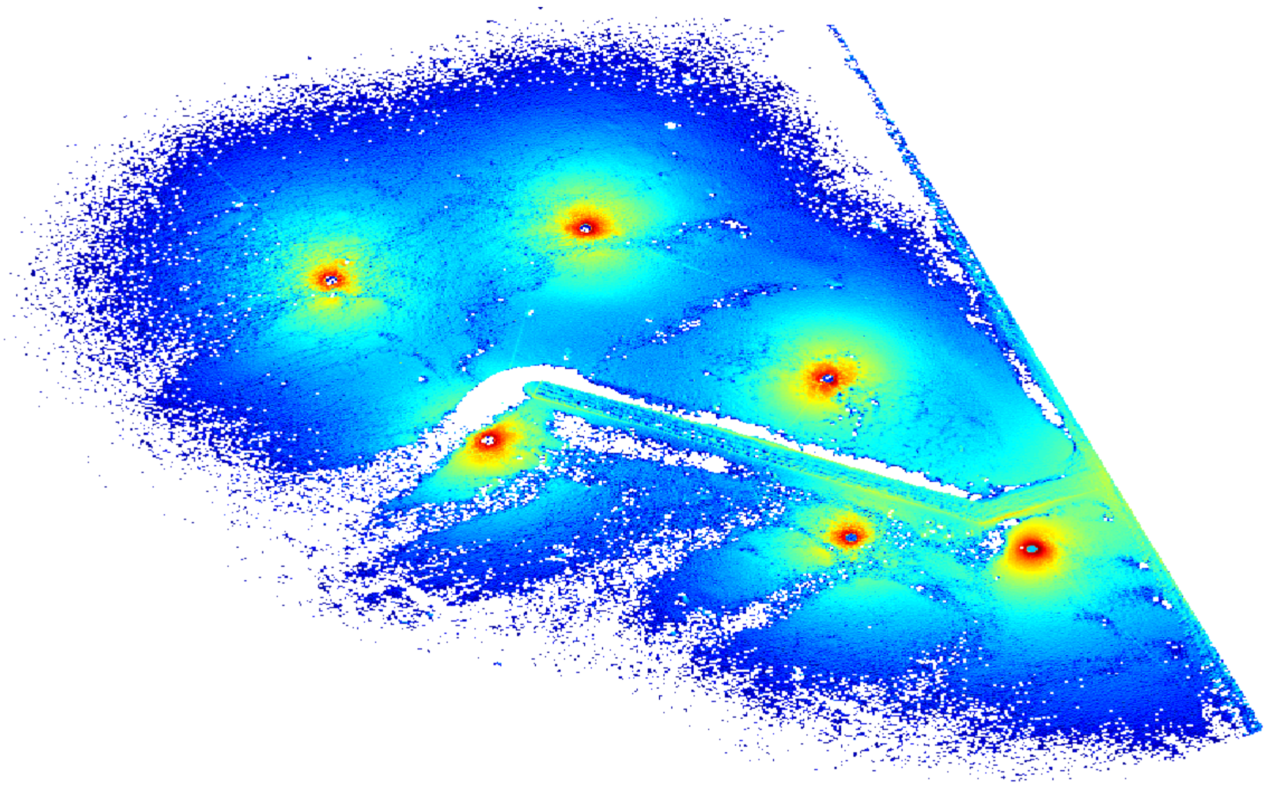
Detecting morphology changes due to oyster reefs in a tidal basin using Terrestrial Laser Scanning

A.H. Visser



BSc student at Delft University of Technology,
Delft, The Netherlands
A.H.Visser@student.tudelft.nl
Supervisors: Dr. R.C. Lindenbergh (TU Delft)
Dr. Ir. B.C. Van Prooijen (TU Delft)

Abstract

Several programs have been written to process laser scan data from two shoals in the Netherlands. The first makes a grid of heights, the second calculates deformations and makes plots. The third utilises a variogram to obtain information about the amplitude, direction and wavelength of tidal ripples, which are processed and plotted by a fourth program. Six scans have been made in order to compare the elevation of the scanner and the high-speed and long-range settings, from which it is found that the elevation is an important factor: a higher elevation results in a better coverage at mid- to long-range distances. For flat areas the long-range setting is not useful, as the area scanned increases by a factor two, whilst the scanning time triples. For areas with stark relief however, the long-range setting increases the scanned area with a factor nine.

Contents

Foreword

This paper is written as a bachelor thesis in the pursuit of the Bachelor of Science degree in Applied Earth Sciences at Delft University of Technology by Adriaan Visser under the supervision of Dr. R.C. Lindenbergh and Dr. Ir. B. C. van Prooijen. All the data used in this paper was scanned by Jeroen van Dalen of the NIOZ. I would like to thank Jeroen especially for his help and time to prepare the data for me and the help using the scanner myself. A special mention goes to Adriaan van Natijne, who helped me to obtain the AHN data.

This thesis concerns itself with the question of the NIOZ whether it is possible to detect changes in elevation on shoals and mud flats using laser scanning data and whether different settings of scanner range would drastically change the resolution and amount of data output.

Introduction

In I*ntroduction* the background of the project, laser scanning and the NIOZ will be discussed. After the research question is posed, the methods of obtaining and processing data are explained in *Methodology and data description*. In *Results* the results are presented and commented upon, after which in *Discussion* some difficulties and points of interest that emanated from the results are discussed. At last, in *Conclusions* the research questions will be answered.

Morphology *(countable and uncountable, plural morphologies)*

1.(uncountable) A scientific study of form and structure, usually without regard to function. Especially:

1. *(linguistics) The study of the internal structure of morphemes (words and their semantic building blocks).*
2. *(biology) The study of the form and structure of animals and plants.*
3. *(geology) The study of the structure of rocks and landforms.*

2.(countable) The form and structure of something.

3.(countable) A description of the form and structure of something.

As defined in the Wiktionary, a wiki-based open content dictionary

Morphology is the study of forms. Forms have always fascinated humans, in geology as well as in other fields. The surface of the earth is sculpted by many processes such as tectonics, glaciers, wind and water. Geomorphologists try to describe how a certain form has come to be, which processes were involved, what the time scale was over which the shaping took place and what the scale of the form is. Morphology is often studied in relation to human activities, such as the coast of South-Holland in the Netherlands.

A sand supplementation of more than one square kilometre of the coast of Delfland was done in 2011, a project known as 'De Zandmotor' (the sand-engine) (Zandmotor, 2014). The aim of this project is to have natural processes distribute the sand along the coast of South-Holland and thus offer protection from the sea. The movement of sand and shape changes of De Zandmotor, ergo its morphology, are closely monitored.

The shoals of Viane lie in the Eastern Scheldt tidal basin, which is a former estuary (figure 1). An estuary is a tidal inlet that runs inwards from the sea along a river to the highest point reached by the tide (Hugget, 2008). Being on this location means fresh water coming from the river is mixed with salt water from the sea, thus creating a brackish environment. As the shoals are located on the interface between a fluvial system and a tidal system they receive energy and sediment inputs from both sides (Fairbridge, 1980).

The building of the Kreekrakdam in 1867 for the railway to Zeeland closed off the Eastern Scheldt estuary from its tributary, the Scheldt river. This resulted in a much better water quality, as the polluted water from the Scheldt could no longer reach the Eastern Scheldt. This also resulted in a changing hydrological regime however, as there was no longer an influx of fresh water, therefore the Eastern Scheldt slowly became a salt water basin, influenced by the North Sea tides (Smaal et al 1992).

In 1986 the Deltawerken were realised, one of the largest coastal defence systems against flooding in the world. The Dutch insisted on keeping Zeelandic Flanders in the treaty of London in 1839, because they did not want to relinquish control over the Scheldt to Belgium, though the treaty stated that the Dutch had to provide and maintain access to the Port of Antwerp, thus the Western Scheldt had to remain open. The Eastern Scheldt, being too shallow for most seafaring vessels, was closed off by the Oosterscheldekering, reducing the tides from 3.40m to 3.25m (Smaal, 1992). The shoals of Viane and Val are bounded by dikes to the NE and NW, and open to the Eastern Scheldt at the others. The main direction of flow is from the NE, going to West (figure 1).

On 8 May 2002 the Eastern Scheldt became a National Park, the largest of The Netherlands. In a National Park, ecosystems and landscapes have to be preserved as much as possible. Therefore the NIOZ started the pilot with oyster-reefs to preserve the landscape. The Pacific oyster (*Crassostrea gigas*), introduced in The Netherlands by fishermen in the 1960s, is the oyster of choice for building reefs. A thick substrate of dead oysters is placed on the shoal, over which meshed wire is stretched to prevent the oysters from washing away. Over time oysters will attach themselves to the reef an be able to keep the integral structure themselves, whilst the meshed wire corrodes away. The three dimensional reefs built by the Pacific oyster are strong enough to withstand the waves, even during storms. Initial observations have already indicated a drastic decrease in erosion and even a change to deposition behind the reefs (Ecoshape, 2009).

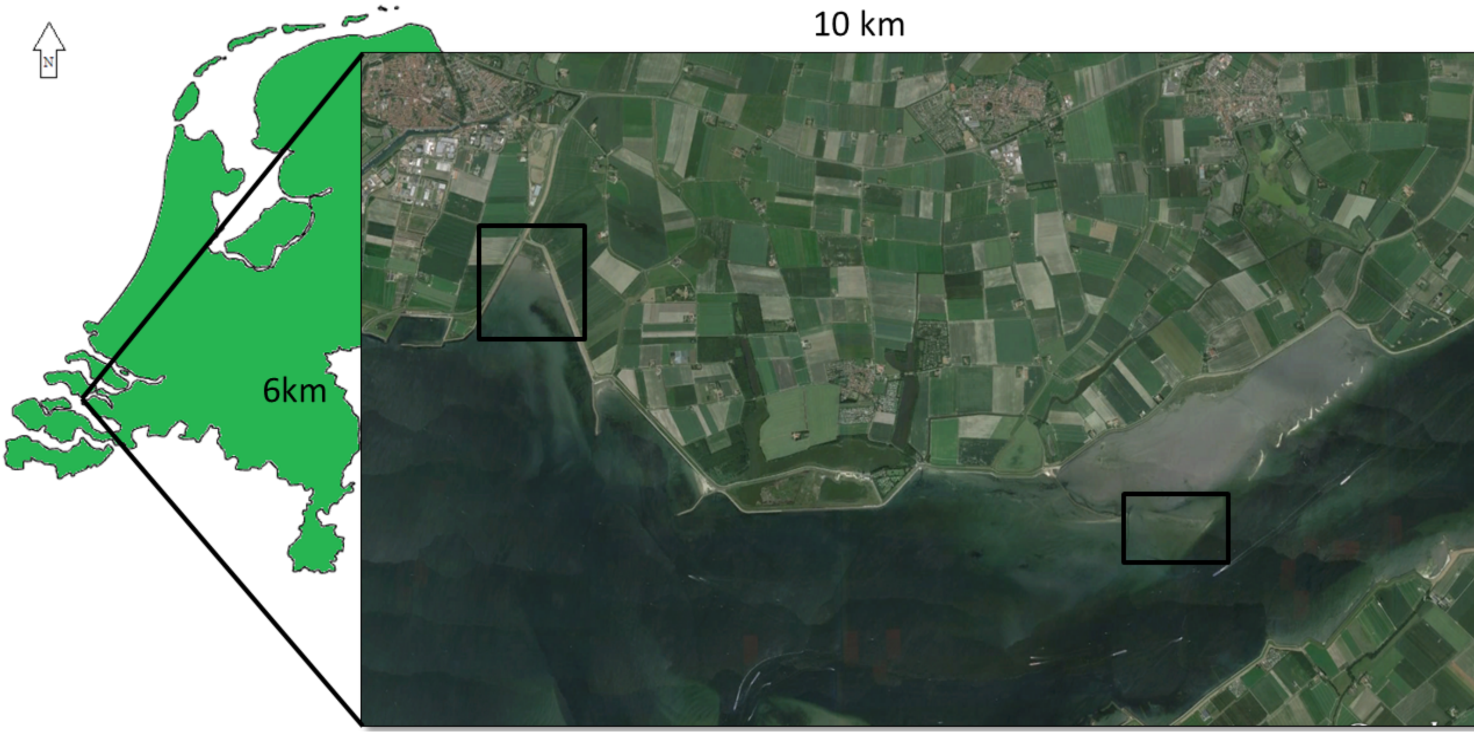


Figure 1: location of the shoals of Viane and Val within The Netherlands, the black boxes are the two scanned areas. Satellite image courtesy of Google Earth, image date 8-7-2013.

Laser scanning

As humans are more and more actively changing nature, monitoring of these changes has become an important science, as one needs to know if the changes have the desired effect. Usually a surface scanning is the most useful technique to determine changes, and laser scanning has become one of the most important monitoring methods for surface deformation. The method is based upon the travel time of a pulse of light through a given medium, usually air. With the velocity of light through air known, a measured time delay can be translated to a distance. Lasers that use this kind of measuring technique are called time-of-flight scanners (Vosselman, Maas 2010). This distance is the length that the laser pulse has travelled from the source to the reflection point on the object and the same distance back, as the detector is placed at almost the same point as the source in a laser scanner. The pulse is sent out into a specific direction, which is represented by the rotational angle and dip of the source, which are known. This results in a spherical coordinate system, (r,θ,φ), where r is the travelled distance, θ denotes the dip and φ the rotational angle. These coordinates can then be tied to a reference system, if the position of the scanner in that system is known. The spherical coordinates created by the scanner do not reflect the scanned area's location on the globe, which is why it is useful to change from the spherical coordinates to a larger, usually Cartesian coordinate system, so that several scans made beside each other spatially can be connected in the same coordinate system. This is usually done automatically by the scanner, as most of them are nowadays equipped with a GPS sensor.

With up to several hundred thousand points per second scanned, this results in a huge point density and high speeds at which objects can be scanned. Data sets of more than 50 million points are therefore no exception. The difficulty in processing these data lies in the amount and precision. As subsequent scans do not scan the same point every time, the data has to be interpolated to form surfaces in order to study deformation and changes over time. The amount of data makes processing difficult, as computational times increase. Several data structures have been developed to speed up computational times, such as Delaunay triangulation (Okabe et al 1992), Octrees and k-D trees (Sedgewick 1992). The surfaces that are discussed in this paper are almost flat, so a 2D representation has been chosen, binning heights in a regular region quadtree, a grid of equal area cells, which are square in this case.

NIOZ

NIOZ, *koninklijk Nederlands Instituut voor Onderzoek der Zee*, royal Dutch Institute for Marine Research, is an institute founded in 1872 and therefore one of the oldest oceanographic research centres in the world. They started out concerning themselves with mainly marine life, gradually expanding to encompass more of the aspects of the ocean, such as oceanographic physics, marine biochemistry and geology. In 2012 the group Ecosystem studies, Marine microbiology and spatial ecology of the Dutch Institute for Ecology merged with the NIOZ. This group, based in Yerseke, Zeeland, provided the data for this paper.

The Spatial Ecology group in Yerseke concerns itself with biophysical interactions and their significance to the biological diversity in estuaries. They map out portions of land using remote sensing techniques, amongst which terrestrial laser scanning, trying to find a connection between the physics of the environment, such as currents and tides, and the spatial distribution of large clumps of marine life, such as oysters and mussels.

Research question

A lot of research has been done in the field of laser scanning and detecting changes with laser scanning. All methods discussed in this paper are already in use by several research groups; the implementation into a matlab script and the quantitative comparison of scanner elevation and heights have been conducted by the author.

The main question of this paper is:

How can terrestrial laser scanning be used detect morphological changes?

In order to answer this question, several sub-questions will be addressed:

- How does one acquire height data using laser scanning?
- What are the difficulties in data acquisition?
- What kind of signals can be distinguished?
- How does one determine changes?
- How does one process the data?

These questions will be answered in order to come up with a measurement protocol of how and when to measure in order to obtain good quality data.

Methodology and data description

In this section the method of data acquisition is discussed, as well as the scanner used and how the data is output. A description of the data is given, with general remarks about the general morphology visible after visualising the raw data. The techniques for deformation testing and the usage of variograms to obtain parameters in bed forms which exhibit a form of periodicity are explained. Finally a qualitative description of the computational algorithm is given.

Acquisition method

The data were obtained using a Riegl VZ-400 scanner (figure 2) (Riegl LMS, 2013). This scanner has a range up to 600m, with a repeatable precision of 3mm and accuracy of 5mm. 42.000 measurements per second can be made, 360° round and 100° tilt. Using the high speed mode, 122.000 measurements can be made with a range of 160m. The scanner has an integrated GPS receiver, which automatically converts the spherical scanning coordinates to any given system; the *Rijksdriehoekscoördinaten* for this purpose (Kadaster, PDOK).

The shoal is scanned utilising several laser scanner positions, 11 for Viane, 6 for Val (figure 4 a,b) starting at the high ground and gradually moving towards the low lying areas when it is low tide. Each scan takes about five to six minutes to complete, then the scanner is moved to the next location. The scanner is positioned on a small cart for manoeuvrability, this way the scanner can be moved with relative ease and it does not have to be completely dismounted and remounted for each scan. The weather conditions are similar during each scanning period, as the scanner cannot function properly in rain or strong winds, because rain distorts the laser beam, whilst sway due to wind makes the coordinates uncertain. The internal gyroscope can correct for deviations from level up to a couple of degrees, so a small tilt in set up positions is not a problem.

Figure 2: Riegl VZ-400 V-Line 3D terrestrial laser scanner, Riegl

The separate scans are tied together using small reflective discs placed on poles across the shoal. These can easily be found in the dataset, as they result in a small area a large distance above the ground level. Using Riegl software RiScanPro coordinates are exported with only six digits, so the x- and y-coordinates were translated from the original RD coordinate-system to a project coordinate system, moving it 60.4km to the east and 403km to the north for the shoal of Viane. This resulted in coordinates between -36 and 900m, with three decimals, thus millimetre accuracy. The data was stored as an ASCII table, containing the (x,y,z) coordinates accurate to 3 decimals, thus millimetres, of the point. In the scan from 6 November 2012 there were 792 points up to 36 metres west of the new origin, these were not used for calculation purposes, since the coordinates are used as matrix subscripts and those cannot be zero or negative.

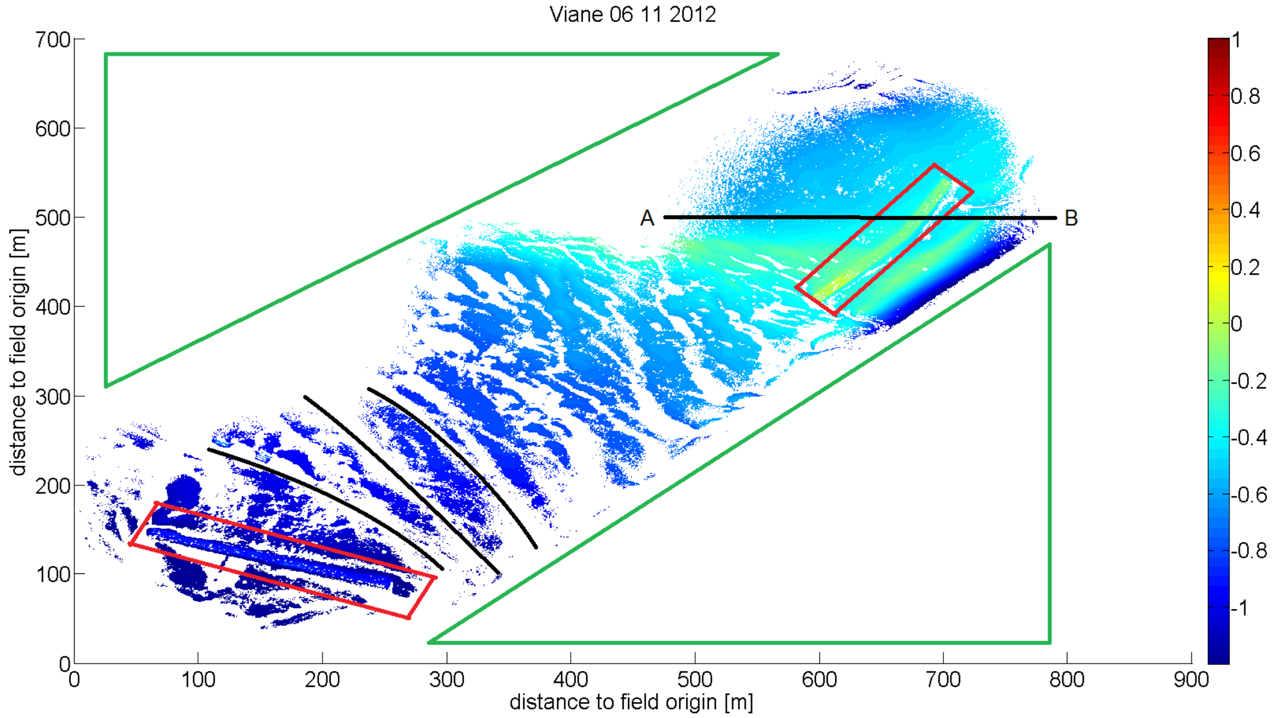
In order to test the settings and elevation of the scanner six scans have been made. Two at each location, one using the high-speed setting with which the NIOZ usually works, one with the long-range mode; this mode takes three times longer to scan the area. Three locations have been scanned: one outside the dike, one on top of the dike (5.5m elevation) towards a fallow field, and one below the dike towards the same field (-1m elevation).

Data description

There are two main areas of interest on the shoal of Viane, figure 3 a: the higher area in the north eastern part and a lower area in the south west. As can be seen from figure 3 a there are a couple of north west – south east trending gullies without data through the map, the black lines in the SW part. This is probably due to to the gullies being deeper than the tides, thus having water on the lowest areas. The laser scanner has trouble penetrating water, as it is highly reflective and absorbent. The ray thus either bounces off the water surface in the other direction instead of reflecting back to its point of origin, being the scanner, or is completely absorbed by the water, not returning to the scanner either. This is supported by the scan on 24 October 2013, figure 3b, where the gullies do not appear, but are filled in with sediments as low as 1.2m below NAP (*Nieuw Amsterdams Peil, Dutch ordinance level ≈ mean sea level)*, which was the tide at that time. These areas would be under water at all times but low tide. The oyster reef is clearly visible in the red marked areas.

The shoal of Val has a few slightly different structures (figure 4b). The straight line of approximately 5m wide is the oyster-reef, which is connected to the dike, which is the straight structure on the eastern side of the data. On this shoal there are a couple of gullies, the same as on Viane.

Due to the method of scanning, the point density is the highest the closest to the scanner. The shoal of Val is around half the size of the shoal of Viane, thus requires half the scans in order to fully scan the shoal. The shoal at Viane lies a bit lower, with the southern reef at -1.1m NAP, therefore a lot of gullies still contain water and do not contain data. The shoal of Val has a good area coverage, with the notable exceptions of the gullies and the north-west of the shoal, which is out of range of the scanner. As can bee seen from figure 4, the area coverage degrades when moving away from the scanner, this will be discussed in *Discussion-grid size* and *-Scanner elevation and settings*.



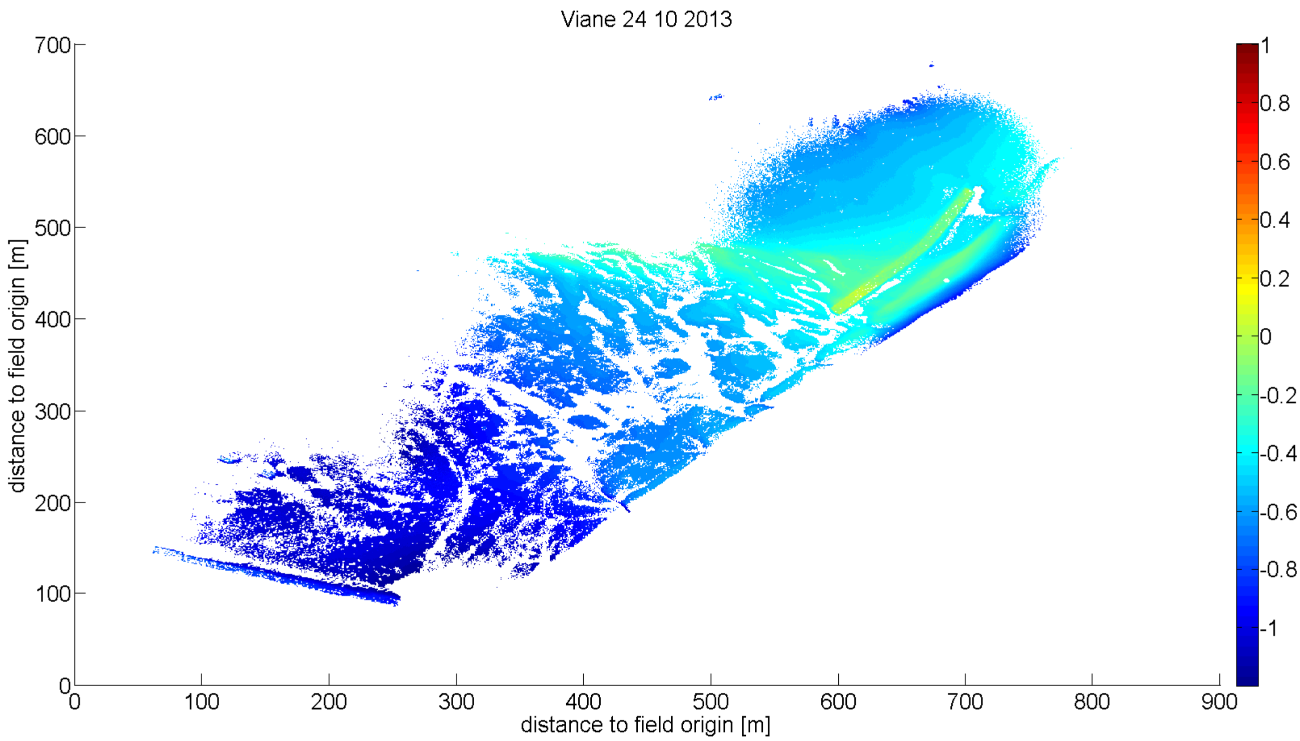
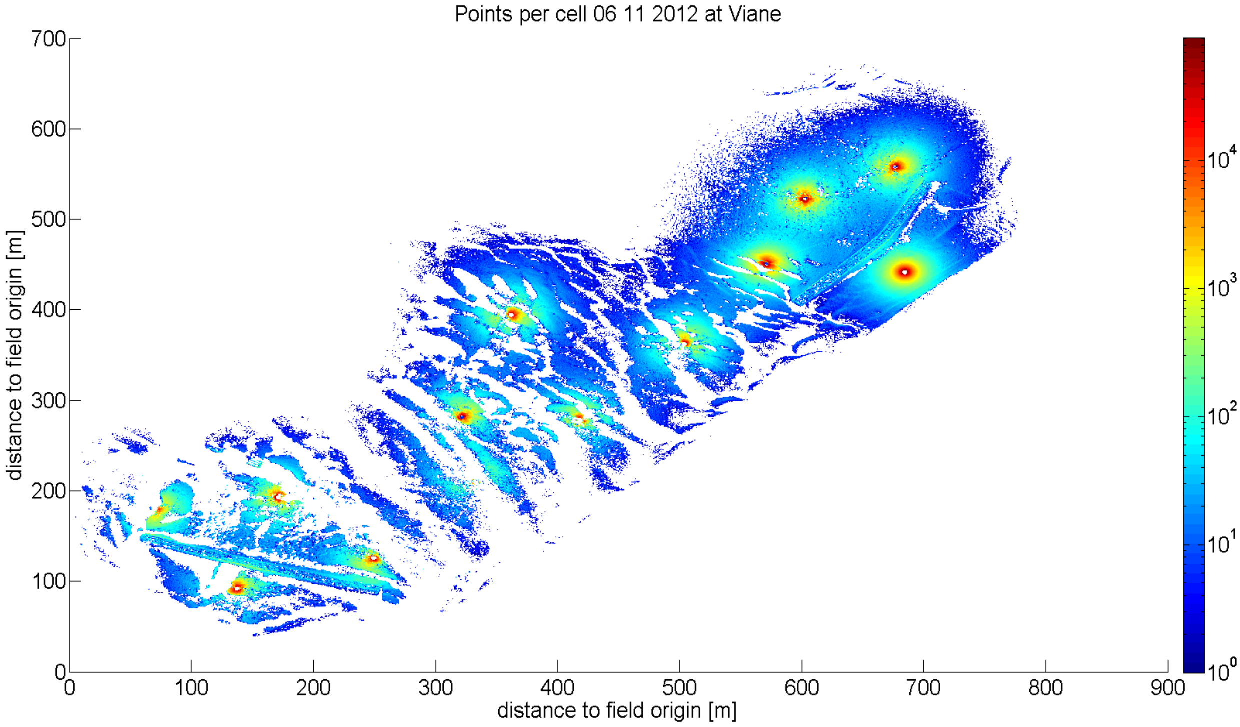


Figure 3a, b: height scan from Viane at 6-11-2012 at 24-10-2013. The data is put into 0.25m² grid cells, the colour scale ranges between -1.2m to 1m relative to NAP.



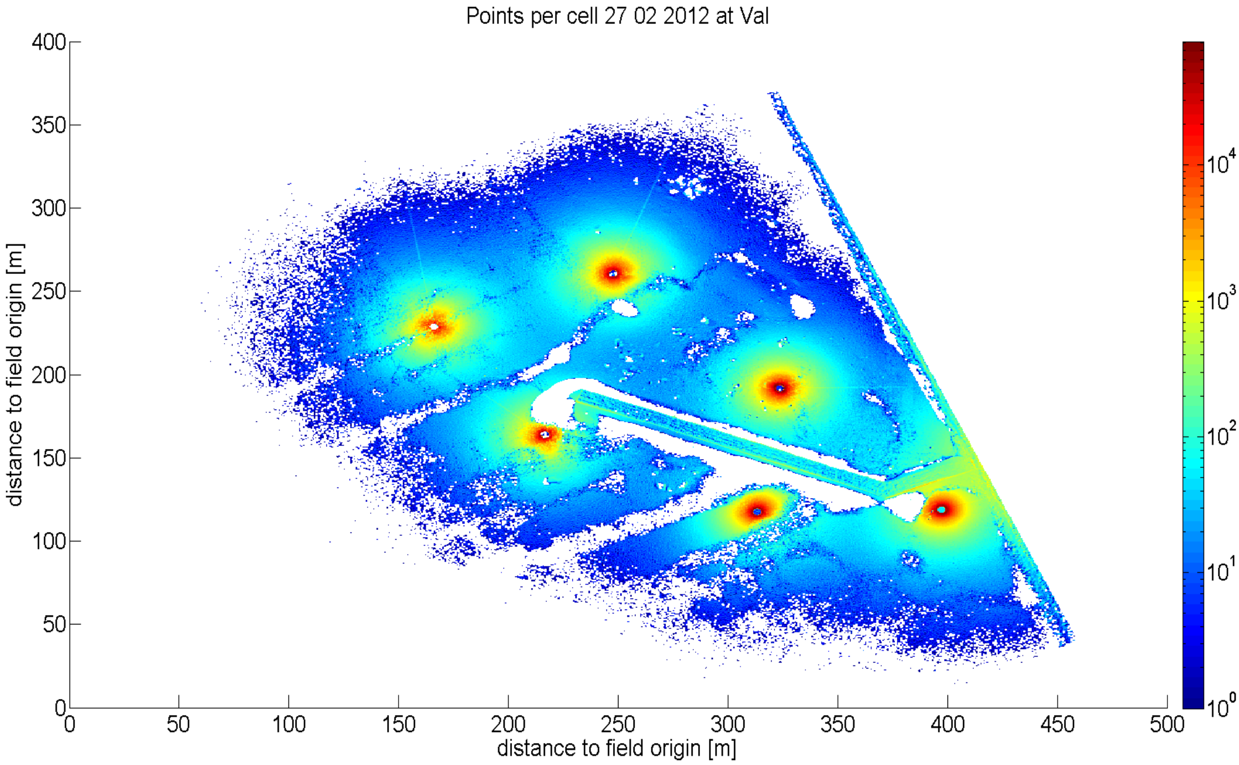


Figure 4a,b: point density per grid cell at 6 11 2012 at Viane and at 27 02 2012 at Val. The colour scale is logarithmic, from 10^0 to $3 \cdot 10^4$.

Deformation testing

The ultimate goal of long-term laser scanning is to determine changes between subsequent scanning epochs. Looking at pure difference, thus one set minus the other, does not take the statistics of the data into account. For instance the laser scanner itself has a certain accuracy and because measurements are grouped into cells, there will be a standard deviation per cell. If one would not take the statistics into account, all changes would be classified as actual differences between the two epochs, whilst for instance a difference of 3mm between two cells which each contain one point can be due to the scanner accuracy of 3mm. Therefore a statistical deformation test is used, a stability test (Teunissen, 2000b). This will be programmed in Matlab, as will be discussed later on.

In this test the A-model models the situation of no deformation on point *r* between epochs *1* and *2*, which implies that the two observations r_1 and r_2 carry the same attribute, being the height in this case, r_s. The observations r_i are heights of the same cell in their respective epoch *i*. This gives the following system:

$$E\left\{\begin{matrix}\underline{r_1}\\ \underline{r_2}\end{matrix}\right\}=A_s\cdot r_s\,;\,A_s=\begin{pmatrix}1\\1\end{pmatrix};Q_y=\begin{pmatrix}\sigma_{\underline{r_1}}^{\ 2} & 0\\ 0 & \sigma_{\underline{r_2}}^{\ 2}\end{pmatrix}\quad (1)$$

where *E{}* is the vector of observations $\underline{r}_i$, A_s the stable model and Q_y the matrix of variances σ^2_{ri} of the measurements r_i. This gives the adjusted, common distance and residual vector. The adjusted, common distance is the distance between the two heights weighted with their respective standard deviations; the residual vector is the vector containing the residual distance from the point r_i to the weighted midpoint between the two.

$$r_s=(A_s^T\cdot Q_y^{-1}\cdot A_s)^{-1}\cdot A_s^T\cdot Q_y^{-1}\cdot\begin{pmatrix}r_1\\r_2\end{pmatrix};\underline{\hat{e}}=\begin{pmatrix}\underline{r_1}\\ \underline{r_2}\end{pmatrix}-\begin{pmatrix}r_s\\r_s\end{pmatrix}\quad (2)$$

The zero hypothesis H_0 states that the given residuals are within bounds of the accuracy of the measurement and thus cannot be classified as a deformation area. The test quantity is calculated as:

$$\underline{T_q}=\underline{\hat{e}}^T\cdot Q_y^{-1}\cdot\underline{\hat{e}}\quad (3)$$

with q as the degrees of freedom. There are two epochs in this test, with one test quantity, height, giving a total of 2 - 1 = 1 degree of freedom. The H_0-hypothesis thus has a chi-squared distribution with 1 degree of freedom:

$$H_0:\underline{T_q}\sim\chi^2(q,0)=\underline{T_1}\sim\chi^2(1,0)\quad (4)$$

Expanding these relations to a direct scalar formula results in:

$$\hat{r}_s=\left(\begin{bmatrix}1 & 1\end{bmatrix}\begin{bmatrix}\frac{1}{\sigma_{r_1}^2} & 0\\ 0 & \frac{1}{\sigma_{r_2}^2}\end{bmatrix}\begin{bmatrix}1\\1\end{bmatrix}\right)^{-1}\cdot\begin{bmatrix}1 & 1\end{bmatrix}\begin{bmatrix}\frac{1}{\sigma_{r_1}^2} & 0\\ 0 & \frac{1}{\sigma_{r_2}^2}\end{bmatrix}\begin{bmatrix}\underline{r_1}\\ \underline{r_2}\end{bmatrix}$$

$$\hat{r}_s = \left(\frac{1}{\sigma^2_{r_1}} + \frac{1}{\sigma^2_{r_2}}\right)^{-1} \cdot \left(\frac{\underline{r_1}}{\sigma^2_{r_1}} + \frac{\underline{r_2}}{\sigma^2_{r_2}}\right) = \frac{\sigma^2_{r_1}\sigma^2_{r_2}}{\sigma^2_{r_1}+\sigma^2_{r_2}} + \frac{\sigma^2_{r_2}\underline{r_1}+\sigma^2_{r_1}\underline{r_2}}{\sigma^2_{r_1}\sigma^2_{r_2}} = \frac{\sigma^2_{r_2}\underline{r_1}+\sigma^2_{r_1}\underline{r_2}}{\sigma^2_{r_1}+\sigma^2_{r_2}} \quad (5)$$

$$\underline{T_q} = \underline{\hat{e}^T} \cdot Q_y^{-1} \cdot \underline{\hat{e}} = \begin{bmatrix} \underline{r_1} - r_s & \underline{r_2} - r_s \end{bmatrix} \begin{bmatrix} \frac{1}{\sigma^2_{r_1}} & 0 \\ 0 & \frac{1}{\sigma^2_{r_2}} \end{bmatrix} \begin{bmatrix} \underline{r_1} - r_s \\ \underline{r_2} - r_s \end{bmatrix} = \frac{(\underline{r_1} - r_s)^2}{\sigma^2_{r_1}} + \frac{(\underline{r_2} - r_s)^2}{\sigma^2_{r_2}} \quad (6)$$

combining equations (5) and (6) results in the final equation for $\underline{T}_q$:

$$\underline{T_q} = \frac{\left(\underline{r_1} - \left(\frac{\sigma^2_{r_1}\underline{r_2} + \sigma^2_{r_2}\underline{r_1}}{\sigma^2_{r_1}+\sigma^2_{r_2}}\right)\right)^2}{\sigma^2_{r_1}} + \frac{\left(\underline{r_2} - \left(\frac{\sigma^2_{r_1}r_2 + \sigma^2_{r_2}\underline{r_1}}{\sigma^2_{r_1}+\sigma^2_{r_2}}\right)\right)^2}{\sigma^2_{r_2}} \quad (7)$$

where $\underline{r}_i$ is the observation value, thus the height of the cell in epoch *i*, and σ_i is the standard deviation of the observations in the cell at epoch *i*. The reason for doing this calculation by hand is that it is required to evaluate for each cell, thus many million times. By hand-calculating the scalar expression, Matlab will not need to evaluate equations (1-3) a million times, but only equation (7), which consists of the basic operations of plus, minus, division and multiplication (squaring), which require less computational power than matrix operations like inversions.

The quantity $\underline{T}_q$ is then checked against a critical value k_α, So $\underline{T}_q > k_\alpha$, where α denotes the reliability level of the test (figure 5). Using this criterion two times of errors can be made: omission and commission errors. An omission error occurs when a deformed point is falsely omitted from being classified as deformed, whilst a commission error occurs when a non-deformed point is classified as being deformed. The value α states that with 99% certainty the point classified as deformed is actually deformed, this thus denotes that the omission error is 1%. By decreasing the chance for omission errors however, the chance commission error rises. There is no clear relationship that states by how much the commission error will change upon changing the omission error however. It thus is a trade-off between the two. The value for α has been chosen as 0.01%, thus k_α = 6.6349.

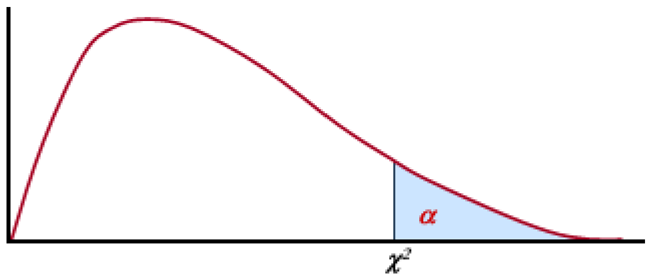


Figure 5: the chi-squared distribution with α denoting the critical region.

Variogram

A variogram can be used to determine the amplitude, wavelength and direction of spatial forms which exhibit a form of periodicity, tidal ripples in this case, see for instance figure 21b, the NW-SE trending bodies in are the tidal ripples. A variogram 2γ(x,y,h) is a function describing the degree of spatial dependence of a field (x,y) of observations (z) (Cressie 1993). A variogram usually requires of the data that it is trend-less, i.e. there is no correlation between the observations and their respective locations. This means that for instance a continuous slope will not result in a proper variogram, as the heights are decreasing downhill. The trend however can also be useful and modelled with a variogram, as is done in this paper.

A variogram is made by first creating the variogram cloud of distance versus height, then grouping that cloud per distance and taking the group average (figure 6). The horizontal axis in figure 6 is the distance between the points, the vertical axis is the correlation between the points at that distance. Each pair of points results in one point in the graph, thus there are N(N-1) total points. If desired, one can fit a continuous function through the points resulting in a theoretical variogram. The variogram is usually described by three parameters: the nugget, sill and range. The nugget is the y-intercept, due to measurement errors, the range is the distance at which variations do not become larger and the sill is the observation value at which this occurs. The experimental variogram is calculated as the average of all observations in a separation distance interval $||h|| = ||p_i - p_j||$, using k bins to group the separation distance (Pluymaekers, 2007):

$$\gamma([h_k]) = \frac{1}{2N_k} \sum_{(z_i, z_j) \in [h_k]} (z_i - z_j)^2 \quad (8)$$

where $[h_k]$ is the k-th distance group with respect to the horizontal distance between p_i and p_j in the data and N_k is the number of points in that group. This variogram can be made per direction, with the direction as the angle between the two points and north, between 0 and 180°.

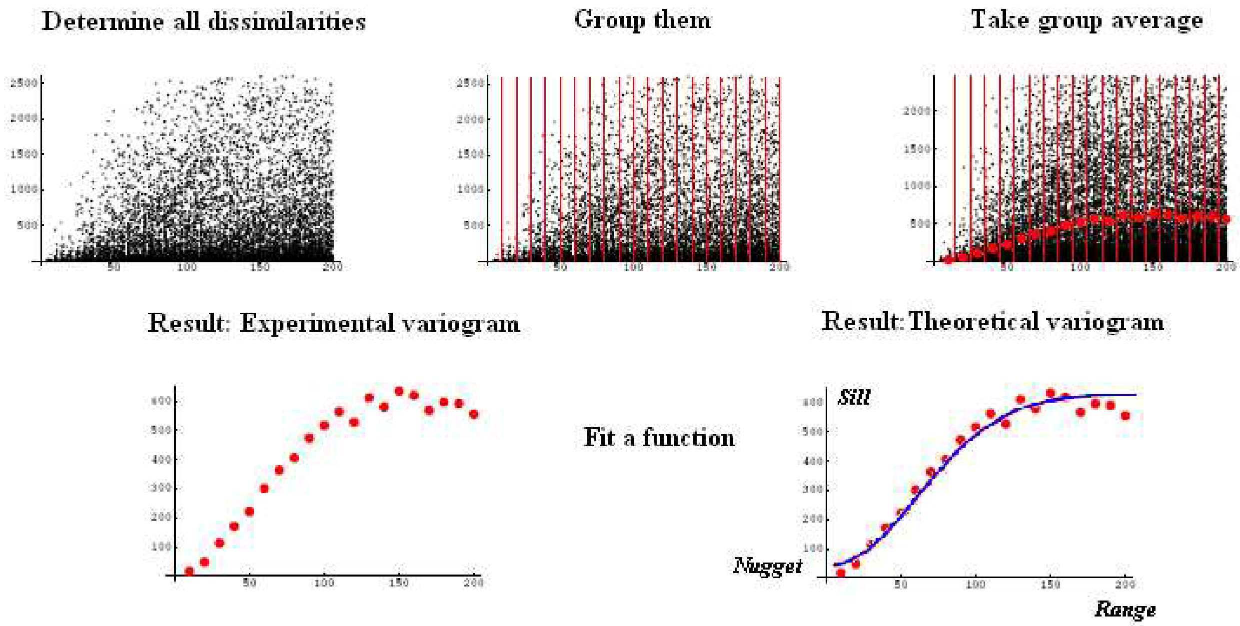


Figure 6: the steps for the computation of variograms. Source: [Lecture notes Multivariate data analysis] (Pluymakers, 2007)

If the data does have a trend in it, a hole-effect can occur if the data has a form of periodicity in it (Pyrcz, 2003). Figure 7 shows this hole-effect for waves on the water. The yellow and red lines are respectively 3 and 4 wavelengths apart, resulting in a higher than average correlation, as all points are on the crests of waves. The grey line shows half a wavelength, thus at this distance the correlation is lower than average, as one point is on the wave crest whilst the other is in the wave through. The periodic forms being sought here are the tidal ripples with a wavelength of approximately 3cm and an amplitude of 1cm. These parameters can be obtained from the directional variogram.

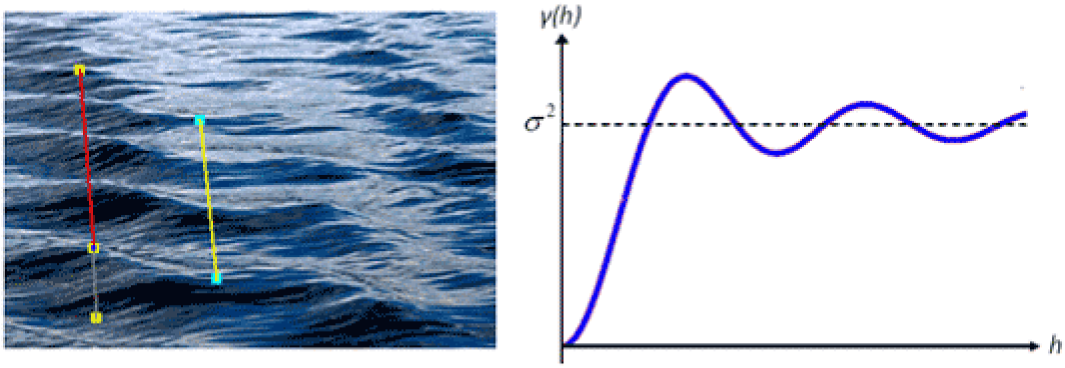


Figure 7: variogram hole-effect (Brusilovskiy, 2009).

Parameter estimation

The orientation of the ripples can be found in the directional variogram perpendicular to the direction of least difference. This is because perpendicular to the dune orientation, thus along its crest, the average variability will be minimal and the periodicity disappears, thus no hole-effect occurs.

The variogram is calculated as half of the square of the maximum variability of points at that distance; and the amplitude is defined as half of the range between the maximum and the minimum. It suffices to take the square root of the maximum of the directional variogram in the direction of the ripples:

$$A=\sqrt{(max(\gamma(h_k)))} \quad (9)$$

The wavelength of the ripples can be determined as well: the dune crest is at the maximum of the variogram, after which it drops to a local minimum. Points separated by a full wavelength will all be peaks, or troughs or wherever at the same relative point on a ripple. These points will have a fairly mutual small variability, whilst points separated a half wavelength, i.e. peaks and troughs, have a high mutual variability. The distance from the origin of the variogram to the minimum of the first hole-effect is thus the wavelength of the ripple (Pluymaekers, 2007).

The above described procedure is illustrated on the basis of one cell in the dataset of Viane at 6 November 2012, with centre point (683.25;445.75). The point cloud in this cell numbers 15179 points (figure 8). Because of the many points in this cell a sub sample containing 3000 random points will be taken. The directional variogram is then made (figure 9). As can be seen in figure 8, there is a trend in the data, so the ripple parameters can be calculated.

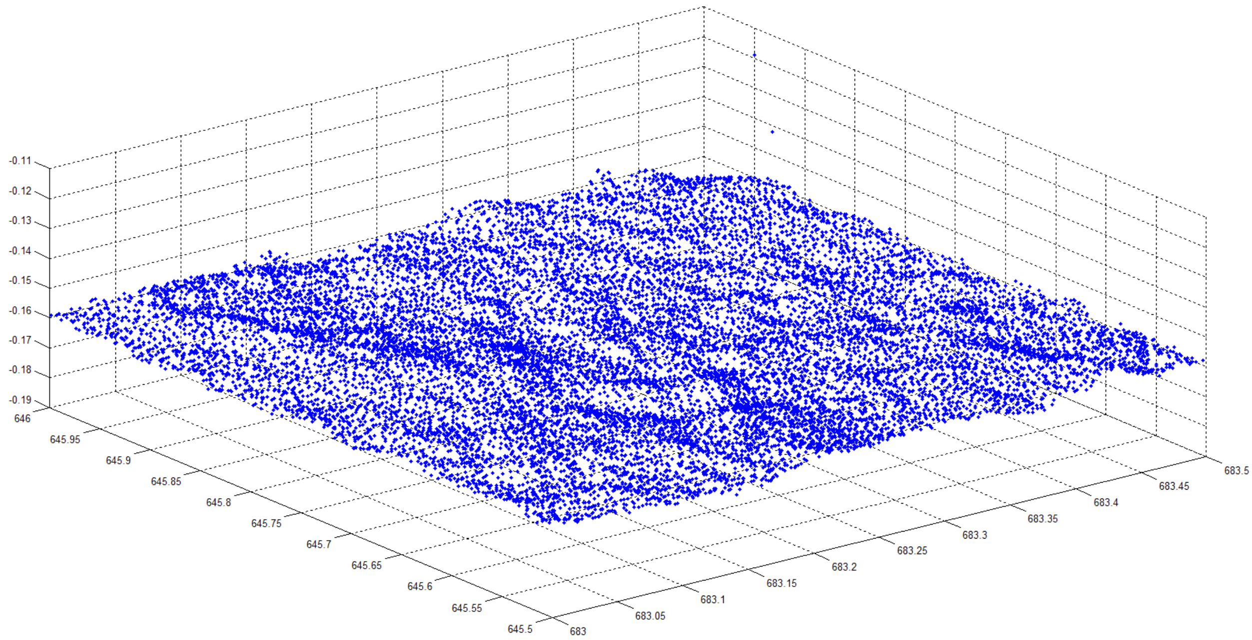


Figure 8: point cloud for the cell with centre (683.25;445.75) at 06-11-2012 at the shoal of Viane.

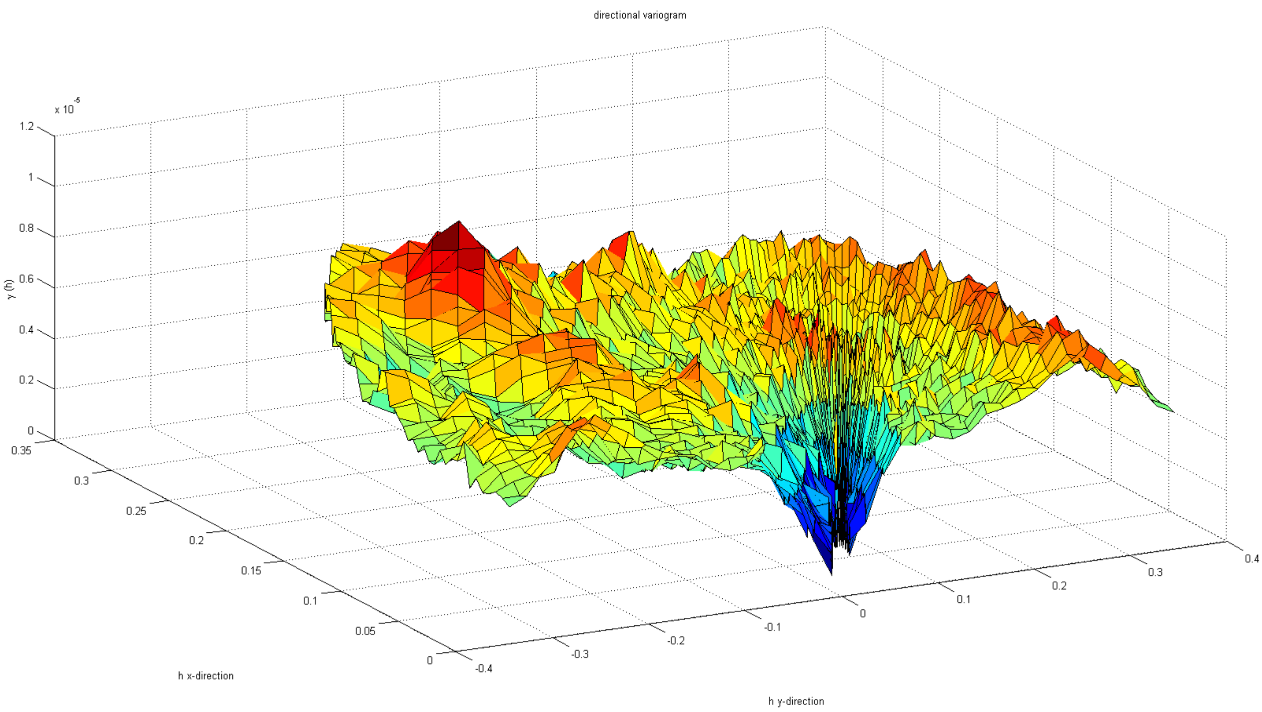


Figure 9: directional variogram for the cell with centre (683.25;445.75) at 06-11-2012 at the shoal of Viane.

Computational algorithm

Using Matlab the (x,y) coordinates of the about 80 million points of the datasets are multiplied by two and then rounded to zero decimals, thus half metres, in order to translate each point to the nearest cell centre. The data is then sorted on ascending x-coordinate and within the x-coordinate on y-coordinate. The same has been done using $1m^2$ and $0.01m^2$ grids. The rounding is done to the nearest integer, because the cell characterised by (x,y) has its centre at (x,y). If the point would be rounded up or down, its coordinates would correspond to one of the corners. All grid cells are now characterised by an (x,y) coordinate in half metres, and each cell contains a number of heights, still accurate to millimetres. A rectangular grid is then made ranging from (1,1) to (max(x),max(y)). This is done so that the coordinates of different datasets are easily placed on top of each other, as their respective coordinate systems undergo the same spatial translation. For the shoal of Viane this resulted in an 1372X1626 grid, containing 2.23 million cells. The southern 200m however contain no data as the origin lies in the middle of the Eastern Scheldt, they are cut away for plotting purposes, resulting in a grid of 972*1626, with 1.58 million cells. The grid is so large because the data covers a lobe from the south west to north east (figure 3), and as matrices are inherently rectangular the grid has to be rectangular. Therefore north western and south eastern triangles, the green marked areas in figure 3a, without data have been added. For each cell the amount of points in the cell, as well as the standard deviation and mean are calculated. Each point more than three standard deviations away from the mean is then discarded, as these points will most likely be reflections of the reflectors, which are mirrored below the ground surface. The mean, standard deviation and number of points per cell are then stored in their respective grid matrices.

This grid has to be constructed, because the raw data points do not always exactly overlap and there are just too many of them to be able to do any quick calculations. By grouping them into the same, relatively coarse grid for each epoch, the different cells can be evaluated with respect to their counterparts in subsequent epochs. Another reason is that the data is a discrete point cloud, which has to be interpolated between the points in order to make a surface, on which two dimensional calculations can be performed for height differences and depositional velocities.

The different scans were then stretched to the same size, meaning rows and columns containing zeros would be added at the high ends of the data if necessary, keeping the origin at (1,1). The zeros are subsequently turned into Not-a-Number (NaN), because that way they can be made not show up on the plot and thereby giving a one colour pane across the grid. This also ensures that when subtracting the two datasets only points that contain a height in both scans were plotted in the difference map. If this would not be done, points would show up as having changed since the previous scan and a difference would be reported which most likely is untrue. The only instance that the reported difference would be correct is if the initial height was exactly zero, these values are collateral for the NaN procedure, but as it is highly unlikely a cell would contain a height exactly zero, the number of erroneously deleted cells is small. The statistical deformation test is performed according to equation (7), after which all maps are plotted. The logarithmic colour map in the point density maps (figure 4) was created using a program written by John Barber.

The variograms are created using a code by Wolfgang Schwanghart. For each cell containing more than 200 points a variogram is made. This threshold has two advantages, a mathematical one namely that a variogram with less than 150 points is to sparse in data, so no good correlations can be found (Traut 2007); the other one being of a geometrical nature: if the point density is too low, the tidal ripples will not be sampled sufficiently, thus the ripple parameters are difficult to find. The primary downside of the variogram calculation is that a redundancy matrix containing the spatial distance between two points has to be made. For N points this results in N*(N-1) matrix entries. This number goes up quadratically with the amount of points and above N = 25.000 or so the computer RAM maxes out.

To reduce computation times every cell containing more than 3000 values is randomly sub sampled. This results however in the stark blue circles around the scanning positions. Due to the mechanics of the scanner, see above, the point density will be highest the closest to the scanner. When sub sampling the cells with lots of points the trend in the data gets destroyed, as there will not be points exactly on ripple highs and lows, thus the trend one wants to find in the variogram disappears and the variogram looks more like the theoretical variogram (figure 6), without any trend. Cells with more than 20.000 points have therefore been disregarded, because the sub sampling will destroy any desired parameters.

Plots can easily be made using the procedure above. The statistical difference test is not used, since the variogram itself is already a statistical function.

Results

First in this section general remarks will be made about the erosion and deposition of sand and the direction of movement of the sand on the shoal. Then the results of the variogram method will be presented, followed by the comparison to the AHN. Finally, the results of the different scanner elevations and settings will be presented.

Figure 10a shows that most of the high, north eastern, area of the shoal of Viane underwent little to no height change between the two epochs, whilst in the low area a significant amount of material was removed on the western side and deposited on the eastern side. This indicates that material moved from close to the land to the water. This means that the tidal waves remove material from the shoal. The Oosterscheldekering storm surge barrier at the same time prevents material from entering the Eastern Scheldt with ease, thus resulting in a net sand loss. This is the sand hunger which the oyster reefs should prevent.

The oyster reef apparently does a proper job at keeping the sand at the shoal, as most of the area behind the reef underwent little change or even increased in height. The total change in sand volume on the shoal of Viane between 6 November 2012 and 24 October 2013 amounts to an erosion of $912m^3$ of sand. As can be seen from figure 10a, the area not protected by the oyster-reefs has lost quite a lot of material, even though the oyster-reefs keep a hold on to $150m^3$ of sand. This means that the oyster-reefs can actually prevent the sand hunger effectively and even enable deposition to take place.

The same can be concluded from the smaller shoal of Val (figure 10b), where the difference is even more pronounced: a total gain of sand on the shoal of $7753m^2$, of which $4383m^3$ occurred behind the reef. As can be seen in figure 10b, a lot of material also accumulates in front of the reef, on its SW side. This is remarkable, as the retreating water during low tide usually takes sand with it, as can be concluded from the shoal of Viane. At the shoal of Val however, sand accumulates in front of the reef. A possible explanation is that the gullies present act as a water highway, creating low velocities on the flats in between them. These low velocities perpendicular to the main direction of flow deposit sand at the flats.

There is however an area directly behind the reef where there is erosion. The reef was either stagnant or gained height (figure 10), whilst it is fringed by an area void of data, which indicates the presence of water. This happened because when the low tide sets in the water from behind the reef flows almost perpendicular to the reef towards the main channel, SE for Viane, SW for Val. The flow however is interrupted by the reef, as it is supposed to. The water thus flows around the reef, creating a trench of erosion at the reef, where all water collects and flows towards the sides of the reef. At Val the water has to flow towards the NW in order to flow out of the catch created by connecting the reef to the dike. The last bit of water stays in a puddle, the white area around the reef. Because of the afore mentioned problems with scanning in water, this area remains void of data.

Almost the entire area underwent deformation, as can be seen in figure 11, with the reef at Viane and the dike at Val being the stable exceptions. This means that even with the Oosterscheldekering preventing sand from entering the Eastern Scheldt, sand still accumulates in certain areas.

Variogram

Two variograms have been made for single cells of 20×20cm, with their centres at (320;241) and (321.6;244) respectively, for the data from Val at 27 February 2012. Both locations are marked in figure 21b, in order to compare the results. For the first cell, the riffle direction is calculated as 45°, with a wavelength of 5.28cm and an amplitude of 6.1mm (figure 12a). The direction is found by searching for the radial with the smallest standard deviation, this is the distance perpendicular to the ripple crest; this is the purple line in figure 12a. 90° are added to this line, and when going past 180°, 180° will be subtracted, as that is the same line orientation. On this radial, the red line in figure 12a, the maximum value is sought for. After this maximum, the first minimum gives the wavelength, as discussed in *Methodology* – variogram. The amplitude finally is the square root of the maximum value within the directional variogram.

The second cell gave Not-a-Number as result, meaning there is no trend in the data. This means that no hole-effect can be found, thus there is no periodicity in the data. When looking at figure 12b, one can conclude that this is indeed the case. The variogram does go up and down a bit, but over all the variogram increases with distance. Comparing these results to the actual point cloud shown in figure 21b, it can be concluded that the variograms give the correct results, though for the (321.6,244) cell the data looks like it has a bit of a trend in it, but that is obscured by the large amount of points contained in that cell.

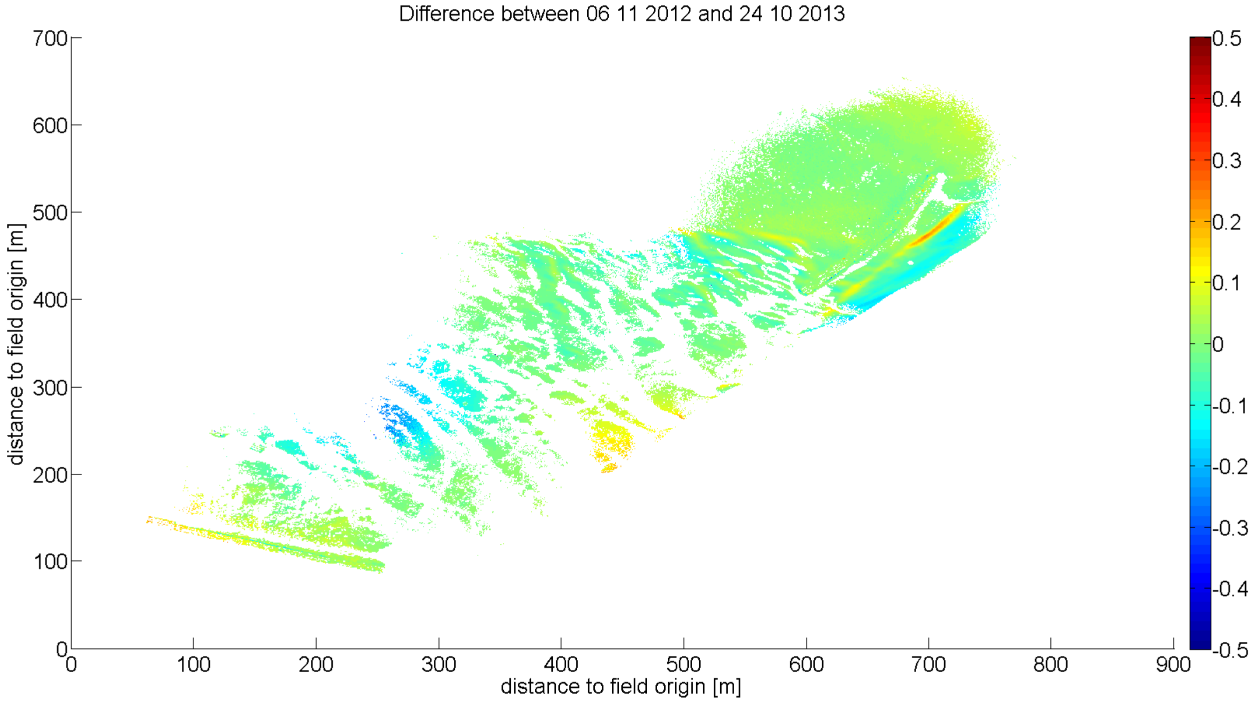
AHN

For the shoal at Val AHN (*Actueel Hoogtebestand Nederland, Current Height database The Netherlands*) data as measured in 2007 is available (figure 13). The shoal of Viane is not scanned, probably due to high water. The height difference between the AHN and the first available laser scan is still within half a metre erosion or deposition (figure 14). Clearly visible is the oyster reef, as it had not been built in 2007. Most of the area behind the reef has undergone erosion as well. This is because the reef has been built in 2011, so there is four years of erosion and one year of deposition between these scans. Comparing figure 14 to figure 10b confirms this, as figure 10b clearly shows mainly deposition behind the reef.

Scanner elevation and settings

Figure 15 shows an interesting image. It is the difference in measured height between the high-speed scan and the long-range one on the same spot, taken directly after one another without moving the scanner. As can be seen, towards the east the long-range scan becomes gradually lower than the high-speed scan, whilst to the west the opposite happens. This indicates that the two scans are tilted relative to each other. Since the change is gradual, it can be concluded that the actual difference, without the tilt, is almost zero on the mudflat. The largest differences between the two scans can be seen on the dike, where each cell differs seemingly randomly between scans. Most of the observed differences on the dike are positive, meaning the long-range scan observed them as being higher than in the high-speed scan. In the long-range scan the points appears to be higher, thus further away from the scanner. This means that the long-range setting got a longer travel time of the pulse than the high-speed setting. Since no benchmark height has been set with different means of observation, no conclusions can be made as to which setting most accurately measures heights.

In figure 16 the difference between placing the scanner at 5.5m altitude on top of the dike and at -1m at the bottom of the dike using the high-speed setting is shown. The toe of the slope shows little difference in height, whilst the relatively high areas are consistently higher in the scan made from the bottom of the dike. This is nicely shown in the fallow land down the centre of the scan, the land was freshly ploughed and the top of the furrows is red, whilst the bottom is green. This means that the travel time of the pulse is not sufficiently decreased between the two scans. Since the scanner is closer to the respective points, the travel time of the laser pulse should go down accordingly. It does go down, otherwise a difference of around 7 metre would be shown, but it does not go down enough, since between 7 and 10 cm difference is still present. Again, no benchmark height is available, so no conclusions can be drawn about which height is better to scan from.



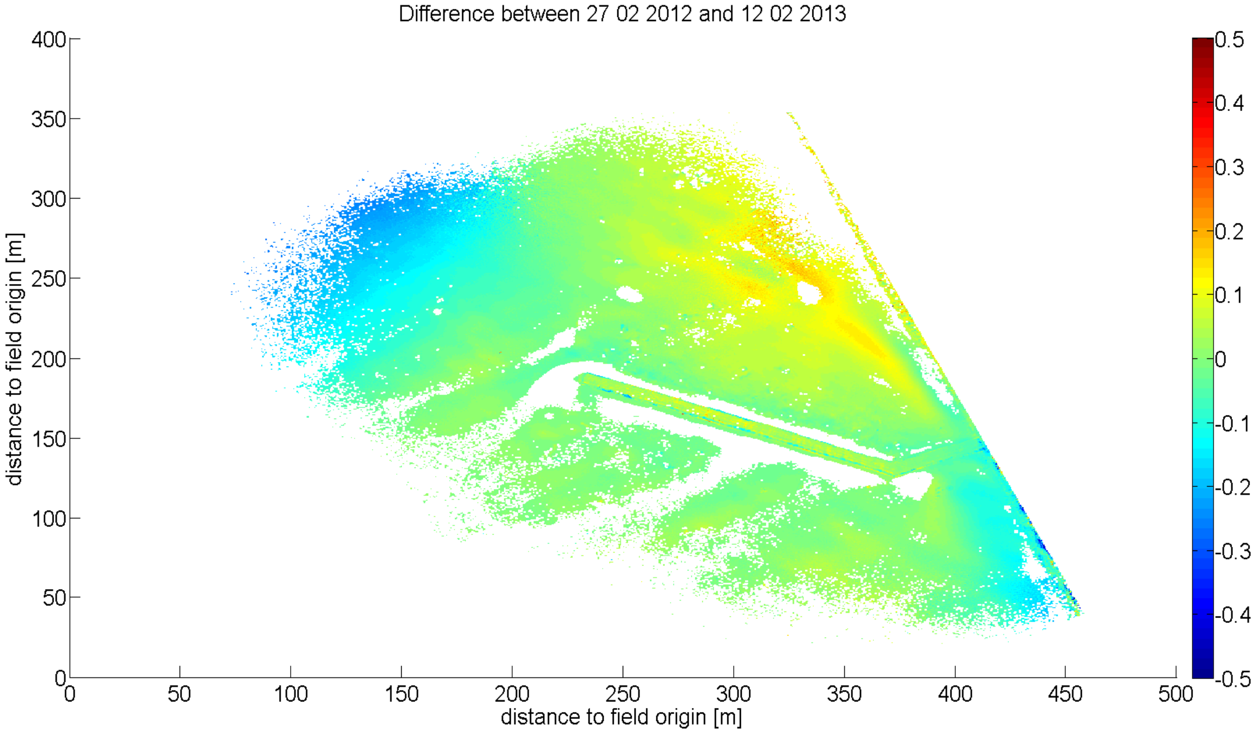
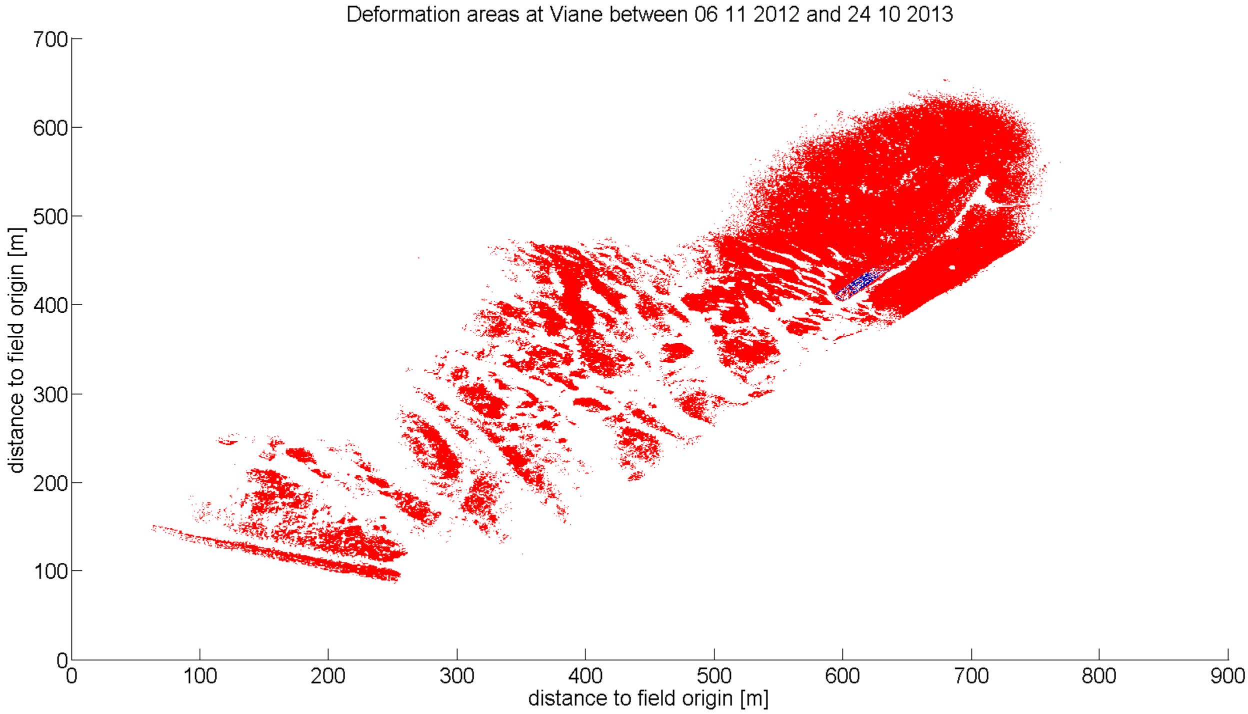


Figure 10a,b: height difference map for Val and Viane for the period of one year . The colour scale ranges between -0.5m to 0.5m relative to NAP.



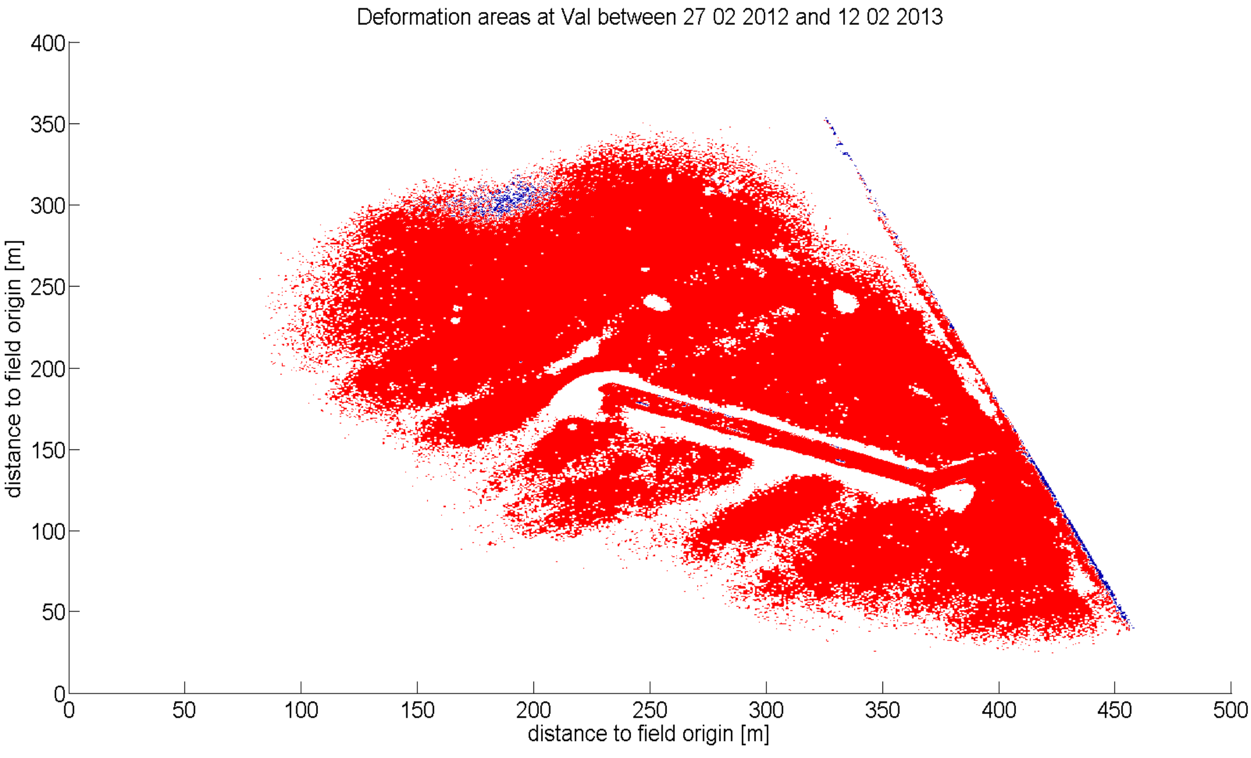


Figure 11a,b: erosion and deposition areas on the shoals of Viane and Val during a year. Red indicates deformation according to the H_0-hypothesis, blue indicates no change.

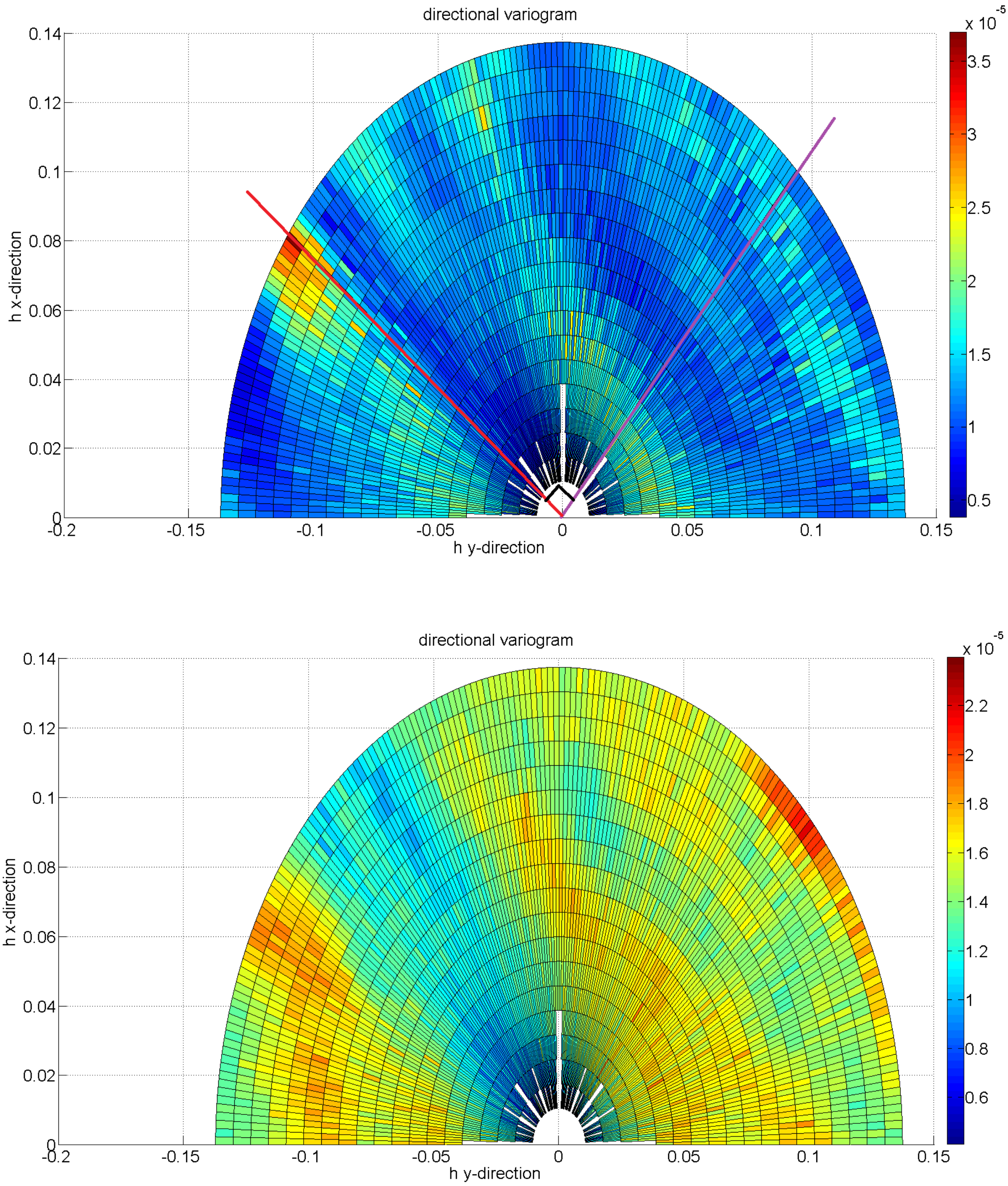


Figure 12a,b: directional variogram of the cell with centre (320;241) and (321.6;244) with 20cm edges at Val 27-02-2012. Colours denote the average variability in the cell for that direction on that distance. The circle runs from north on the lower left, via east on top to the south.

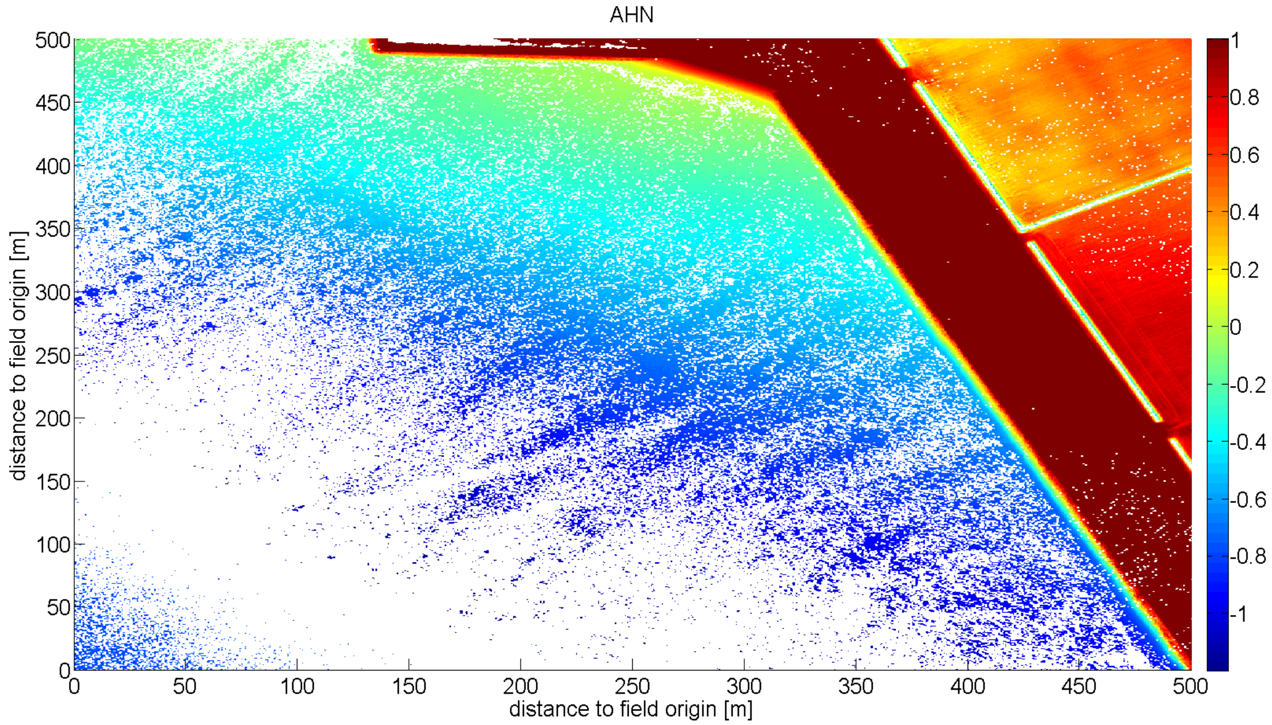


Figure 13: AHN heights of the shoal of Val. Heights acquired in 2007 by Rijkswaterstaat using airborne laser scanning. The colour scale runs from -1.2 to 1m.

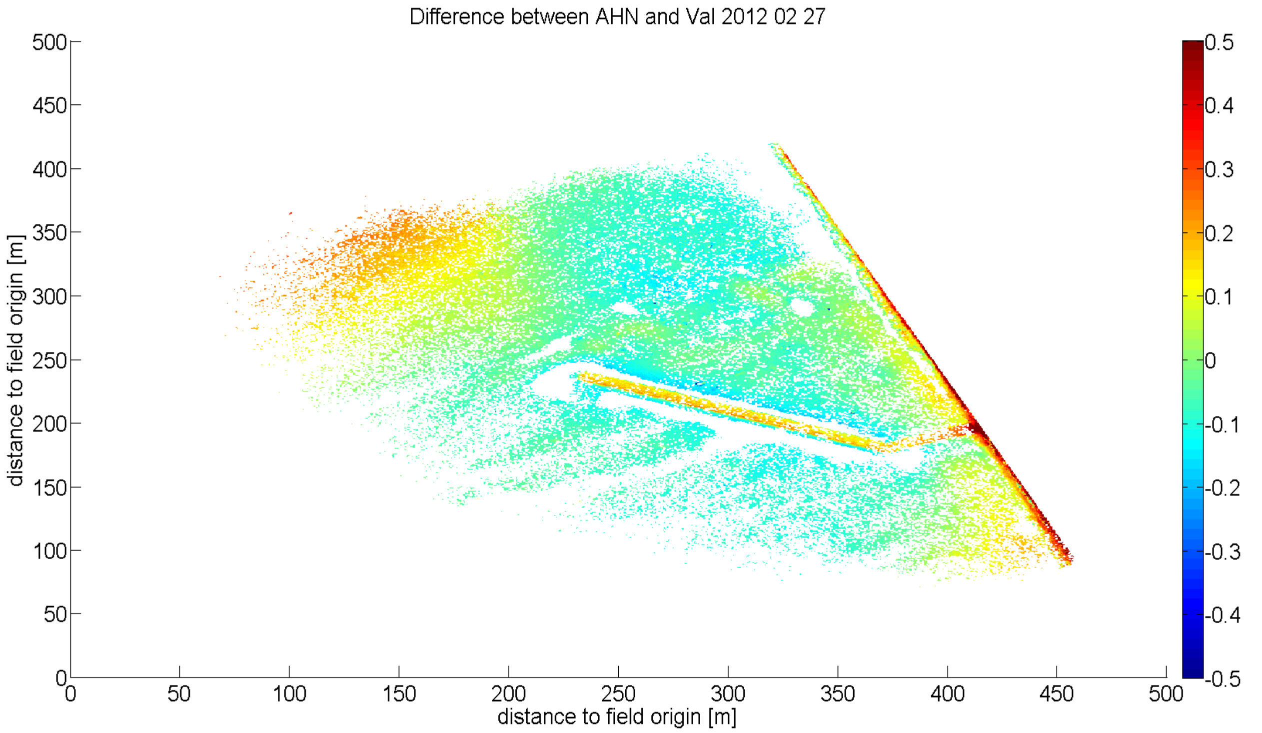


Figure 14: difference between the AHN of 2007 and the laser scan at 27-02-2012 at the shoal of Val. The colour scale runs from -0.5 to 0.5m.

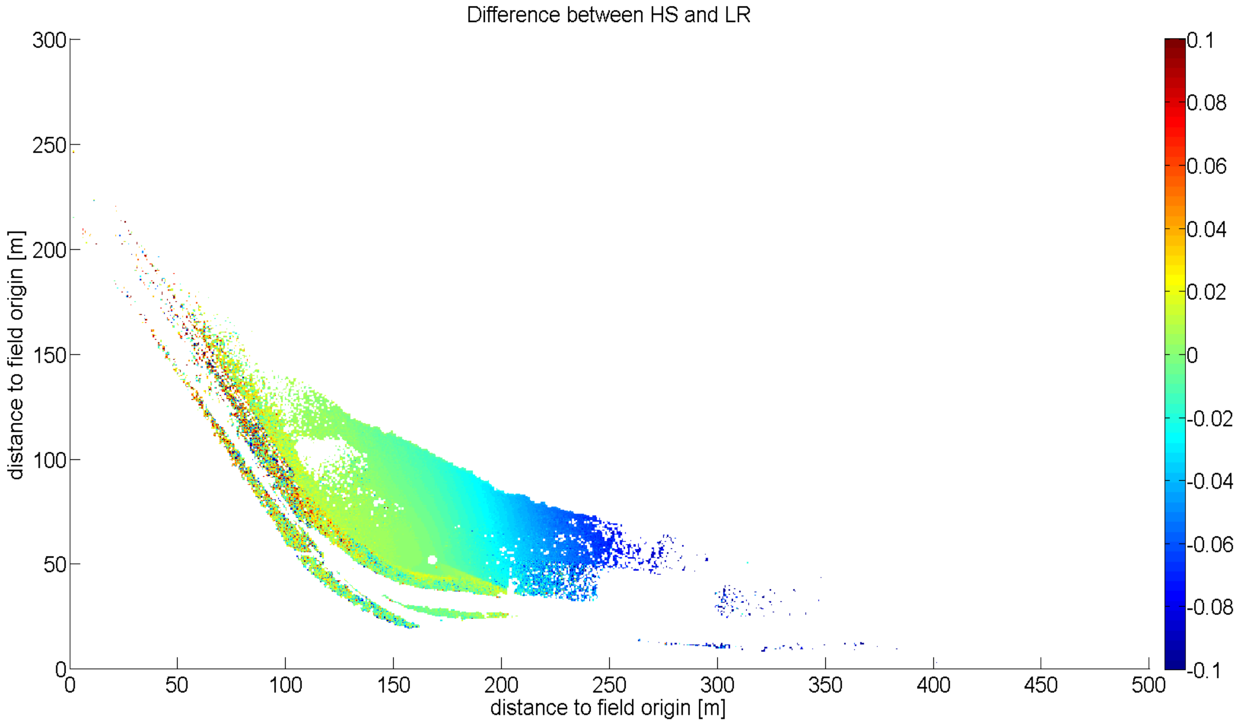


Figure 15: difference in height between a high-speed scan and a long-range one at the same location, scanned directly after one another. The colour scale runs from -10 to 10cm.

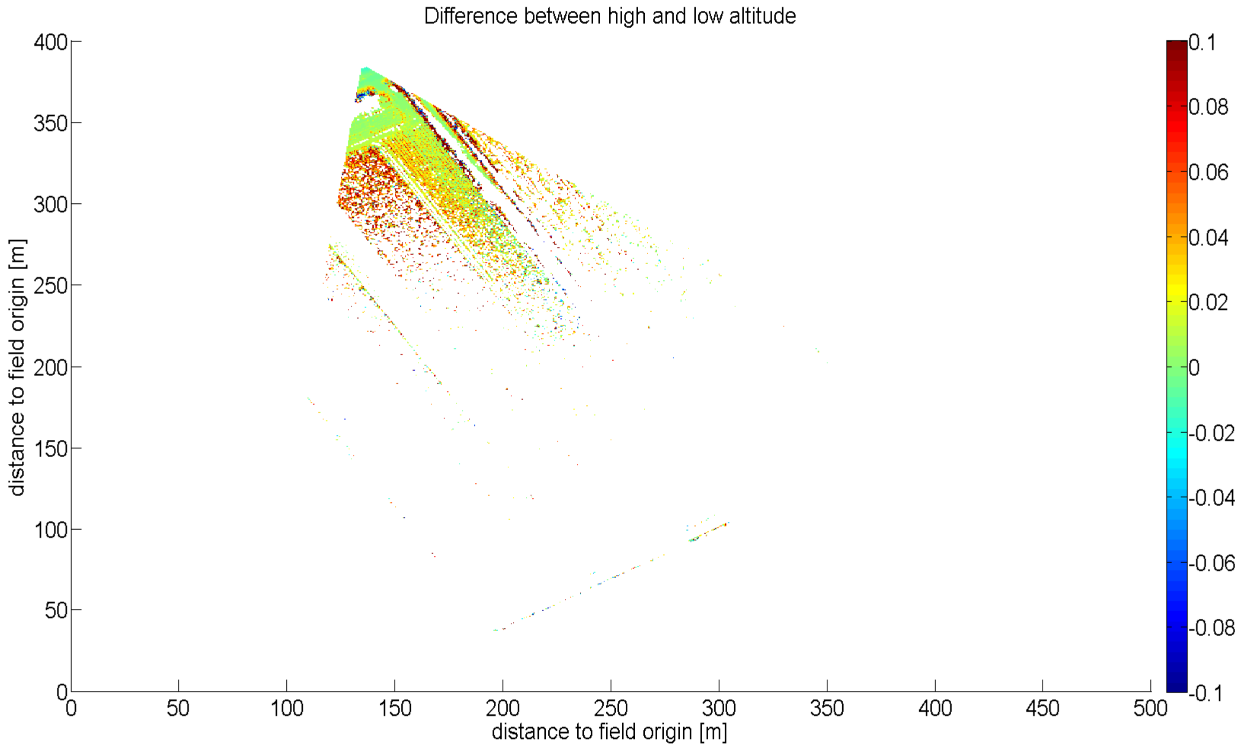


Figure 16: difference in height between the scanner at high and low altitude, utilising the high-speed mode. The colour scale runs from -10 to 10cm.

Discussion

In this sections first some remarks will be made about the grid size, after which the variogram will be discussed. A few remarks will be made about the required computational time and the scanner elevation and settings will be discussed, as well as the movement of sand on the shoals. Finally recommendations for further research are given.

Grid size

The grid size has a large influence on the resolution of the data. A high height resolution can be obtained with a fine grid. However, too fine a grid becomes computationally cumbersome. Since the RAM on the machine processing this data can contain $1.8*10^9$ points, grids containing more than that amount of points cannot be processed, unless more RAM is installed. Grids of $1m^2$, $0.25m^2$ and $0.01m^2$ have been produced. The full scale figures in the above section have been made with the $0.25m^2$ grid, as the data in the $0.01m^2$ grid for the shoal of Viane becomes so sparse that the area appears almost void of data, except directly around the scanner. A qualitative comparison between the two grids is given below using a transect through the shoal of Viane at y = 500m (figure 18, the line AB in figure 3a) and the height map of the shoal of Val at 27 February 2012. As can be seen the continuous line, being the transect based on the 10×10cm grid, is much more capricious than the dashed line, which is based on the 50×50cm grid. There are two main reasons for this: the line plotted from the 50×50cm grid has a point density five times less than the 10×10cm grid, but the transects are plotted on the same scale. This means that the 50cm line cannot have the fine structure of the 10cm line. The other reason is that these points do not represent an infinitesimally small line, but a 10 or 50cm wide band, over which the heights have been averaged per cell. The 50×50cm grid has 25 times more area per cell compared to the 10×10cm, thus the height will have less extravagant behaviour, due to it being smoothed over 25 times more data.

As can be clearly seen from figure 20 the points are clustered in a circle around the scanner locations, which is as expected, given the method of scanning. The scanner cannot scan directly underneath itself, in a circle of 1.5m diameter. The scanner is moved approximately 100m from its neighbouring scan position, well within the 160m range of the scanner. However, after 30m the point cloud becomes less dense, to vanish completely after 50m. This results in a set that has a point sparsity at further distances from the scanner, requiring a larger grid, such as $0.25m^2$, in order to still give a continuous map of the area. This however also fills in gaps which should not necessarily be filled in, such as the water surrounding the oyster-reef and the SW-NE trending gully directly north of the reef at Val. These clearly show on the $0.01m^2$ plot, but are almost completely filled in on the $0.25m^2$ grid. The scanning positions result in only one or two empty cells in the larger grid, making them all but invisible and only recognisable as scanner position when viewed along with the smaller grid.

Using CloudCompare© the entire dataset can be visualised. It shows (figure 21) that the point cloud density is much higher close to the scanner, whilst at 40m it is too thin to make a full $0.01m^2$ grid. This means that close to the scanner small bed forms can be evaluated, whilst outside the 30m range only bed forms that are larger than 50cm can be identified. This is the geometric reason for the 200 point limit set for calculating a variogram. It can furthermore be concluded that one lone point on the fringes of the data is representative for $0.25m^2$ or for only $0.01m^2$ in the finer grid. This means that the in the coarse grid outlying areas are coloured covering a fair amount of area, whilst in the finer grid the degradation is much more pronounced.

Any coarser grid than $0.25m^2$ would not be of use. The resolution becomes even lower, whilst improving calculation speeds. The calculation speed of the $0.25m^2$ grid however is already fast enough for the 0.48km^2 shoal of Viane. The area additionally assigned heights using a larger grid is small, as there are not many points on the fringes of the data. Another downside is that where there should be no points, such as on the water areas around the oyster reefs, heights will be averaged and stretched across that area, making them invisible.

Variogram

The variogram is proving difficult to make. Creating a variogram of a single cell is no problem, including obtaining the three ripple parameters. However all the points in the cell were cut out of the whole dataset first, using an if-statement for each entry; if the point lies in the cell keep it, otherwise discard it. Doing this for all of the cells which contain more than 200 points would take a tremendous amount of time, as it takes about two minutes to go through the entire dataset. This was the first problem encountered when trying to make the grid, ultimately solved by sorting and rounding the data, this can however not be done for the variogram, as the actual location of the point is required. Sorting the full data table and then storing the x-coordinates of the left and right edges of the cell and within that range searching for all points within the y-axis solved the problem.

Computational time

As mentioned in the introduction, one of the biggest problems with laser scanning is the amount of data that is obtained. Huge amounts of data require huge amounts of processing time, as each operation has to be done millions of times. For simple operations like plus and minus this is not a problem. However, when calculating the variograms computational times blow up. Making the grid takes five minutes for 80M points on 600x800 cells, which includes storing the number of points per cell and the standard deviations, the latter taking the most time. The variogram however, took 10.4 hours for Val 2012 and 6.6 for Val 2013. The reason variograms require such large computational times is because it entails several matrix operations on large matrices, see above in *computational algorithm*. For all four datasets the program took 45 hours non-stop in order to finish. Reading and processing the points-per-cell matrix taxes between half a second and a second, reading and sorting the point table takes around two minutes; the variogram calculation for each cell takes up to 14.5 hours (table 1).

Computational times could be decreased by using a different type of code, such as C, which is more suited to this kind of operations. For instance within the script *Variotest* (appendix B) , the entire data table has to be loaded in order to find the required coordinates, whilst in C a dataset can be negated without storing it completely in the memory first. It reads every line and checks the statements associated with the data to be read and executes them directly.

Scanner elevation and settings

The placement of the scanner at high or low altitude does not matter significantly in terms of resolution. The resolution can be inferred from the amount of points per cell, since more points in a cell means a denser clustering of measurements. Figure 22 shows this difference. The area close to the scanner has a lot more points for the low scan than for the high scan, which is understandable when considering the scanning principle: closer to the scanner the point density is higher. The high position is approximately 6.5m higher and 4m to the NW, thus 7.6m away. The blue area in the NW of figure 22 is an area behind a small bush. This bush gives an occlusion, the white area, and the high positioned scanner has a smaller area of occlusion, since the angle to the bush is much smaller. For the majority of the scan on the land it does not matter where one positions the scanner, as the amount of points per cell is approximately the same for both positions.

Without a benchmark height, not much can be said as to which settings are better to use regarding accuracy. From figure 24 it can, however, be concluded that the effective range of the scanner using the high-speed setting is about 180m, whilst when utilising the long-range setting this increases to about 260m. The point cloud in the SE of the land is noticeably denser in the plot of the long-range than than of the high-speed. Furthermore it can be concluded that the tilt angle is the limiting factor on flat surfaces, like the ones discussed in this paper. When the angle between the laser pulse and the surface is not so small, the long-range setting can scan a lot further with a reasonable point density, for the dike at 500m is scanned almost completely and at 600m around 50% of the dike is still scanned (figure 24b).

The elevation of the scanner is very important for height resolution, as is illustrated in figures 23 and 24b. The scan at the higher elevation (figure 24b) clearly achieves a much higher point density, as the grid field coverage is kept connected much further out. This is because of the increase in tilt angle the scanner can make.

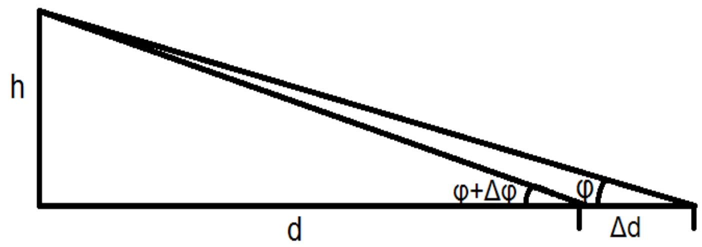


Figure 17: schematic drawing of the scanner during scanning

Consider figure 17, where *h* denotes the scanner elevation, *d* the scanning distance, *φ* the angle with respect to the horizontal and *Δd* and *Δφ* increments in *d* and *φ* between steps of the scanner. The scanner has an angular increment of 0.0024°. The angle *φ* is now given by:

$$\tan(\phi)=\frac{h}{d} \quad (10)$$

For the next point to be scanned the laser beam is shifted by *Δφ*, resulting in a shift in position of *Δd*:

$$\tan(\phi+\Delta\phi)=\frac{h}{d-\Delta d}\approx\frac{h(1+\frac{\Delta d}{d})}{d} \quad (11)$$

This relation is valid only if *Δd* << *d*. Using now the trigonometric relation for the tangent:

$$\tan(\phi+\Delta\phi)=\frac{\tan(\phi)+\tan(\Delta\phi)}{1-\tan(\phi)\tan(\Delta\phi)}\approx\tan(\phi)+\tan(\Delta\phi) \qquad 45° >> \varphi >> \Delta\varphi$$

results in:

$$\tan(\phi+\Delta\phi)-\tan(\phi)=\frac{h\Delta d}{d^2}\approx\tan(\Delta\phi)$$

$$\Delta d \approx \frac{d^2}{h} \tan(\Delta \phi) \quad (12)$$

From equation (12) it is evident that when the scanner is placed at a higher elevation, the incremental distance between two consecutive scan points goes down linearly. Also shown in this equation is that the incremental distance goes up quadratically when moving away from the scanner. In the grid this results in a lower point density.

Recommendations for further research

Creating a surface of the data using a Delaunay-triangulation would greatly increase the resolution of the process, as every point is considered separately, instead of binned together in a cell.

If a benchmark height would be measured, the different scanner elevations and settings could be compared with respect to the absolute height and then a statement can be made about which elevation and settings are optimal for the area in question.

Since most of the operations in the computational algorithms are read-and-write, languages like C would be more suited, since they do not require to load all data of a scan into RAM first. For instance the dataset from the AHN is 30GB, which clearly does not fit into the RAM of the system used. Matlab requires this set to be completely loaded into the RAM, before 220MB of points on the shoal of Val can be extracted. Other languages can negate the data without storing it into RAM and directly extract the desired points upon reading.

Process	**Time (s)**
Points-per-cell Val 2012	0.369859
Reading point table Val 2012	114.400116
Variograms Val 2012	37410.531257
Points-per-cell Val 2013	0.431782
Reading point table Val 2013	80.010602
Variograms Val 2013	23811.658084
Points-per-cell Viane 2012	1.1019825
Reading point table Viane 2012	137.20284
Variograms Viane 2012	47976.936523
Points-per-cell Viane 2013	1.047573
Reading point table Viane 2013	145.725783
Variograms Viane 2013	52195.692878
Total time	161875s = 45 hours

Table 1: process time for the variogram calculations

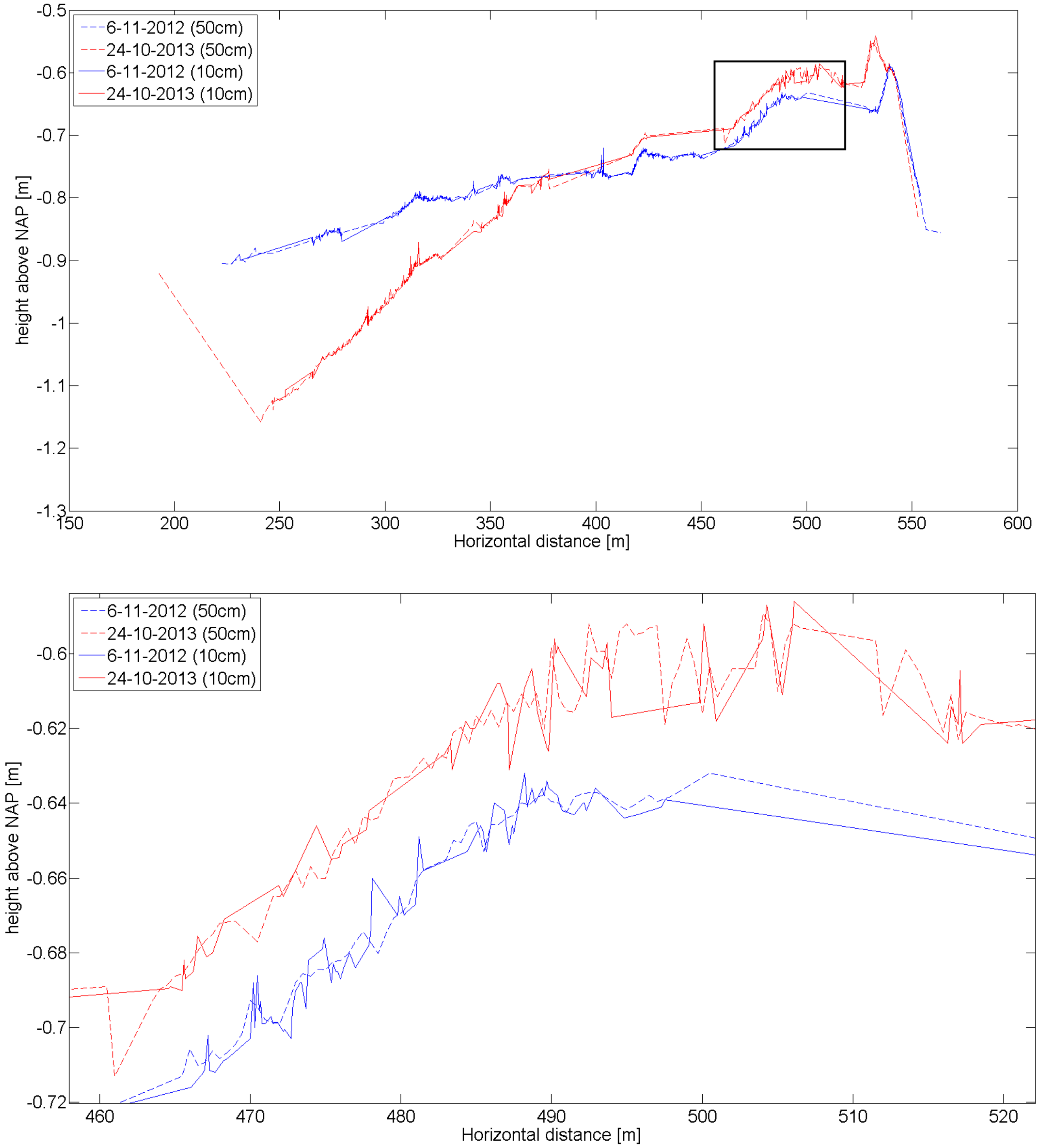


Figure 18a,b: east-west transects at y=500, comparison between the half meter grid (dashed line) and 10cm grid (continuous line). Blue denotes the 2012 set, red 2013. The black box in a is the zoomed section b.

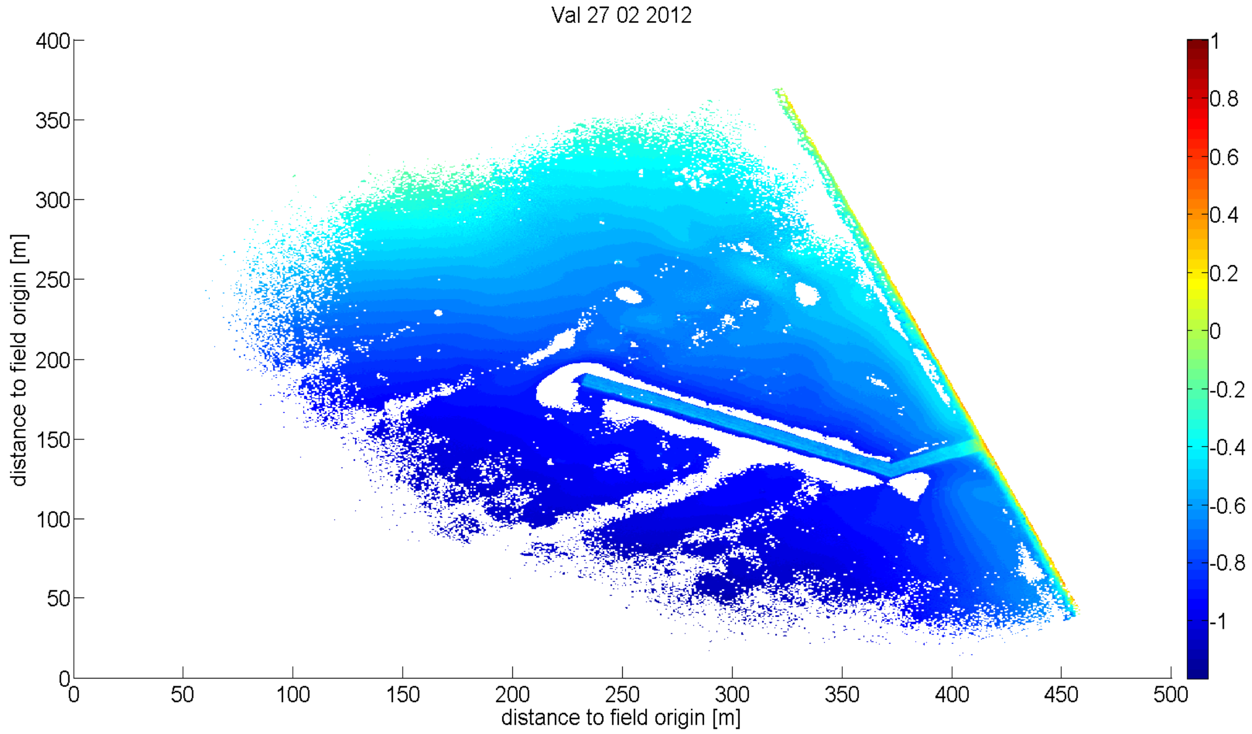


Figure 19: scan of the shoal of Val at 27-02-2012, plotted using a 0.25m² grid. The colour scale runs from -1.2 to 1m.

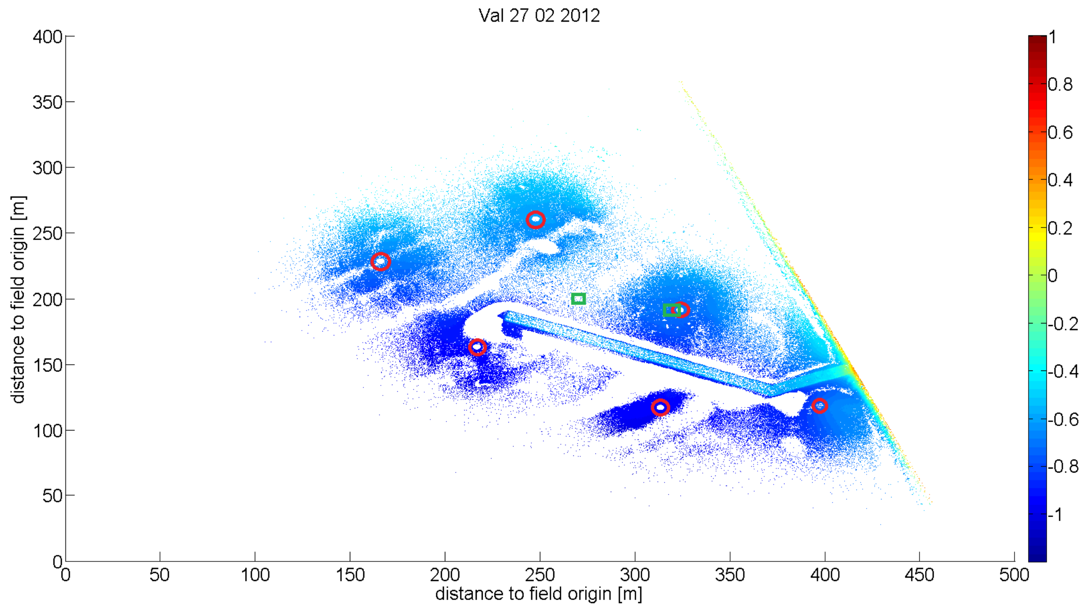


Figure 20: scan of the shoal of Val at 27-02-2012, plotted using a 0.01m² grid. The red circles show the scanner positions, the green squares the locations of figures 20a and b. The colour scale runs from -1.2 to 1m.

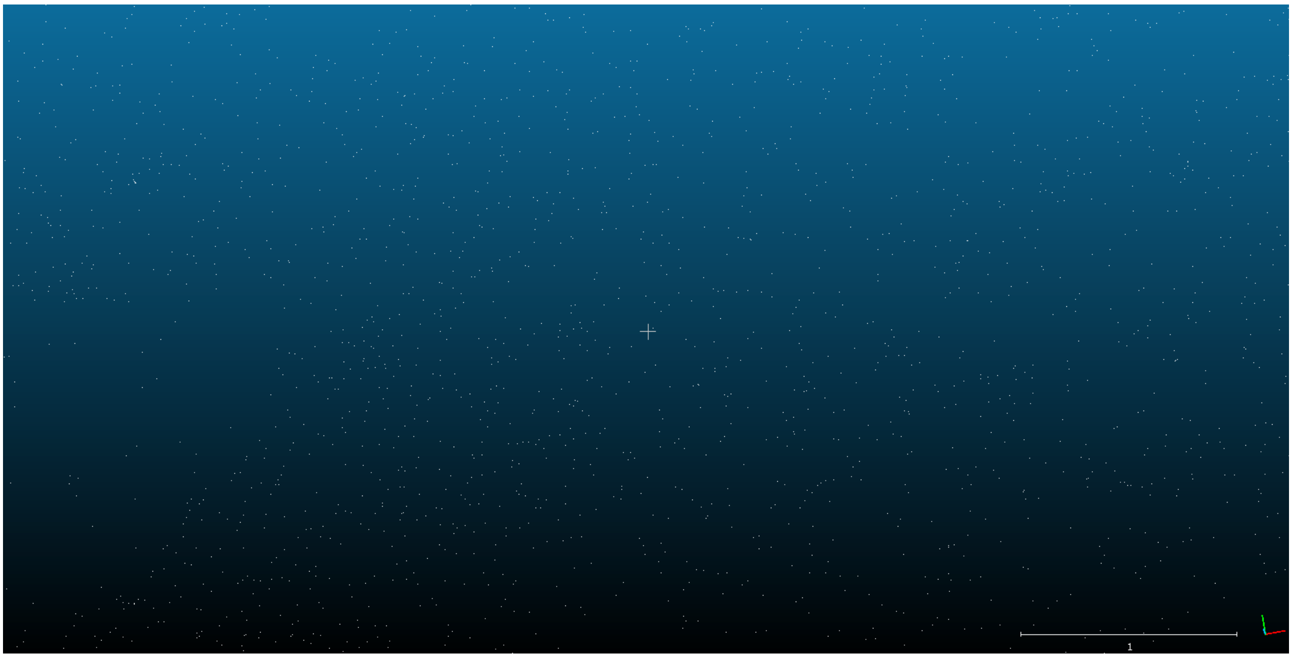


Figure 21a: point cloud behind the oyster-reef at 40m from the nearest scanner position on 27-02-2012 at the shoal of Val. Scale is shown in the lower right, the white line is 1m.

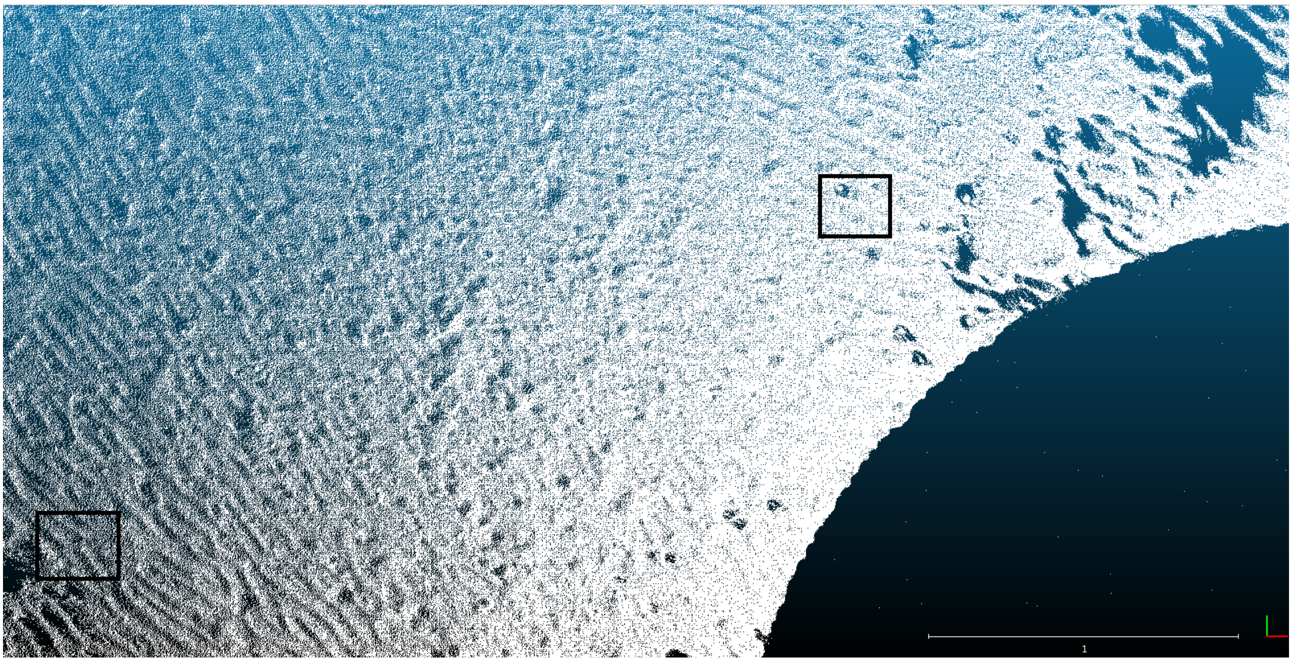


Figure 21b: point cloud behind the oyster-reef, with the scanner in the circle in the lower right corner. on 27-02-2012 at the shoal of Val. Scale is shown in the lower right, the white line is 1m. The black boxes show the areas of the two variograms discussed in the Results – Variogram *section.*

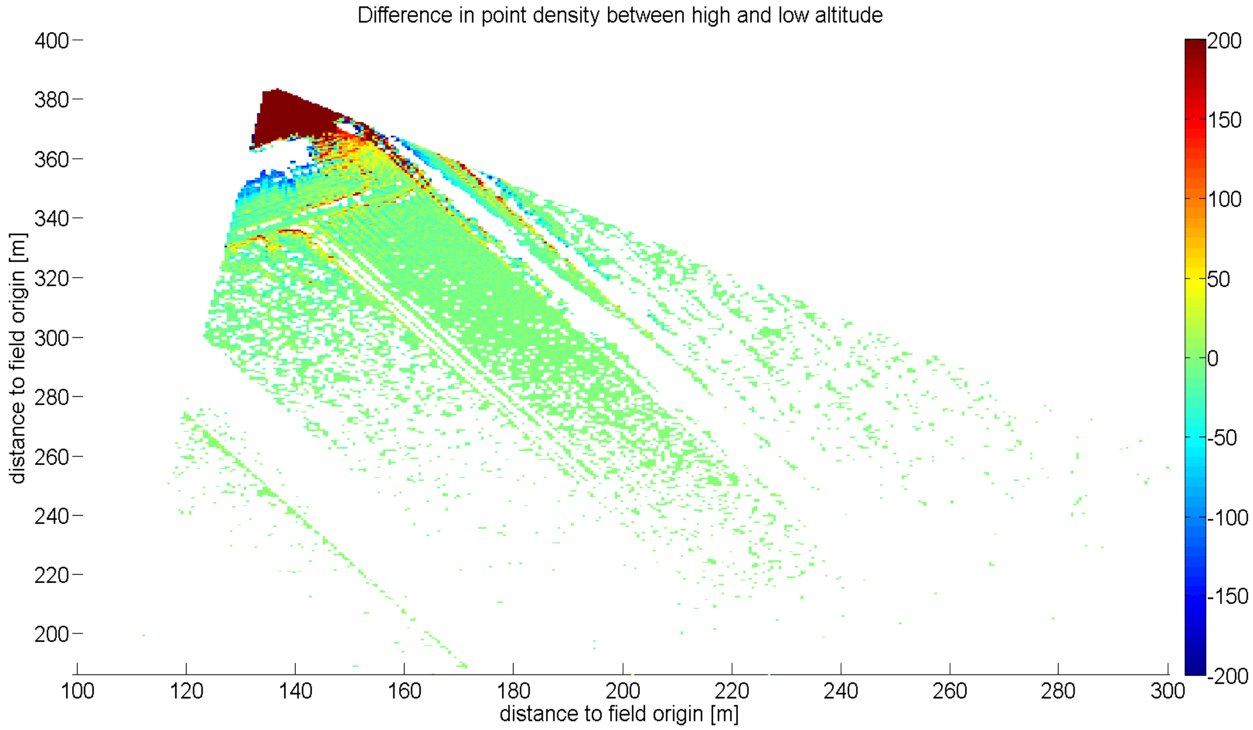


Figure 22: difference in point density between the scan below and on top of the dike. The colour scale runs from -200 to 200.

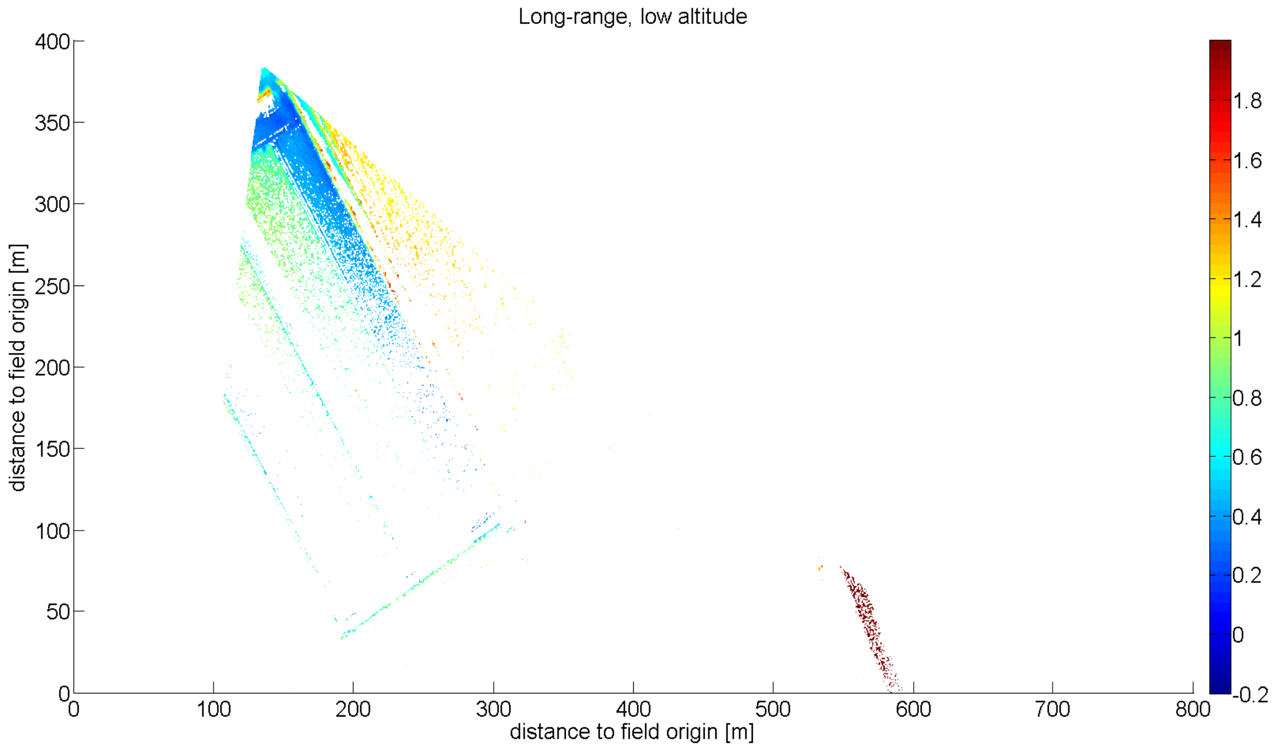


Figure 23: height scan of the fallow land from the bottom of the dike, using the long-range setting. The colour scale runs from -0.2 to 2m.

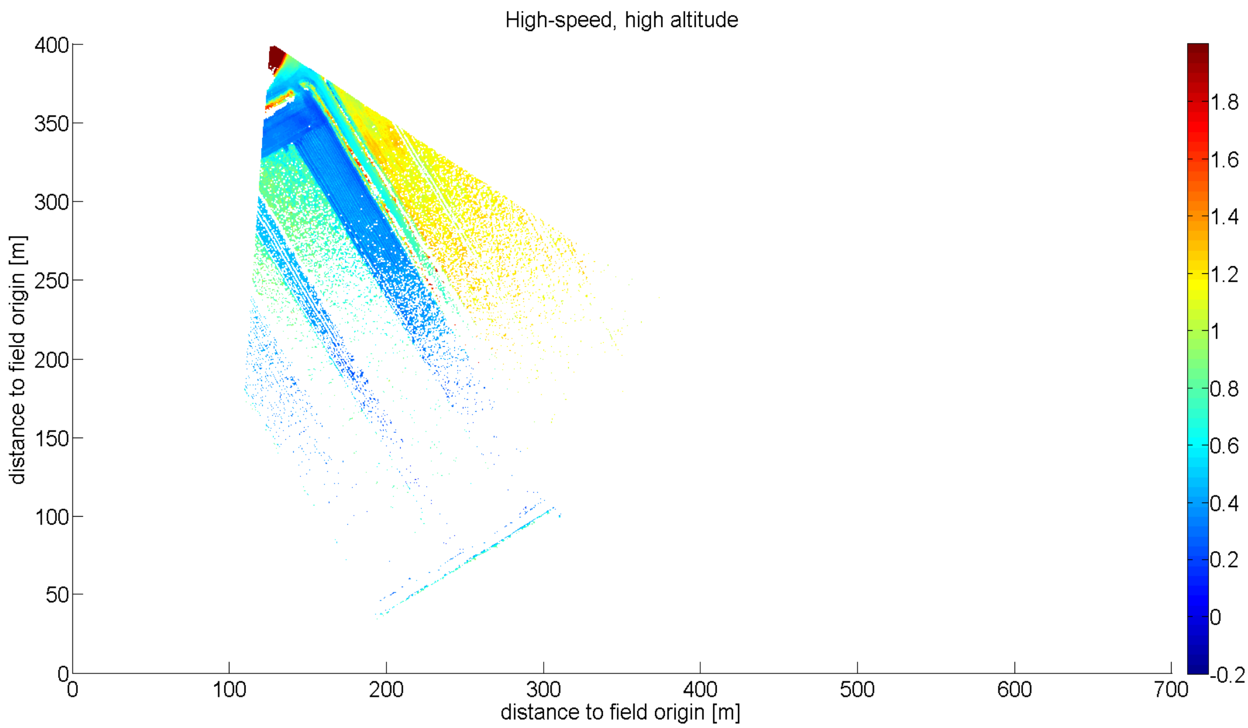


Figure 24a: height scan of the fallow land from the top of the dike, using the high-speed setting. The colour scale runs from -0.2 to 2m.

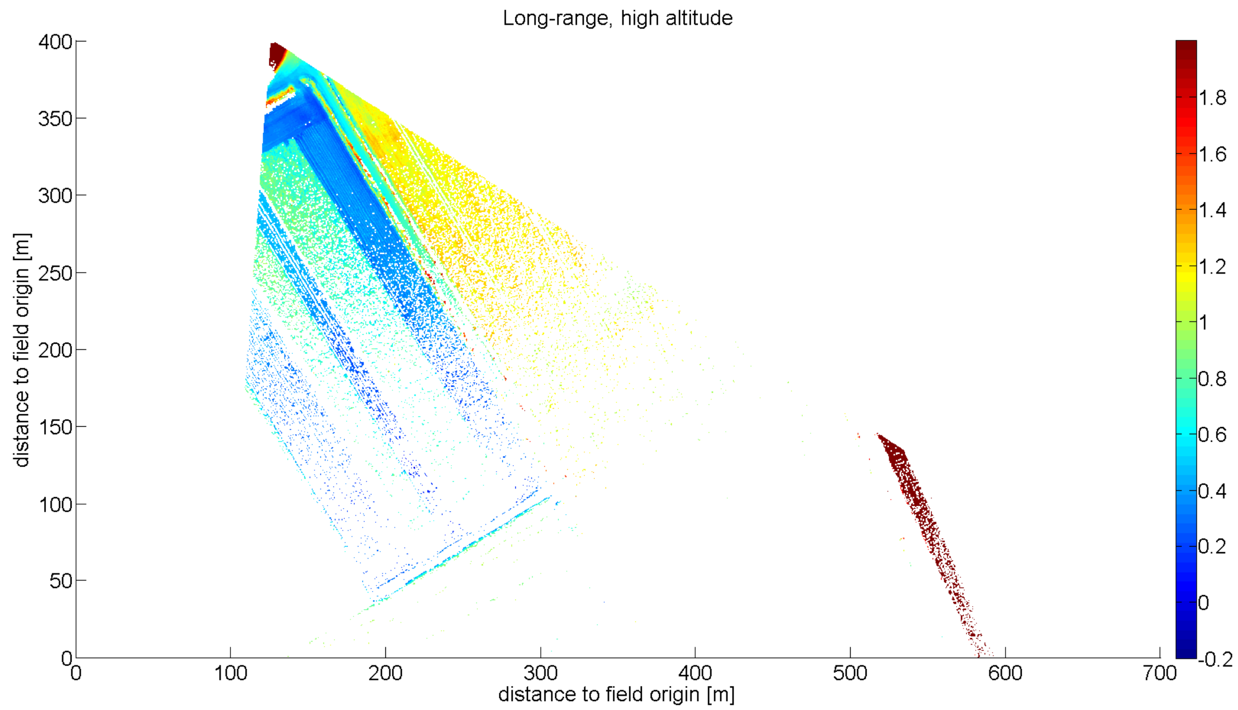


Figure 24b: height scan of the fallow land from the top of the dike, using the long-range setting. The colour scale runs from -0.2 to 2m.

Conclusions

The data obtain by using terrestrial laser scanning is of such high resolution that processing problems occur because of the amount of data. In order to overcome this, a grid has been created in order to reduce the amount of points and to make difference maps with ease when using the same grid for each epoch. This difference map is used in conjunction with a statistical test to determine whether the change is large enough to be classified as a true change, or that the change is due to statistical spread of the data. The variogram method is used to search for small bed forms, tidal ripples in this case.

Six scans have been made on three positions, with long-range and high-speed settings on each position. From these scans it can be concluded that the long-range setting takes three times longer than when using high-speed, but scans only two times the area on a flat location such as tidal flats. The elevation of the scanner however increases the effective range of the scanner linearly with the increase in height.

Answers to the research questions are as follows:

How does one acquire height data using laser scanning?

The method used to obtain the data for this paper works reasonably well. The entire area can be scanned in a short period of time at low tide, with a high accuracy. The only downside is the small range of the scanner, which can be overcome by either making more scans closer to each other or using different settings on the scanner.

From the scans made for this paper it can be concluded that using the long-range setting instead of the high-speed setting increases the range of the scanner on a flat area with 45%, thus doubling the scanned area without loss of resolution. As the long-range scan takes three times as long as the high-speed one, it it not recommended therefore to use the long-range setting on a flat area. When scanning strong relief such as dikes or buildings, the long-range triples the range, thus the area increases ninefold. Placing the scanner at a higher elevation increases the range without downsides.

What are the difficulties in data acquisition?

The main difficulties lie in the range of the scanner and the presence of water in the area. The range problem can be overcome as mentioned before, with the downside that doing more scans requires more time. Whilst this is not necessarily a problem, for instance when scanning sand dunes in the desert, it is a problem when there's only a short time span available, such as the shoals discussed here, as they can only be scanned during low tide.

The problem with scanning wet areas can be overcome by using a green laser in combination with a red laser. The red laser reflects on the water surface, giving the correct position of the laser beam, whilst the green laser refracts at the water surface and reflects at the bottom of the water body, if the water is clear enough. The travel time of the green pulse then gives a distance, with the travel time of the red pulse giving the distance to the water surface.

The reflectors used to tie separate scans in an area together can reflect in the scan, thus having a mirror image below the ground surface in the data. This is easily overcome by calculating the mean and standard deviation for each cell and then discarding all points more than three standard deviations away from the mean.

What kind of signals can be distinguished?

The main morphology of the data can be visually interpreted, with algorithms for the calculation of differences between epochs and the presence and parameters of ripples and dunes in the data.

The directional variogram is a useful tool to determine the direction, amplitude and wavelength of periodic bed forms, tidal ripples in this case. The downsides are that it requires a high enough point density in order to have data on both ripple highs and lows and that the redundancy matrix increases quadratically in size with the amount of points, thus for a high point density per area, smaller grid cells have to be used in order to still be able to calculate the variogram. There are other tools available to distinguish these signals, but these are not discussed in this paper.

How does one determine changes?

Changes are determined by subtracting the matrices containing the desired parameter at both epochs. Where possible a statistical stability test is used to determine whether the measured difference is a real difference or a measurement error. The stability test requires a standard deviation for each cell. For the height difference this can be obtained by calculating the standard deviation of all height measurements in that cell. For the variogram the statistical test is not used, because the variogram already is a measure for statistical correlation. Any parameter calculated from that variogram will be linked intrinsically to the statistics of the point cloud. It has therefore been decided to omit the stability test when calculating temporal differences for the ripple parameters.

How does one process data?

The data is first put into a grid, in order to reduce the amount of points from around 80M to 0.5M, thus a reduction in size of 160 times. This speeds up calculations and when using the same grid for each epoch, it makes sure the cells overlap exactly, thus enabling the making of difference maps. The resolution of $0.25m^2$ cells is sufficient to detect large scale deformations such as erosion and deformation. The smaller bed forms, such as tidal ripples, can be found using a directional variogram, however, this does take a lot of time for large data sets.

Now it its possible to address the main question:

How can terrestrial laser scanning be used detect morphological changes?

TLS is a good way to obtain a large amount of high quality data in a short amount of time. When searching for large scale phenomena such as erosion and deposition, a program with a run time of 20 minutes can be run for the data of two scans of different epochs of the same area. When the finer relief such as tidal ripples are wanted, another program can be used, this takes approximately 10 hours per dataset.

When one wants to scan larger areas, TLS becomes cumbersome, as the effective range of the scanner is only 180m. In this case other techniques that are more suited for large areas have to be used.

References

Brusilovskiy, E., (2009). *Spatial Interpolation: A Brief Introduction*. Presentation for Business Intelligence Solutions

Cressie, N. (1993). *Statistics for Spatial Data* Wiley-Interscience.

Ecoshape, http://www.ecoshape.nl/oyster-reefs-eastern-scheldt.html visited 06-06-2014

Fairbridge, R.W. (1980) *The estuary: its definition and geodynamics cycle.* Chichester: John Wiley & Sons. Chemistry and Biochemistry of Estuaries, p 1-35

Huggett, R. J. (2007). *Fundamentals of geomorphology*. Routledge. 2Nd edition 2008, p 334

Kadaster http://www.kadaster.nl/web/Themas/Registraties/Rijksdriehoeksmeting/Rijksdriehoeksstelsel.htm

NIOZ www.nioz.nl and http://www.nioz.nl/yre-en

Okabe, A., Boots, B., Sugihara, K., (1992) *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*, Wiley, New York, USA

Pluymaekers, S.J.A., (2007), *A data analysis to bed dynamics in the Western Scheldt estuary*, Master thesis geomatics

M.J. Pyrcz and C.V. Deutsch (2003) *The whole story on the hole effect* Newsl. Geostat. Assoc. Australas, 18:3–5

Riegl LMS (2013), *Riegl VZ-400 fact sheet,* 18-09-2013

Sedgewick, R., (1992) *Algorithms in C*, Addison-Wesley, Reading, USA,

Smaal, A. C., & Nienhuis, P. H. (1992). The Eastern Scheldt (The Netherlands), from an estuary to a tidal bay: a review of responses at the ecosystem level. *Netherlands journal of sea research*, 30, 161-173.

Teunissen, P.J.G., (2000a) *Adjustment Theory; an introduction,* 1st ed. Delft University Press, Delft, The Netherlands,

Teunissen, P.J.G., (2000b) *Testing Theory; an introduction,* 1st ed. Delft University Press, Delft, The Netherlands,

Trauth, M. H., Gebbers, R., Marwan, N., & Sillmann, E. (2010). *MATLAB recipes for earth sciences* 3rd edition, Berlin: Springer.

Vosselman, G., Maas, H-G. (editors) et al, (2010) *Airborne and terrestrial laser scanning,* CRC press, Taylor and Francis group

Zandmotor, de, www.zandmotor.nl visited 06-06-2014

Appendices

A: List of figures

B: List of tables

Matlab scripts

The scripts made are: *Gridbuilder*, *differencemap*, *Transectbuilder*, *Variotest* and *Varioplot*.

Gridbuilder:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This program makes a grid for the heights contained in a matrix with
% structure [x y z]. The edge length can be specified, after which the
% points in each cell will be averaged to obtain a single height. Height,
% amount of points per cell and standard deviation per cell are output.
% 2014, A H Visser
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear all; close all; clc
tic % Timer for reading time
% Read and sort the dataset
Originaldata = dlmread('Viane_2013-10-24.txt'); %Read dataset
toc
tic %Start timer (purely for reference purposes)
Edgelength = 0.5; %Grid edge length in metre
Originaldata (:,1) = round(1/Edgelength*(Originaldata(:,1))); %Round the dataset
Originaldata (:,2) = round(1/Edgelength*(Originaldata(:,2))); %Round the dataset
Xmax = max(Originaldata(:,1));
Ymax = max(Originaldata(:,2));

SortedData = sortrows(Originaldata); %Sort the dataset on ascending x-
coordinate,
% within an x coordinate on ascending y coordinate,
% within a y coordinate on z coordinate
clear Originaldata %Clear this to speed up calculations, by freeing up memory
Grid = zeros(Ymax,Xmax); %Make the grid for heights, Nvalues and SDs
Nvalues = zeros(Ymax,Xmax);
SDvalues = zeros(Ymax,Xmax);

% Used to cut any negative values, can be expanded for y is nessecary.
% If there are lots of negative coordinates, consider translating the grid
instead
ll = 1;
while SortedData(ll,1) < 1
    SortedData(ll,:) = 0;
    ll = ll+1;
end
SortedData(1:ll-1,:)=[]; % Delete the zeros resulting from the loop
LengthSD = length(SortedData);
GridValue = zeros(ceil((LengthSD)/1000),1); % Set up the collection table
jj = 1;

while jj < LengthSD %Crawl through the data pointwise
    XXX = (SortedData(jj,1));
    YYY = (SortedData(jj,2));
    ii = jj;
    nn = 1;
    while SortedData(ii,1) == XXX && SortedData(ii,2) == YYY && ii < LengthSD
        GridValue(nn) = SortedData(ii,3); %As long as (x,y) is the same,
        nn = nn+1;                        %collect heights
        ii = ii+1;
    end
    GridValue(nn:end) = [];
    SDVal = std(GridValue(1:nn-1));
```

```
    MeanV = mean(GridValue(1:nn-1));
    GVL = length(GridValue);
    for kk = 1:GVL % Remove all points more than 3 SDs away from the mean
        if GridValue(kk)>MeanV+3*SDVal || GridValue(kk)<MeanV-3*SDVal
            GridValue(kk) = 0;
        end
    end
    GridValue(~any(GridValue,2),:)=[];
    SDvalues(YYY,XXX) = std(GridValue); %Calculate the standard deviation of the
points in this cell
    Grid(YYY,XXX) = mean(GridValue); %Store the height of this cell in the Grid
    Nvalues(YYY,XXX) = length(GridValue); %Store the amount of points in this
cell
    GridValue = zeros(ceil(LengthSD/1000),1); %Reset the collection table
    jj = ii; %Commence crawling the dataset at the next gridpoint
end
toc

%Write all three matrices to file
dlmwrite('NewSubsampledGrid0.25sqmViane_2013-10-24.txt', Grid, 'newline', 'pc')
%Write the heigth, amount of points
dlmwrite('NewSDvalues0.25sqmViane_2013-10-24.txt', SDvalues, 'newline', 'pc')
%per cell and standard deviations to file
dlmwrite('NewNvalues0.25sqmViane_2013-10-24.txt', Nvalues, 'newline', 'pc')
```

Differencemap:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This program combines the output of Gridbuilder for two epochs in the
% same area. Maps are created for height, height difference, deformation
% and points per cell.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear all; close all; clc

tic
% Read all six datasets
SetA = dlmread('SubsampledGrid0.25sqmVal_2012-02-27.txt'); %Earlier set heights
SetB = dlmread('SubsampledGrid0.25sqmVal_2013-02-12.txt'); %Later set heights
NSetA = dlmread('Nvalues0.25sqmVal_2012-02-27.txt'); %Earlier set n-values
NSetB = dlmread('Nvalues0.25sqmVal_2013-02-12.txt'); %Later set n-values
SSetA = dlmread('SDvalues0.25sqmVal_2012-02-27.txt'); %Earlier set standard
deviations
SSetB = dlmread('SDvalues0.25sqmVal_2013-02-12.txt'); %Later set standard
deviations
toc

tic
GridSize = 0.5; % The length of the edges of the grid in metre
SetA(1:200/GridSize,:)=[]; %Delete the southern 200m for plotting purposes
SetB(1:200/GridSize,:)=[];
NSetA(1:200/GridSize,:)=[];
NSetB(1:200/GridSize,:)=[];
SSetA(1:200/GridSize,:)=[];
SSetB(1:200/GridSize,:)=[];

[A,B] = size(SetB); %Determine the size of both sets
[C,D] = size(SetA);

if A>C                        %Expand the sets to the same size
```

```
    SetA((C+1:A),:)=0;
    NSetA((C+1:A),:)=0;
    SSetA((C+1:A),:)=0;
else
    SetB((A+1):C,:)=0;
    NSetB((A+1):C,:)=0;
    SSetB((A+1):C,:)=0;
end
if B>D
    SetA(:,(D+1:B))=0;
    NSetA(:,(D+1:B))=0;
    SSetA(:,(D+1:B))=0;
else
    SetB(:,(B+1:D))=0;
    NSetB(:,(B+1:D))=0;
    SSetB(:,(B+1:D))=0;
end

[E,F] = size(SetB); %Redetermine the size of both sets

% Cells containig no data, thus those which are 0, are set to NaN to avoid
% a single coloured pane across the set, as well to calculate differences
% only from those cells containing data in both epochs.
for ii = 1:E
    for jj = 1:F
        if SetB(ii,jj)== 0
            SetB(ii,jj) = NaN;
            NSetB(ii,jj) = NaN;
        end
        if SetA(ii,jj)== 0
            SetA(ii,jj) = NaN;
            NSetA(ii,jj) = NaN;
        end
    end
end

% The stability test
T = zeros(E,F);
VSetA = SSetA.*SSetA;
VSetB = SSetB.*SSetB;

for kk = 1:E
    for ll = 1:F
        T(kk,ll) = ((SetA(kk,ll)-(VSetA(kk,ll)*SetB(kk,ll)
+VSetB(kk,ll)*SetA(kk,ll))/(VSetA(kk,ll)+SSetB(kk,ll)))^2)/VSetA(kk,ll) ...
             + ((SetB(kk,ll)-(VSetA(kk,ll)*SetB(kk,ll)+VSetB(kk,ll)*SetA(kk,ll))/
(VSetA(kk,ll)+VSetB(kk,ll)))^2)/VSetB(kk,ll);
    end
end

%Make the difference map after the datasets have been NaN-ed,
% because cells that do contain a value in one set but dont in another
% should not show up
Changemap = SetB-SetA;
Changemap0 = Changemap;
Changemap0(isnan(Changemap0)) = 0;
TotalSand = round(sum(sum(Changemap0))*(GridSize^2));
TotalHigh = round(sum(sum(Changemap0(300:700,460:780))*(GridSize^2))); %Val:
300:700,460:780 Viane: 960:1240,1060:1460
TotSand = ['The total amount of sand displaced is ', num2str(TotalSand), ' cubic
metre'];
```

```
HighSand = ['The total amount of sand accumulated behind the oyster reef is ',
num2str(TotalHigh), ' cubic metre'];
disp(TotSand)
disp(HighSand)

MaxDiff = max(max(Changemap));    %With max,min,sum it calculates it per column
MinDiff = min(min(Changemap));    %by using the operator twice, it operates on
the
                                  %row vector of the first operator
text = ['The maximum deposition is ', num2str(MaxDiff), 'm', ...
    ' and the maximum erosion is ', num2str(MinDiff), 'm'];
disp(text)

xcoor = GridSize:GridSize:F*GridSize;
ycoor = GridSize:GridSize:E*GridSize;

figure;
surface(xcoor, ycoor, SetA, 'EdgeColor','none');
caxis([-1.2 1])
set(gca, 'fontsize', 20)
title 'Val 27 02 2012'
xlabel 'distance to field origin [m]'
ylabel 'distance to field origin [m]'
colorbar
hold;

figure;
surface(xcoor, ycoor, SetB, 'EdgeColor','none');
caxis([-1.2 1])
set(gca, 'fontsize', 20)
title 'Val 12 02 2013'
xlabel 'distance to field origin [m]'
ylabel 'distance to field origin [m]'
colorbar
hold;

figure;
surface(xcoor, ycoor, Changemap, 'EdgeColor','none');
caxis([-0.5 0.5])
set(gca, 'fontsize', 20)
title 'Difference between 27 02 2012 and 12 02 2013'
xlabel 'distance to field origin [m]'
ylabel 'distance to field origin [m]'
colorbar
hold;

% Critical value for the stability test to be considered true
Kalpha = chi2inv(0.99,1);

Tmap = zeros(E,F);
for oo = 1:E
    for tt = 1:F
        if T(oo,tt)>Kalpha
            Tmap(oo,tt )= 1;
        else
            Tmap(oo,tt) = -1;
        end
    end
end
Tmap=Tmap.*(~isnan(T));
for mm = 1:E
```

```
    for nn = 1:F
        if Tmap(mm,nn)== 0
            Tmap(mm,nn) = NaN;
        end
    end
end

figure;
surface(xcoor, ycoor, Tmap, 'EdgeColor','none');
caxis([-1.1 1.3])
set(gca, 'fontsize', 20)
title 'Deformation areas at Val between 27 02 2012 and 12 02 2013'
xlabel 'distance to field origin [m]'
ylabel 'distance to field origin [m]'
hold;

figure;
surface(xcoor, ycoor, NSetA, 'EdgeColor','none');
set(gca, 'fontsize', 20)
title 'Points per cell 27 02 2012 at Val'
xlabel 'distance to field origin [m]'
ylabel 'distance to field origin [m]'
logzplot colorbar
hold;

figure;
surface(xcoor, ycoor, NSetB, 'EdgeColor','none');
set(gca, 'fontsize', 20)
title 'Points per cell 12 02 2013 at Val'
xlabel 'distance to field origin [m]'
ylabel 'distance to field origin [m]'
logzplot colorbar

toc
```

Transectbuilder:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This program combines the output of Gridbuilder for two epochs in the
% same area and makes a transect through it.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear all; close all; clc
tic
% Read the fine gridded data
SetA = dlmread('SubsampledGrid0.01sqmViane_2012-11-06.txt');
SetB = dlmread('SubsampledGrid0.01sqmViane_2013-10-24.txt');
toc

Location = 500; % y-coordinate of the transect
Location = Location-200;

GridSize1 = 0.1; % The length of the edges of the grid in metre
SetA(1:200/GridSize1,:)=[]; %Delete the southern 200m for plotting purposes
SetB(1:200/GridSize1,:)=[];

[A,B] = size(SetB); %Determine the size of both sets
[C,D] = size(SetA);

% Expand the sets to the same size
```

```
if A>C
    SetA((C+1:A),:)=0;
else
    SetB((A+1):C)=0;
end
if B>D
    SetA(:,(D+1:B))=0;
else
    SetB(:,(B+1:D))=0;
end

% Extract the transects
TransA = SetA(Location/GridSize1,:).';
TransB = SetB(Location/GridSize1,:).';
E = length(TransA);

% Make all zero's, thus cells without data, NaN, in order to obtain a good
% plot
for ll = 1:E
    if TransA(ll) == 0
        TransA(ll) = NaN;
    end
end

for ii = 1:E
    if TransB(ii) == 0
        TransB(ii) = NaN;
    end
end

xA = 1:0.1:E/10;
xA = xA(~isnan(TransA));
xB = 1:0.1:E/10;
xB = xB(~isnan(TransB));

tic
% Read the coarse gridded data
SetC = dlmread('SubsampledGrid0.25sqmViane_2012-11-06.txt');
SetD = dlmread('SubsampledGrid0.25sqmViane_2013-10-24.txt');
toc

GridSize2 = 0.5; % The length of the edges of the grid in metre
SetC(1:200/GridSize2,:)=[]; %Delete the southern 200m for plotting purposes
SetD(1:200/GridSize2,:)=[];

[F,G] = size(SetD); %Determine the size of both sets
[H,I] = size(SetC);

% Expand the sets to the same size
if F>H
    SetC((H+1:F),:)=0;
else
    SetD((F+1):H)=0;
end
if G>I
    SetC(:,(I+1:G))=0;
else
    SetD(:,(G+1:I))=0;
end

% Extract the transects
```

```
TransC = SetC(Location/GridSize2,:).';
TransD = SetD(Location/GridSize2,:).';
J = length(TransC);

% Make all zero's, thus cells without data, NaN, in order to obtain a good
% plot
for ll = 1:J
    if TransC(ll) == 0
        TransC(ll) = NaN;
    end
end

for ii = 1:J
    if TransD(ii) == 0
        TransD(ii) = NaN;
    end
end

xC = 1:0.5:E/10;
xC = xC(~isnan(TransC));
xD = 1:0.5:E/10;
xD = xD(~isnan(TransD));

% Plot the four lines in one plot
figure
plot(xC, TransC(~isnan(TransC)), 'Color', 'b', 'LineStyle', '--');
hold all;
plot(xD, TransD(~isnan(TransD)), 'Color', 'r', 'LineStyle', '--');
hold all;
plot(xA, TransA(~isnan(TransA)), 'Color', 'b');
hold all;
plot(xB, TransB(~isnan(TransB)), 'Color', 'r');
hold all;
LEGEND = legend ('6-11-2012 (50cm)', '24-10-2013 (50cm)', '6-11-2012 (10cm)',
'24-10-2013 (10cm)');
set(gca, 'fontsize', 20)
set(LEGEND, 'Location', 'NorthWest')
ylabel 'height above NAP [m]'
xlabel 'Horizontal distance [m]'
```

Variotest:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% First the matrix containing the amount of points per cell will be read,
% then all cells with more than 200 and less than 20.000 will be extracted.
% The original data is then read and sorted, after which the points given
% by the NSet are grouped and a variogram is calculated for them.
% From the variogram the various parameters are stored and processed to
% obtain the amplitude, wave length and direction of the bed forms.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear all; close all; clc

tic
NSet = dlmread('Nvalues0.25sqmVal_2012-02-27.txt'); %Read N-values

[MM,NN] = size(NSet);
CoorTable = zeros(MM*NN,5);
```

```
ii=1;
for mm = 1:MM
    for nn = 1:NN
        if NSet(mm,nn) >= 200 && NSet(mm,nn) <= 20000 % exctract desired points
            CoorTable(ii,1) = nn;
            CoorTable(ii,2) = mm;
        end
        ii = ii+1;
    end
end
CoorTable(~any(CoorTable,2),:) = []; % Delete the zero rows
CoorTable = sortrows(CoorTable);
Dunes = CoorTable; % Set up the parameter collection table
CoorTable(:,1) = CoorTable(:,1)/2; % Translate them to coordinates
CoorTable(:,2) = CoorTable(:,2)/2;
CoorTable(:,3:5) = [];
toc

clear NSet; clear MM; clear NN;

tic
Originaldata = dlmread('Val_2012-02-27.txt'); %Read dataset
SortedOD = sortrows(Originaldata); % Sort the data for easier processing
clear Originaldata; % Free memory
SortedOD(:,1) = SortedOD(:,1); %-500 for Val
SortedOD(:,2) = SortedOD(:,2); %-600 for Val
toc

tic
for cellNo = 1:length(CoorTable); % Crawl the table with the coordinates of
    xx = CoorTable(cellNo,1);      % cells with more than 200 points
    yy = CoorTable(cellNo,2);
    % Only search a small part, namely within the x-bounds to speed things
    % up
    [DuneminX,~,~] = find(SortedOD(:,1) >= xx-0.25, 1,'first');
    [DunemaxX,~,~] = find(SortedOD(DuneminX:end,1) > xx+0.25, 1,'first');
    DunemaxX = DunemaxX-1+DuneminX;

    VarioSet = zeros(DunemaxX-DuneminX,3);

    % Find the points in the cell, x bounds are given, the y bounds
    % are searched for here
    kk = DuneminX;
    ll = 1;
    while kk < DunemaxX
        if SortedOD(kk,2) >= yy-0.25 && SortedOD(kk,2) <= yy+0.25
            VarioSet(ll,:) = SortedOD(kk,:);
        end
        ll = ll+1;
        kk = kk+1;
    end
    VarioSet(~any(VarioSet,2),:) = [];

    % Sub sample if there are more than 3000 points in a cell
    if ll>3000
        % Calculate the variogram with 20 distance bins and 1 degree angle
        VarioValues = variogram(VarioSet(:,1:2),VarioSet(:,3), ...
            'nrbins', 20, 'anisotropy', true, 'thetastep', 1, ...
            'subsample', 3000);

        N = size(VarioValues.val,1);
```

```
        % Search for the direction with the least variation
        SDval = std(VarioValues.val);
        [~,MinSDcol,~] = find(min(SDval));
        Dune.Dir = VarioValues.theta(1,MinSDcol)+pi/2;
        % Directions range is [0 pi], everything is brought to this range
        if Dune.Dir > pi
            Dune.Dir = Dune.Dir - pi;
        end

        % Columns are numbered in degrees
        DuneGrad = round(Dune.Dir/pi*180);

        % The maximum is the ripple crest
        [ValRow,~,~] = find(max(VarioValues.val(:,DuneGrad)));

        % Finding the first local minimum after the maximum
        for mm = ValRow+1:N-1;
            ValMin =
1*(VarioValues.val(mm,DuneGrad)<VarioValues.val(mm+1,DuneGrad) ...
                && VarioValues.val(mm,DuneGrad)<VarioValues.val(mm-1,DuneGrad));
        end

        % the distance to the first local minimum after the maximum gives
        % the wave length of the bed form
        [WaveRow,~,~] = find(ValMin==1,1,'first');

        % If there is no trend in the points within the cell, it's possible
        % no local minimum occurs, in order to prevent errors, those are
        % set to NaN

        if ~isempty(WaveRow)
            Dune.Length = VarioValues.distance(WaveRow,1);
            Dune.Amp = sqrt(max(VarioValues.val(:)));
        else
            Dune.Length = NaN;
            Dune.Amp = NaN;
            Dune.Dir = NaN;
        end

        Dunes(cellNo,3) = Dune.Amp;
        Dunes(cellNo,4) = Dune.Length;
        Dunes(cellNo,5) = Dune.Dir;

        clear VarioValues M N SDval Dune DuneGrad;
        clear ValRow ValMin WaveRow;
    else % Do not sub sample cells with less than 3000 points
        % Calculate the variogram with 20 distance bins and 1 degree angle
        VarioValues = variogram(VarioSet(:,1:2),VarioSet(:,3), ...
            'nrbins', 20, 'anisotropy', true, 'thetastep', 1);

        N = size(VarioValues.val,1);

        % Search for the direction with the least variation
        SDval = std(VarioValues.val);
        [~,MinSDcol,~] = find(min(SDval));
        Dune.Dir = VarioValues.theta(1,MinSDcol)+pi/2;
        % Directions range is [0 pi], everything is brought to this range
        if Dune.Dir > pi
            Dune.Dir = Dune.Dir - pi;
        end
```

```
        % Columns are numbered in degrees
        DuneGrad = round(Dune.Dir/pi*180);

        % The maximum is the ripple crest
        [ValRow,~,~] = find(max(VarioValues.val(:,DuneGrad)));

        % Finding the first local minimum after the maximum
        for mm = ValRow+1:N-1;
            ValMin =
1*(VarioValues.val(mm,DuneGrad)<VarioValues.val(mm+1,DuneGrad) ...
                && VarioValues.val(mm,DuneGrad)<VarioValues.val(mm-1,DuneGrad));
        end

        % the distance to the first local minimum after the maximum gives
        % the wave length of the bed form
        [WaveRow,~,~] = find(ValMin==1,1,'first');

        % If there is no trend in the points within the cell, it's possible
        % no local minimum occurs, in order to prevent errors, those are
        % set to NaN

        if ~isempty(WaveRow)
            Dune.Length = VarioValues.distance(WaveRow,1);
            Dune.Amp = sqrt(max(VarioValues.val(:)));
        else
            Dune.Length = NaN;
            Dune.Amp = NaN;
            Dune.Dir = NaN;
        end

        Dunes(cellNo,3) = Dune.Amp;
        Dunes(cellNo,4) = Dune.Length;
        Dunes(cellNo,5) = Dune.Dir;

        clear VarioValues M N SDval Dune DuneGrad;
        clear ValRow ValMin WaveRow;
    end
end
toc

% Store the matrix containing the location and values of the parameters
save('Dunes0.25sqmViane_2012-11-06.mat', 'Dunes', '-v7.3')
```

Varioplot:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This program combines the output of Variotest for two epochs in the
% same area. Maps are created for ripple amplitude, direction and
% wave length.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

clear all; close all; clc;

% load both sets
load('Dunes0.25sqmVal_2012-02-27.mat');
SetA = Dunes; %structure [x y amp length dir]
load('Dunes0.25sqmVal_2013-02-12.mat');
SetB = Dunes;

GridSize = 0.5; % Grid size in metre
```

```
maxX = max(SetA(1,:));
maxY = max(SetA(2,:));

% Make matrices per parameter per epoch
AAmp = zeros(maxY,maxX);
BAmp = zeros(maxY,maxX);
ALength = zeros(maxY,maxX);
BLength = zeros(maxY,maxX);
ADir = zeros(maxY,maxX);
BDir = zeros(maxY,maxX);

% Extract all parameter values and store them in the grid
for ii = 1:length(SetA)
    xx = SetA(ii,1);
    yy = SetA(ii,2);
    AAmp(yy,xx) = SetA(ii,3);
    ALength(yy,xx) = SetA(ii,4);
    ADir(yy,xx) = SetA(ii,5);
end

for jj = 1:length(SetB)
    xx = SetB(jj,1);
    yy = SetB(jj,2);
    BAmp(yy,xx) = SetB(jj,3);
    BLength(yy,xx) = SetB(jj,4);
    BDir(yy,xx) = SetB(jj,5);
end

% Determine the size of both sets
[A,B] = size(BAmp);
[C,D] = size(AAmp);

% Expand the sets to the same size
if A>C
    AAmp((C+1:A),:)=0;
    ALength((C+1:A),:)=0;
    ADir((C+1:A),:)=0;
else
    BAmp((A+1):C,:)=0;
    BLength((A+1):C,:)=0;
    BDir((A+1):C,:)=0;
end
if B>D
    AAmp(:,(D+1:B))=0;
    ALength(:,(D+1:B))=0;
    ADir(:,(D+1:B))=0;
else
    BAmp(:,(B+1:D))=0;
    BLength(:,(B+1:D))=0;
    BDir(:,(B+1:D))=0;
end

% Delete the fringes of the matrix containing no data
% First comment this area and determine the size visually from the plots
AAmp(1:100/GridSize,:)=[];
AAmp(:,1:100/GridSize)=[];
BAmp(1:100/GridSize,:)=[];
BAmp(:,1:100/GridSize)=[];
ALength(1:100/GridSize,:)=[];
ALength(:,1:100/GridSize)=[];
```

```
BLength(1:100/GridSize,:)=[];
BLength(:,1:100/GridSize)=[];
ADir(1:100/GridSize,:)=[];
ADir(:,1:100/GridSize)=[];
BDir(1:100/GridSize,:)=[];
BDir(:,1:100/GridSize)=[];

[E,F] = size(AAmp);

% Cells containig no data, thus those which are 0, are set to NaN to avoid
% a single coloured pane across the set, as well to calculate differences
% only from those cells containing data in both epochs.
for kk = 1:E
    for ll = 1:F
        if AAmp(kk,ll)== 0
            AAmp(kk,ll) = NaN;
            ALength(kk,ll) = NaN;
            ADir(kk,ll) = NaN;
        end
        if BAmp(kk,ll)== 0
            BAmp(kk,ll) = NaN;
            BLength(kk,ll) = NaN;
            BDir(kk,ll) = NaN;
        end
    end
end

xcoor = GridSize:GridSize:F*GridSize;
ycoor = GridSize:GridSize:E*GridSize;

% Calculate the difference maps
AmpDif = AAmp-BAmp;
LDif = ALength - BLength;

% Make the plots
figure;
surface(xcoor, ycoor, LDif, 'EdgeColor','none');
caxis([-0.03 0.03])
set(gca, 'fontsize', 20)
title 'Difference in ripple length at Val'
xlabel 'distance to field origin [m]'
ylabel 'distance to field origin [m]'
colorbar
hold;

figure;
surface(xcoor, ycoor, AAmp, 'EdgeColor','none');
caxis([0 0.1])
set(gca, 'fontsize', 20)
title 'ripple amplitude at Val 27 02 2012'
xlabel 'distance to field origin [m]'
ylabel 'distance to field origin [m]'
colorbar
hold;

figure;
surface(xcoor, ycoor, BAmp, 'EdgeColor','none');
caxis([0 0.1])
set(gca, 'fontsize', 20)
title 'ripple amplitude at Val 12 02 2013'
xlabel 'distance to field origin [m]'
```

```
ylabel 'distance to field origin [m]'
colorbar
hold;

figure;
surface(xcoor, ycoor, ALength, 'EdgeColor','none');
%caxis([-1.2 1])
set(gca, 'fontsize', 20)
title 'ripple length at Val 27 02 2012'
xlabel 'distance to field origin [m]'
ylabel 'distance to field origin [m]'
colorbar
hold;

figure;
surface(xcoor, ycoor, BLength, 'EdgeColor','none');
%caxis([-1.2 1])
set(gca, 'fontsize', 20)
title 'ripple length at Val 12 02 2013'
xlabel 'distance to field origin [m]'
ylabel 'distance to field origin [m]'
colorbar
hold;

figure;
surface(xcoor, ycoor, ADir, 'EdgeColor','none');
caxis([1.5 1.7])
set(gca, 'fontsize', 20)
title 'ripple direction at Val 27 02 2012'
xlabel 'distance to field origin [m]'
ylabel 'distance to field origin [m]'
colorbar
hold;

figure;
surface(xcoor, ycoor, BDir, 'EdgeColor','none');
caxis([1.5 1.7])
set(gca, 'fontsize', 20)
title 'ripple direction at Val 12 02 2013'
xlabel 'distance to field origin [m]'
ylabel 'distance to field origin [m]'
colorbar
hold;
```

Computer specifications

Component	Type
Processor	Intel © i5 750, 2.67GHz
RAM	8GB at DDR3
Motherboard	MSI P55-CD53
Operating System	Windows 7 Professional
Matlab	Version R2012a build 7.14.0.739