Identifying Deformation Regimes At Kijkduin Beach Using Permanent Laser Scanning

Sylke van der Kleij

Supervisors:
Dr. R.C. Lindenbergh
Dr. S.E. Vos

Faculty of Civil Engineering and Geosciences

Coordinator:
Dr. K.H.A.A. Wolf

Delft University of Technology
11th of October 2018

Abstract

The Netherlands has a long coastline, and as such has always had an important relationship with the sea. This coastline changes constantly through various processes and it is therefore important to understand these processes and to ensure that the coast is properly defensed against the threat of the sea. Presently there are still many knowledge gaps which are not yet clearly understood.

This thesis investigates how different deformation regimes on the beach and in the dunes can be identified using k-means clustering. K-means clustering is an unsupervised classification method, which means that the outcome is based on the software's analysis of an image, without specifying classes beforehand. The results are unbiased, and can then be interpreted to see whether deformation regimes can be identified.

Data was acquired at Kijkduin beach using a Permanent Laser Scanning (PLS) system called CoastScan. CoastScan created hourly scans of a one kilometer stretch of the beach, over a six-month time span. The data used in this thesis consists of one scan per day over the course of one month (January 2017). The data was processed using Matlab, including downsampling and interpolating, and time series showing the variations in topography were created.

Incomplete time series were filtered out, as well as time series that were not located on the beach or in the dunes. The remaining time series were clustered into clusters using the k-means clustering algorithm. Clustering was performed for a varying number of clusters. The results created using ten clusters were assessed to be the best, and these results are therefore used in further interpretation. The clustering process was repeated twice, including the height of the data points and excluding the height of the data points. This created a height-based clustering solution and a shape-based clustering solution.

The resulting cluster assignments were then interpreted using the clustered time series, and the interpretations were validated using Google Maps satellite images and field observations. The beach was divided into three different zones following the height-based clustering solution: the intertidal zone, the backshore, and the dunes. Deformation regimes influencing the beach are thought to be the waves and the tides in the intertidal zone, as well as swash. Aeolian processes influence the backshore and the dunes. Variations are also thought to be caused by pedestrians. More variation is observed during the second half of the month, which is thought to be caused by a storm that occurred on the 13th and 14th of January. The disadvantage of the height-based clustering is that the clusters are mostly divided according to height, and no unexpected results are found.

However, the shape-based clustering did provide unexpected results. Following shape-based clustering, the sections defined during height-based clustering were re-evaluated and moved. The backshore disappeared into the intertidal zone and the dunes. Two small spots were found that consisted of outliers in the intertidal zone, interpreted to be caused by birds, and a sand bar migrating land inwards was identified in the intertidal zone.

In conclusion, it is indeed possible to identify different deformation regimes at Kijkduin using k-means clustering of time series. The shape-based clustering was deemed to perform better than the height-based clustering. Finally, recommendations are given concerning further research.

Contents

1 Introduction

The Netherlands has a long coastline, and as such has always had an important relationship with the sea. This coastline changes constantly through various processes and it is therefore important to understand these processes and to ensure that the coast is properly defensed against the threat of the sea. Presently there are still many knowledge gaps which are not yet clearly understood.

The goal of this thesis is to investigate whether the data obtained through the use of the CoastScan permanent laser scan can be used to identify deformation regimes on the beach and help to understand coastal processes better. Being able to identify these deformation regimes could be valuable in protecting the coast. If it is known how the coast changes and at which locations, and whether it weakens the coastal defense, then measures can be taken to mitigate these problems.

1.1 Research Questions

The research question of this thesis is:

How can different deformation regimes be identified at Kijkduin through use of k-means clustering of time series collected by CoastScan?

In order to answer this research question, several subquestions should also be answered:

- What are (expected) properties of the beach at Kijkduin?
- What is a Permanent Laser Scan?
- How can the Permanent Laser Scan data be processed?
- How can a continuous time series be created from the CoastScan data?
- Why use unsupervised classification?
- How does k-means clustering work?
- How can the results be visualised?

1.2 Thesis Outline

This thesis starts in chapter 1 where a general introduction of the thesis is given, and research questions are identified. In chapter 2, information about beach morphology is given and the principle of LiDAR scanning is described, with special attention to the CoastScan scanner. In chapter 3 a site introduction of Kijkduin is given. In chapter 4, the method for processing and analysing of the acquired LiDAR data is outlined. The results found in this analysis are summarised in chapter 5. Chapter 6 discusses the choices made and problems encountered during processing and analysing the data. Chapter 7 returns to the research question posed in chapter 1 and aims to answer this question. The results will then be further discussed and recommendations for further research are given.

2 Monitoring Beaches With Laser Scanning

This chapter contains a literature study on topics relevant to the research. First, an introduction to coast definitions and morphology is given. Secondly, an introduction to Light Detection And Ranging technology is given. Thirdly, the CoastScan project is introduced.

2.1 Coasts

Coasts are the areas of interface between the land and the sea, the shoreline is the actual margin between the land and the sea (Nichols 2009).

The Dutch coast is a depositional coastline which means that the gradient is gentle and a lot of wave energy is dissipated in shallow water (Nichols 2009). Sediment accumulates, the main source is sand replenishments, while rivers are a smaller source, as well as wind-blown sediments or direct erosion of the coastline. This sediment is then deposited on the coast by various natural processes, such as tidal forces, waves, and winds (Nichols 2009).

The coast is not a stable and constant environment. Instead, it is a dynamic environment that can change rapidly in response to natural processes. The coastal zone is the part of the land surface that is influenced by coastal processes. In sandy coasts, this usually extends from the dunes to the point in the sea where the waves still actively interact with the seabed (Masselink and Gehrels 2014). Major physical processes that influence coastal dynamics are waves, winds, currents and tides. The effect of these physical processes on the coast varies. For example, winds generate waves, but also transport sand and sediments to form dunes. Waves breaking on the shore deposit sediments on the coast, but also wash sediments away, eroding the coast (Masselink and Gehrels 2014).

2.1.1 Divisions of a sandy coast

The coastal area of a sandy coach is typically made up of different elements. These different elements are briefly discussed here, starting from the sea and moving land inwards to the dunes.

Subtidal zone: the part of the coast that is always submerged (Nichols 2009).

Shoreface: the region of the shelf between the low-tide mark and the depth to which waves normally affect the sea bottom, which is the fair weather wave base. The lower depth the shore face reaches depends on wave energy in the area, and is typically between 5 and 20 metres. The width of the shoreface is governed by the shelf slope and the depth of the fair water wave base, this can be hundreds of metres to kilometres across (Nichols 2009).

Foreshore or intertidal zone: the seaward part of the beach where waves go back and forth, so the region between mean high water and mean low water. Gently dipping towards the sea. Due to the waves continuously hitting the shore, sediment often has a high degree of roundness and is often sorted very effectively into different sizes. This results in a low-angle stratification of well-sorted, well-rounded sediment, which is characteristic of wave-dominated sandy beach environments (Nichols 2009). Depending on the tidal range, the foreshore can have a vertical distance of anything from a few decimetres to multiple meters. The seaward extent of the foreshore also depends on the slope and it may be anything from a few metres where the shelf is steeply sloping or the tidal range is small, to over a kilometre in places where there is a high tidal range or a gently sloping shelf (Nichols 2009). Sedimentary structures that can be seen during low-tide are wave ripples. However, these ripples can easily be disturbed by animals living in the intertidal area or by humans walking over them (Nichols 2009).

Berm: a ridge marking the division between the foreshore and backshore, although a berm does not have to be present. If there is a berm, seawater only washes over the top of the berm under storm conditions. The berm is formed by sediment deposited on the landward side by waves (Nichols 2009).

Backshore: the part of the beach between the berm and the dunes. Sediments are well-sorted and well-rounded. Mainly coarse sand and medium sand. Sedimentary structures include parallel bedding and low-angle cross-bedding.

Dune: along coasts, any sand that dries out on the upper part of the beach can be lifted up by onshore winds and redeposited as aeolian dunes. Coastal dunes form as ridges parallel to the shoreline, and are held together by vegetation. They can build up to heights over 10 metres and can stretch hundreds of meters land inwards. The limiting factor is the supply of sand from the beach. Aeolian deposits are well-sorted and often consist of fine sand, as fine sand is lighter and more easily picked up and carried by winds (Nichols 2009).

Figure 1 shows a graphical representation of the beach, with the elements that are most important to this thesis.

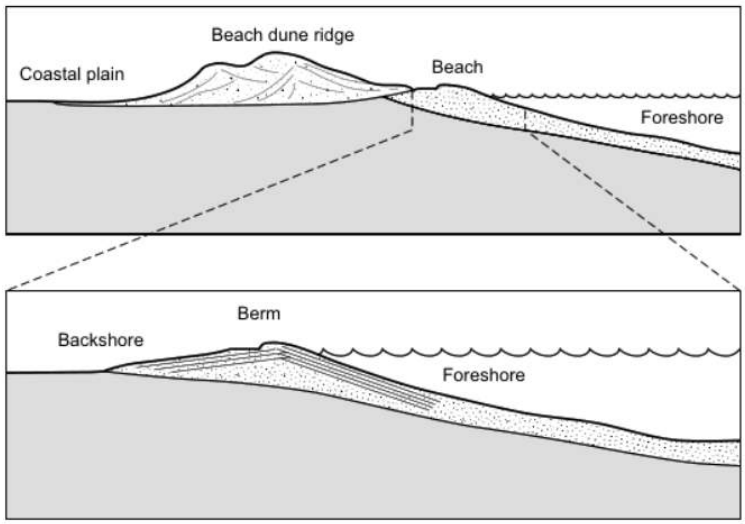


Figure 1: Graphical representation of the beach, including the foreshore and the dunes (after (Nichols 2009), p201)

2.1.2 Spatial-temporal scale

A big challenge in working with coastal areas is the complexity of coastal behaviour over multiple spatial and temporal scales as figure 2 shows.

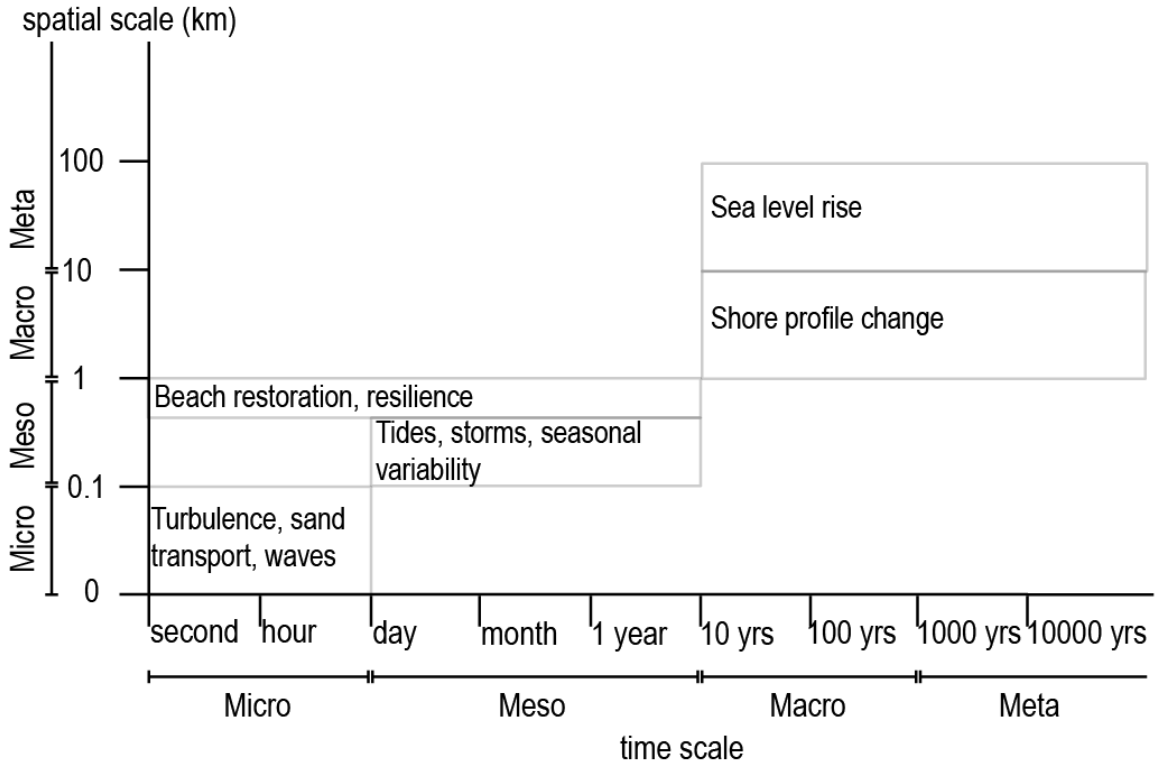


Figure 2: Coastal processes at different spatial-temporal scales, based on the coastal tract (Modified from (Cowell et al. 2003))

The temporal scale of this thesis is in the order of weeks-months, therefore the changes observed are expected to be caused by tidal changes, storms, seasonal changes or due to beach restoration. Spatial changes caused by these processes are likely to be visible in the shape of sand bars. Ripple formation and migration may also be visible.

Beach restoration can be natural, after a storm, or artificial, if sand is added through human intervention. The resilience of the coast depends on the ability to return back to normal after a hazardous event, such as a storm. While the changes caused by a storm often happen over a period of hours or a day, the recovery of the coast can take much longer. This long term recovery is difficult to measure and model due to the spatially small changes happening over a temporally big scale (Vos et al. 2017).

2.1.3 Morphological processes

Various morphological processes influence the shape of the beach. The processes that play the biggest roles in the changing the beach are aeolian processes, waves, and tides. These processes are explained in more detail.

Aeolian: Aeolian transport plays an important role in beach growth (Nichols 2009). As wind blows across a bed of sand, the grains move by saltation, a process which can be explained as hopping. As a sand grain lands, other grains are launched into the air and make a hop before landing and continuing the saltation process. Irregularities in the surface of the sand and turbulence of the air create patches where the sand grains pile up. Grains in these piles are then even more susceptible to being picked up by the wind and saltate. The result is a series of equally spaced piles of sand, aligned perpendicular to the wind. These sand ripples have a height on the mm to cm scale (Nichols 2009). The formation and migration of these ripples may be visible in the CoastScan data.

If these ripples continue to grow, they may eventually become dunes. Dunes have a height of roughly 10 cm to 100 m (Nichols 2009). At low tide, sand in the intertidal zone is exposed to the air. As the sand dries, it is available to be picked up and redeposited by the wind. The sand then accumulates at the head of the beach, trapped in the dunes (Nichols 2009). The growth of dunes takes a lot of time, and is therefore not something that will be visible in the CoastScan data.

Whether sediment is picked up by the wind depends on whether the threshold velocity is reached. If the wind velocity is below the threshold velocity the grain will stay put, if the wind velocity is above the threshold velocity the grain will be lifted into the air (Nichols 2009). Sand in the dune troughs is therefore unlikely to be lifted up, because the wind velocity there is lower than it is at the dune crest.

A rise in the water table will affect aeolian processes if the water comes up to the level of the interdune areas. Wet sediment will not be picked up by the wind, so it becomes stable. A rise in water table therefore means that sediment accumulates. However, if the water table falls, more sediment in the interdune area will become available to be transported by the wind, promoting erosion (Nichols 2009). This process is shown in figure 3.

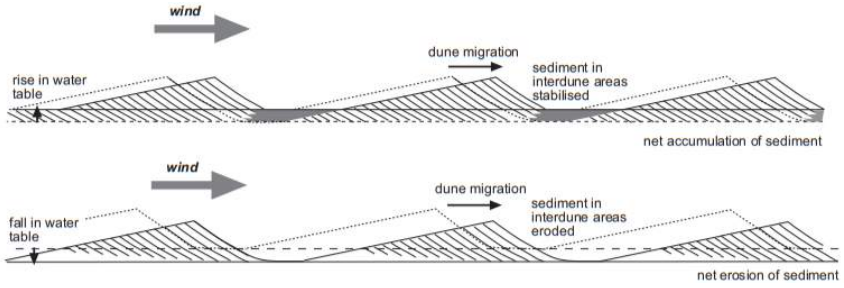


Figure 3: Aeolian dunes are influenced by the level of the groundwater table: if the water table is high the interdune areas are wet and sediment cannot be lifted into the air (after (Nichols 2009), p. 124)

Waves: The transport of sediment cross the shoreface is caused by wave uprush and backwash, together known as swash. Waves are active in the swash zone, the region of the beach where the waves rush up on the beach and then expose the beach when the waves wash back, until a new wave rushes up. Flow in the swash zone is very turbulent, which leads to large sediment transport rates. As this process takes place at the beachface, it plays a big role in reshaping the shoreline (Hobbelen 2018).

The response of a beach to swash sediment depends mainly on the steepness of the beach face. If the slope is too steep, more sediment is eroded by the backwash than is accreted by the runup. If the slope is too gentle, the runup accretes more sand than the backwash erodes. Normally, the swash zone balances around an equilibrium slope (Hobbelen 2018).

Tidal currents: Tides are produced by the gravitational pull of the moon acting on Earth. The tidal range on the west coast of the Netherlands is microtidal, which means that the tidal range is smaller than 2 meters (Nichols 2009). The major impact of tides on the shoreline is the shift between high

and low tide, creating the intertidal zone. The strength of the tidal currents varies throughout the tidal cycle, thus the capacity to carry sediment also varies. If there is a dominant direction of the tidal current, bedforms migrate in that direction (Nichols 2009). Also, tides shift the location of the beach exposed to the swash zone and thus the location of the beach where erosion and accretion takes place.

Sand bar migration: Sand bars are an important source of sand for aeolian transport. During a storm, large amounts of sediment that are eroded from the beach and dunes are deposited in the form of bars on the subtidal part of the beach. When weather conditions then return to normal, the water depth at the new sand bar will be lower than it was before the storm (Hobbelen 2018). Due to this drop in water depth, the waves shoal and their shape becomes skewed and asymmetric. During the passage of these waves, fluid is accelerated strongly. The waves then set the sediment on the sand bars into motion and transport this sand towards the beach (Hsu, Elgar, and Guza 2006). This way, the sand bar migrates towards the beach, extending it. This then means that a larger amount of sand will be available for aeolian transport.

Onshore migration is set back by the occurrence of new storms, as the sand bars are then eroded away again (Hobbelen 2018).

2.2 LiDAR

One of the most used methods for optically measuring 3D surfaces is travel time estimation of laser pulses, also known as LiDAR (Light Detection And Ranging)(Vosselman and Maas 2010). This section explains the basic principles used in LiDAR. This section also elaborates on the different ranging and scanning techniques LiDAR can employ.

2.2.1 LiDAR Principle

LiDAR makes use of light emitted by laser. This light is far more directional, bright and spatially coherent than other light sources. The spatial coherence allows the laser beam to stay in focus when projected on a surface (Vosselman and Maas 2010).

The light then passes through a lens, through a transmission medium, onto the object being scanned. The reflected signal then also passes through a lens and into a photodetection mechanism. It is then possible to distinguish different objects and materials because of the differences in reflections (Vosselman and Maas 2010).

2.2.2 LiDAR Ranging Techniques

LiDAR range measuring works by sending laser pulses (direct time-of-flight) or by sending continuous laser beams (phase-measurement) (Vosselman and Maas 2010). Only time-of-flight laser scanning is applicable to this thesis.

Time-of-flight uses the time delay between sending a signal and recording the reflected signal. The time it takes to send a signal to an object and receive it back is recorded. In direct time-of-flight LiDAR technology, the knowledge that light waves travel with a finite and constant velocity in a given medium is used (Vosselman and Maas 2010). A laser source is used to emit a pulse to an object (Vos et al. 2017). Then, by measuring how long it takes light to travel from the source to the target surface and back to a light detector, the distance between the source and the target object is estimated (Vosselman and Maas 2010). In combination with the known position and orientation of the laser scanner, 3D coordinates of the target object are obtained (Vos et al. 2017).

The method is illustrated in figure 4.

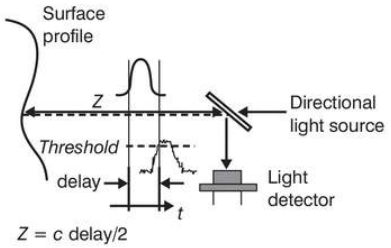


Figure 4: Time-of-flight laser scan method for optically measuring a 3D surface (Vosselman and Maas 2010)

To calculate the range ρ, which is the distance to the object, the speed of light c is divided by a correction factor for air n. This is then multiplied by the time of the round trip τ divided by two (Vosselman and Maas 2010). Equation (2.1) describes this relation:

$$\rho = \frac{c}{n}\frac{\tau}{2} \tag{2.1}$$

The speed of light in a vacuum is c=299 792 458 m/s, or approximated as 3×10^8. As the light travels through air, a correction factor n of 1 can be assumed (Vosselman and Maas 2010).

2.2.3 LiDAR Scanning Techniques

Kinetic

Kinetic airborne LiDAR systems have been available since the early 1970s time, but did not come into widespread use until GPS (Global Positioning System) became available at the end of the 1980s. For airborne LiDAR it is important to know the vertical position of the airplane. With the availability of GPS, a method was developed that allowed precise registration of position and orientation over larger areas. This made it possible to determine the scanner position in horizontal and vertical coordinates in the sub-decimeter range. Nowadays, laser pulse rates reach 300 kHZ (Vosselman and Maas 2010). Airborne laser scanning is a common technique for generating high quality 3D models of the landscape (Vosselman and Maas 2010).

Static

After the success of airborne laser scanning, LiDAR technology also got implemented on static platforms. This type of scanning is called Terrestrial Laser Scanning (TLS). A terrestrial laser scanner uses a LiDAR sensor and a rotating head and mirror to create a near 360x360 degree coverage of the environment of the scanner (Vos et al. 2017). TLS can automatically scan its surroundings at a high speed, measuring millions of points. The scanner acquires a dense point cloud from which a 3D model can be created, with a high level of detail (Lemmens 2011).

Often, TLS is used from different positions to avoid gaps in the point cloud, meaning that the laser scanner has to be moved. The different point clouds obtained from the different scanning positions are then combined to form one final point cloud.

Another option is to keep the scanner stationary and repeatedly scan the same scene to detect changes. This is called a Permanent Laser Scanning system (PLS). It is permanent spatially, as it always scans from the same standpoint, and temporarily, as the system scans continuously (Vos et al. 2017).

Terrestrial laser scanning is illustrated in figure 5.

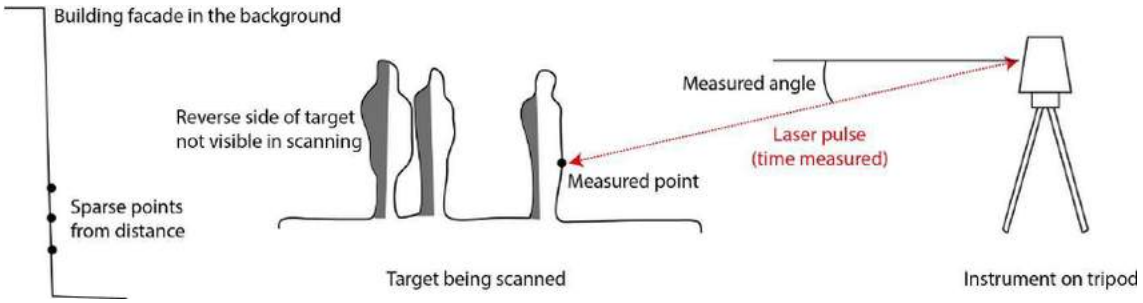


Figure 5: Terrestrial laser scanning principle (Virtanen et al. 2014)

2.2.4 Influences on Accuracy

LiDAR measurements are not perfect and subject to errors caused by different factors that influence the surveying process. These factors and following errors have to be considered in order to be able to evaluate data quality.

The performance of a scanner is described by a lot of different factors, such as accuracy, resolution, range noise and more. Manufacturers generally provide specifications on a scanner's performance, but these specifications are heavily influenced by the scanning conditions and processing procedures. Therefore, the supplied specifications are often not entirely accurate (Reshetyuk 2006).

Accuracy, precision, resolution

When judging the performance of a scanner, it is first important to know the difference between accuracy, precision, and resolution. Accuracy is a measure of how closely the measured values are believed to be to the true values, with the measured value being the range. The accuracy of the range distance is generally on the mm scale (Vos et al. 2017). Precision is the degree of repeatability in a measurement, and is often referred to as noise (Slob 2010). If the same point is scanned twice, the precision indicates to what degree the second measurement differs from initial measurement. Resolution is the size of the smallest feature detectable by the scanner. Resolution depends on the distance between two adjacent pulses, the pulse-spacing (Boehler, Vicent, Marbs, et al. 2003).

Error Sources

This section investigates the factors that influence the measurement process and thus the data quality, i.e. the accuracy of the point cloud. Knowing the accuracy of individual points, an accuracy estimate for the 3D model can be made by using error propagation techniques (Reshetyuk 2006).

There are four main sources of error in time-based laser measurements: instrumental, object-related, environmental and methodological (Reshetyuk 2006). These errors are not fully separate from each other and can influence each other, making it hard to say which of these is the most important error source. As CoastScan is a Permanent Laser Scanner using the time-of-flight principle (Vos et al. 2017), errors only present in airborne scanning or phase-based scanning will not be considered.

Instrumental Error

Instrument mechanism includes the hardware properties, calibration and settings. Additionally, the scanner settings such as point spacing and maximum range also influence the quality of the data, as a smaller point spacing and smaller range generally give a better quality (Soudarissanane et al. 2011). The placement of the scanner relative to the location and orientation of the scanned surface determines the incidence angle, the range and the point density of the laser points sampling the surface.

(Soudarissanane et al. 2011). Because of the low incidence angle used in CoastScan, scatter increases, making it hard to map water.

Object-Related Error
Object-related errors are caused by the objects being scanned. For example, reflectance, size, curvature and orientation are all factors that can cause object-related errors (Reshetyuk 2006).

The strength of the reflected signal is heavily influenced by the reflective abilities of the surface. A return echo coming from a low reflecting target will have a lower amplitude compared to that from a high reflecting target. Systems may wrongly put out longer ranges for low reflecting targets (Vosselman and Maas 2010). Shiny surfaces also influence the reflective abilities of a surface, as they are hard to record. Different reflectivity's can result in systematic errors in range (Boehler, Vicent, Marbs, et al. 2003). An example of a shiny surface CoastScan may encounter is the sea.

The laser beam will have a certain size that may not completely match with an object's size. It is possible that the beam hits an object edge, and that part will be reflected there, while the rest of the light is reflected from the adjacent surface, a different surface behind the object or not at all. The scanner will then measure incorrect points and the object representation in the point cloud will be larger than it really is. This phenomenon is also referred to as edge effect (Boehler, Vicent, Marbs, et al. 2003). Edge effect is likely to occur more the larger the distance to the object gets.

Environmental Error
The difficulty of environmental errors is the large variability they introduce. There are a lot of different environmental factors that can cause small errors. Only the factors that apply for CoastScan are considered.

- Light: the higher the relative reflectivity difference between different parts of the area of interest, the better a structure can be identified. If light is shining on the area of interest very brightly, the intensity difference will get smaller, making it harder to identify structures through relative reflectivity (Boehler, Vicent, Marbs, et al. 2003).
- Precipitation: rain droplets can also influence scans, as they can reflect light, resulting in noise in the data (Boehler, Vicent, Marbs, et al. 2003). RIEGL claims that their scanner can identify and delete range measurements caused by rain drops or dust using deviation and reflectance filters (RIEGL 2017). There was less rain than average in January, but on the 12th and 13th there was snow (KNMI 2017), which may show up in the CoastScan data if the snow did not immediately melt.
- Fog: Fog causes the number of points per scan to decrease and strongly decreases the effective range of the scanner (Hejbudzka et al. 2010). Fog occurs in the dataset on the 9th of January.
- Temperature: a scanner will only function properly if it is used within a certain temperature range. In the case of CoastScan, the temperature is not expected to exceed these boundaries. However, even within this temperature range, deviations in distance measurement may occur (Boehler, Vicent, Marbs, et al. 2003). January 2017 was colder than average with an average temperature of 1.6 °C instead of 3.1 °C, especially the second half of the month was cold (KNMI 2017).

Methodological Error
Methodological errors are errors that emerge due to the chosen survey methodology.

One important error is caused by the approach chosen for georeferencing the point clouds. CoastScan uses direct georeferencing, where the scanner determines its own position and orientation independently through use of targets in the scanned area with known GPS locations (Vos et al. 2017). The

scanner then uses its own position and orientation to obtain the coordinates of all points in the point cloud (Reshetyuk 2006). If the laser orientates itself incorrectly, the laser uses the wrong location to compute all further measurements, resulting in position offset errors. Therefore, the absolute accuracy of CoastScan data is heavily dependent on correct GPS data of the location of the scanner. However, in this thesis the relative accuracy is of more importance, and the coordinates relative to each other remain correct even if the scanner's location was computed incorrectly.

2.3 Measuring Coasts Through CoastScan

Having broadly introduced LiDAR technology, the focus will now be on the CoastScan specifically, the Permanent Laser Scanning system used to acquire the data used in this thesis.

2.3.1 Set-Up

CoastScan is a Permanent Laser Scanning system (PLS) that was situated in Kijkduin, on the roof of a hotel, overlooking the beach. The area of the beach that is measured by the CoastScan scanner is approximately 1 km over the length of the shore and covers the sea, beach and dunes (Hobbelen 2018).

The setup consisted of a Riegl-VZ 2000 laser scanner mounted on a stable measuring frame, protected by all-weather protective housing. The scanner was not moved from its location during data acquisition. It scanned the same section of the beach continuously, creating one high-resolution scan every day and hourly low-resolution scans. Therefore, the CoastScan setup was a Permanent Laser Scanning system (Vos et al. 2017). The Riegl-VZ 2000 has two processors, allowing it to perform data registration and data acquisition at the same time (RIEGL 2017).

The Riegl-VZ 2000 makes use of single pulse ranging and time-of-flight measuring to obtain 3D coordinate measurements. The laser is of Laser Class 1, which means it is safe for the human eye. Therefore, it is safe to use CoastScan continuously on the beach, where people are exposed to the laser (RIEGL 2017).

Figure 6: The Riegl-VZ 2000 laser scanner on top of Hotel NH Atlantic Den Haag at the test location (after (Vos et al. 2017))

The scanner does not know its own position in space. To help the scanner locate itself, six 5-centimeter reflectors are more or less evenly spread over a 180 degrees window, both on the beach and on the rooftop the scanner is located on. The scanner uses these reflectors to orientate itself with an accuracy of 3-5 cm (Vos et al. 2017).

The purpose of CoastScan is the structural monitoring of the beach and provide measurements over multiple spatio-temporal scales. Most strategies miss either the spatio-temporal detail or the quality to provide enough information, especially if the time period over which data is acquired is long. A permanent laser scan can provide data at the necessary spatial and temporal scales (Vos et al. 2017).

CoastScan has trouble capturing data points in water, and it is therefore assumed that except for a few scattered data points, the scanner only captures the beach and not the sea.

The set-up is shown in figure 6.

2.3.2 Expected Quality of Measurements

According to Riegl, the accuracy is 5 mm with a precision of 3 mm, with a maximum range of 2500 m (RIEGL 2017).

The system can acquire data with a speed of up to 500,000 measurements per second (RIEGL 2017). As the point clouds generally contain 2-3 million measurements, the scanning time is expected to be between 4 to 6 seconds. In this short time scale, it is unlikely that an event will happen that greatly affects the scan.

To check the accuracy of the scanner in the vertical Z direction, a path at the top of the dunes was analysed by taking the average Z-value. The path is expected to stay the same over time, and should have the same position over time (Hobbelen 2018). This was found to be true in general, except for one to three days per scan, which can be explained by the presence of people or other temporary objects.

2.3.3 Aim

Previously existing measuring campaigns generally covered large areas and large time periods, which resulted in low spatial and temporal resolutions. Due to low spatial and temporal resolutions, changes occurring on the micro and mezzo scale may be omitted (Vos et al. 2017).

CoastScan has the necessary spatial and temporal scale to monitor the beach closely. The aim of the CoastScan project was to provide measurements over multiple spatio-temporal scales. The aim of the CoastScan project was to have hourly measurements of the beach, for several consecutive months. CoastScan was active from November 2016 until May 2017 (Hobbelen 2018). Around Christmas 2016 a bug prevented CoastScan from data acquisition, but this issue was solved before 2017 started. From then CoastScan functioned without fail, except for one day in February (Vos et al. 2017).

3 Site Description

The data is acquired at Kijkduin, a subdistrict of The Hague, the Netherlands (see figure 7). Kijkduin is bordered by the North Sea and as such Kijkduin is located in the dunes and on the beach.

The beach is sandy, and the beach and dune row are naturally very narrow in this area. In addition to the protective barrier of the beach and dunes already being so narrow, erosion is a problem. The Dutch coast suffers from structural erosion, mainly from the beaches and from the shore face of the first row of dunes. Sand is blown away by the wind or eroded away by the sea during storms at high tide. In order to maintain the coastline of 1990, artificial sand was added to the beach every year, this process is called beach nourishment (Lindenbergh et al. 2011). Nourishment is usually a repetitive process, as nourishment does not stop the physical forces that cause erosion but only mitigates the effects (Rijkswaterstaat 2018).

Figure 7: Study site (Google Maps)

However, just below the area of interest the sand engine was constructed in 2011 (also visible in figure 7 in the bottom left of the closeup). The sand engine is a sandbar-shaped peninsula expected to further widen the beach at Kijkduin. It is an experiment in the management of a dynamic coastline, with the aim of using natural processes to keep the beach and dunes at a safe width (Stive et al. 2013). The idea is that the sand will be redistributed along the coast by waves, wind and currents acting on the sand engine, making beach nourishment unnecessary for at least 20 years. These processes also continuously change the shape and size of the sand engine: during the first year alone the maximum width decreased from 0.96 km to 0.84 km, while its length increased from 2.4 km to 3.6 km (Stive et al. 2013). How exactly the sand engine will develop over time is hard to predict. In addition to the peninsula, two other sand additions were added under water.

The dune height along the Kijkduin beach is more than 10 meters above sea level and as such form a good protection mechanism against the sea (Hobbelen 2018).

During the spring and summer the beach is frequently visited by beach goers, this number is a lot lower during winter, when the beach is mainly visited by hikers and windsurfers. These people will show up in the scanning data.

The tidal range near the study site is approximately 1.8 m, the intertidal zone is approximately 70-80 meters wide. (Hobbelen 2018).

4 Methodology

4.1 Preparing Data

A LiDAR scanner sends new laser pulses at a high rate, producing a lot of range measurements resulitng in a 3D cloud of coordinates. Using these data points, a 3D image of the region of interest is constructed.

The first task is to convert the raw data into positions in a geodetic reference system, with a x-, y- and z-coordinate. This creates a point cloud, which is the basis for further processing (Lemmens 2011). The point clouds used in this thesis were provided, therefore making this step unnecessary.

3D point clouds are simple, but very powerful collections of data points, able to represent shape, size, position and orientation of objects in space. Classification can be difficult, as point clouds can be very variable due to factors such as irregular sampling or different types of objects (Grilli, Menna, and Remondino 2017).

The data used in this research is a series of laser scans, one per day for the entire month of January 2017, taken at the same time of day. The scans are taken at low tide, which means that the intertidal area is visible on the scans. The output data is in xyz coordinates and each data point consist of a horizontal angle, a vertical angle and the range to the measured point.

4.1.1 Visualisation and structuring

After a point cloud has been derived from the laser scanner, the point cloud is often visualised to check whether the complete object that was being scanned was recorded. An image of a point cloud efficiently shows the large amount of information the point cloud contains (Vosselman and Maas 2010). Figure 8 shows the point cloud obtained by the scanner on the 1st of January 2017, visualised using CloudCompare.

Although the visualisation of point clouds is useful to get a first idea of the scanned area, further processing is necessary to extract the information contained in the point cloud. Point clouds are delivered as lists of x-, y- and z-coordinates, usually listed in the order in which the points were acquired by the scanner. Processing the data makes it possible to determine neighbouring points within a certain radius, and to assign these points to a certain object (Vosselman and Maas 2010).

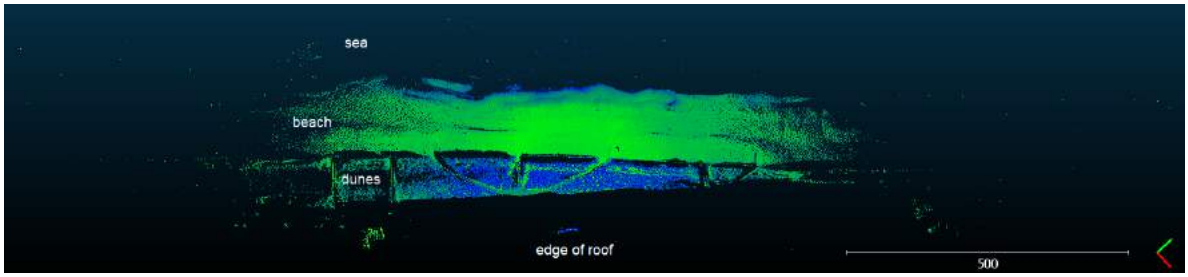


Figure 8: Visualisation of the point cloud captured by CoastScan on the 1st of January 2017 (top view)

4.1.2 Downsampling

In order to obtain usable datasets the pointclouds were downsampled. Downsampling of the point cloud is also useful because it shortens the computation time significantly.

The CoastScan datasets were downsampled using the Matlab function `pcdownsample`, which downsamples 3D point clouds using a box grid filter. The grid step specifies the size of a 3D box, and points within the same box are merged to a single point in the output. The properties of the points are averaged and the shape of the point cloud is preserved (MATLAB 2018).

The CoastScan dataset was downsampled using a grid step of 0.5 m. The number of data points in the original point cloud and the number of data points in the downsampled point cloud are shown in table 1 for the first 10 days of January. The low percentage of preserved points on the 9th of January is remarkable, but can be explained as apparently the original data points were all very close together. This is discussed further in the discussion.

Day	Original size	Downsampled size	Percentage Left
1	2 692 693	254 731	9.46
2	2 724 873	262 000	9.62
3	2 705 369	258 702	9.56
4	2 687 163	226 985	8.45
5	2 762 155	289 540	10.5
6	2 782 446	307 549	11.1
7	2 252 992	129 363	5.74
8	2 722 933	280 697	10.3
9	2 223 452	41 816	1.88
10	2 697 960	261 050	9.68

Table 1: Number of data points in the original point cloud vs. number of points in the downsampled point cloud for the first 10 days of January

4.1.3 Data Point Selection

A number of data points is specified and then extracted from the point cloud using the Matlab function `datasample`. `Datasample` returns the specified number of observations sampled uniformly at random (MATLAB 2018). `Datasample` is applied on the point cloud for the first of January, the coordinates of the sampled points are saved and used to create time series. The number of data points specified is 10 000, because the original shape of the point cloud is preserved, and further computations do not take too long.

Figure 9 shows a visualisation of the 10 000 data points compared to a visualisation of the original point cloud.

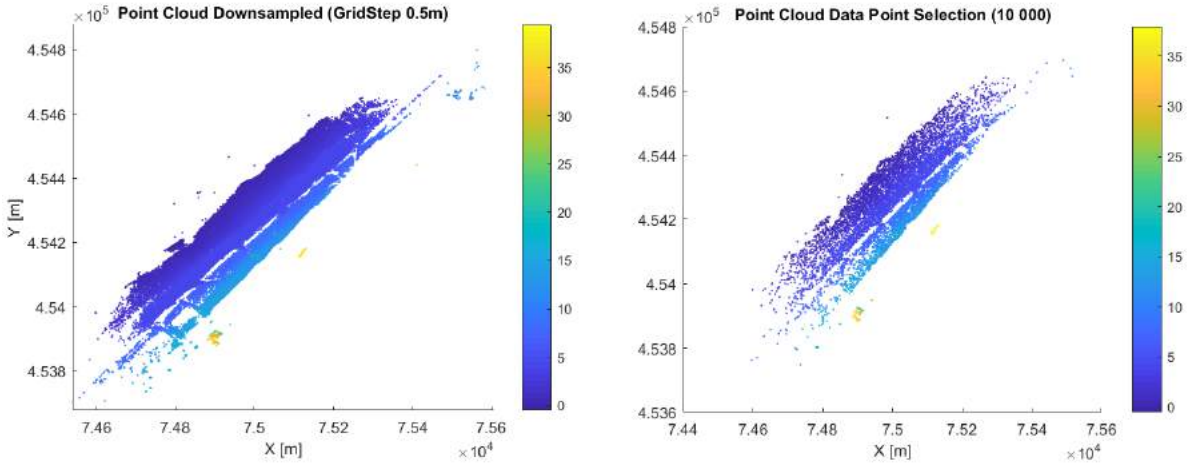


Figure 9: Comparison between the downsampled point cloud and the 10 000 sample points (dimension colourbar is in [m])

4.2 Time Series

After the point cloud data is prepared, time series can be created. A time series is a collection of observations listed in chronological order (Huang et al. 2016). A time series is usually taken at successive points, equally spaced in time. In the case of this thesis, the time series is 31 days long with one measurement per day.

4.2.1 Interpolating

Although the laser scanner always scans the same area, it does not measure exactly the same points every day. Therefore, it is possible that a data point of the 1st of January is not present on the 2nd of January, but it is present again sometime later in the month. When creating a time series, a data point for every day is desirable, to avoid gaps in the time series. Therefore, a region of interest (ROI) is defined for every data point.

The format of the ROI is a 3x2 matrix containing the 3D bounding box (MATLAB 2018). For this thesis, the ROI was reset for every point at the x and y value minus or plus 5 meters, and the z at minus or plus infinity. This creates a box around every point. This value was chosen in order to keep the number of complete time series high. A smaller distance would reduce the number of complete time series, but it would have a positive effect on the quality of the time series. Small height differences may disappear because using 5 m creates quite a big area over which the mean is taken.

Using the now defined ROI, the Matlab function `findPointsInROI` is applied. This function returns the linear indices of the points within the ROI, stored in a column vector (MATLAB 2018). The z-values over the full 31 days are then computed using not the exact z-value of the specified data points, but using the mean z-value of all the points within their respective ROIs. The output is an array of 31 rows, because there are 31 days in January, where the number of columns is equal to the specified number of data points.

4.2.2 Plotting

Next, the resulting time series can be plotted, either separately or together. The days are on the x-axis and the height of the point (the z-value) on the y-axis.

Days where there is missing data will not be included in the time series, and the time series is interrupted during those days. An example of two time series can be seen in figure 10 for a dune path where data is expected every day, and a sea point, where data is expected to be scarce. This illustrates that not all time series contain enough information for further analysis.

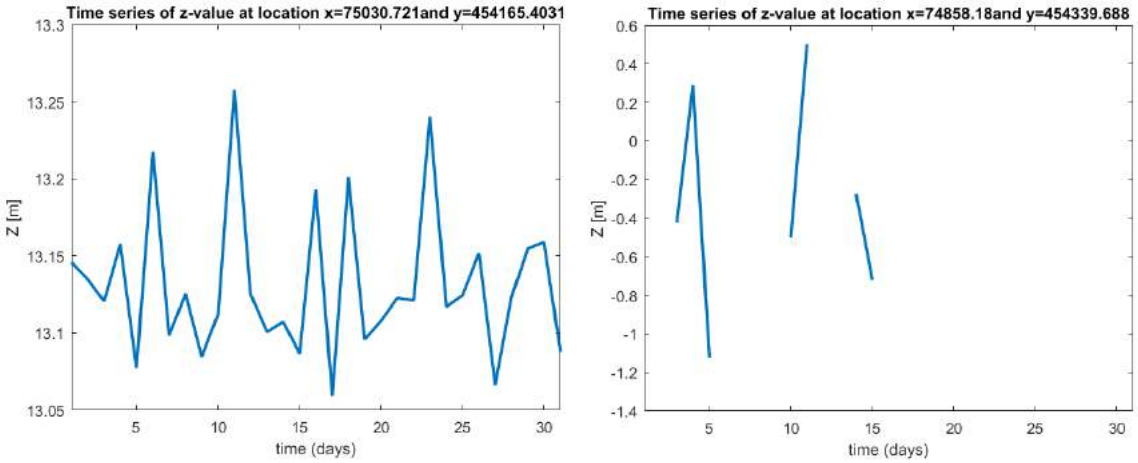


Figure 10: Comparison between the quality of time series, left a location on a dune path and right a sea location.

Although the missing data is not a problem when plotting time series, it becomes problematic in further steps. Therefore, all time series containing at least one NaN (Not a Number) entry are removed from the data point selection.

When all of the 10 000 time series are plotted in one window (see figure 11), it becomes apparent that something must have happened on the 9th of January, as a lot of points do not have data on that day. In order to prevent problems during k-means interpolating, all data from the 9th of January is removed. There is also a clear divide visible between the lower time series and the higher time series. The lower points are located on the beach and the dunes, and are the points interesting to this thesis. The higher points are either flagpoles or data points from the roof of the building, and are not useful for the classification. Therefore, z-values higher than 21 m are set to NaN and removed.

After these selections, 8 251 locations with complete time series were left, out of the original 10 000 locations. The point cloud of these 8 251 locations is visualised in figure 12. A gentle downward gradient towards the sea can be seen, because the height differences are a lot smaller. Furthermore, data points around the edges of the beach and the few scattered sea points have been removed.

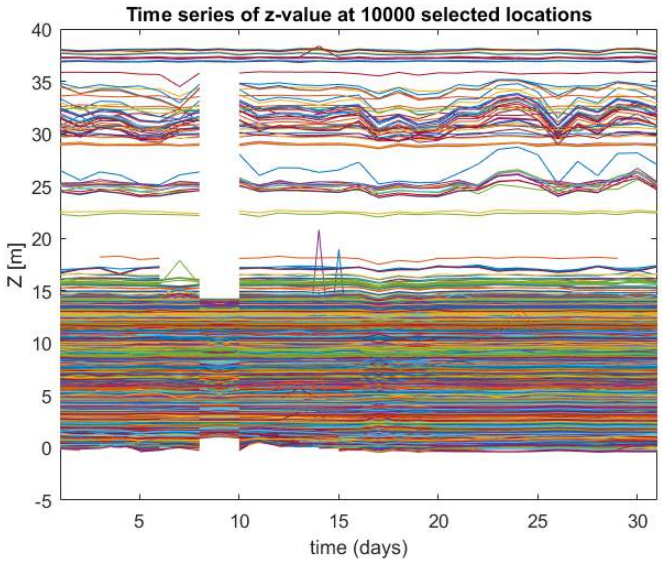


Figure 11: Graph showing all 10 000 time series plotted in one window, there is a clear gap between after 16-17 m, which are the highest dune points.

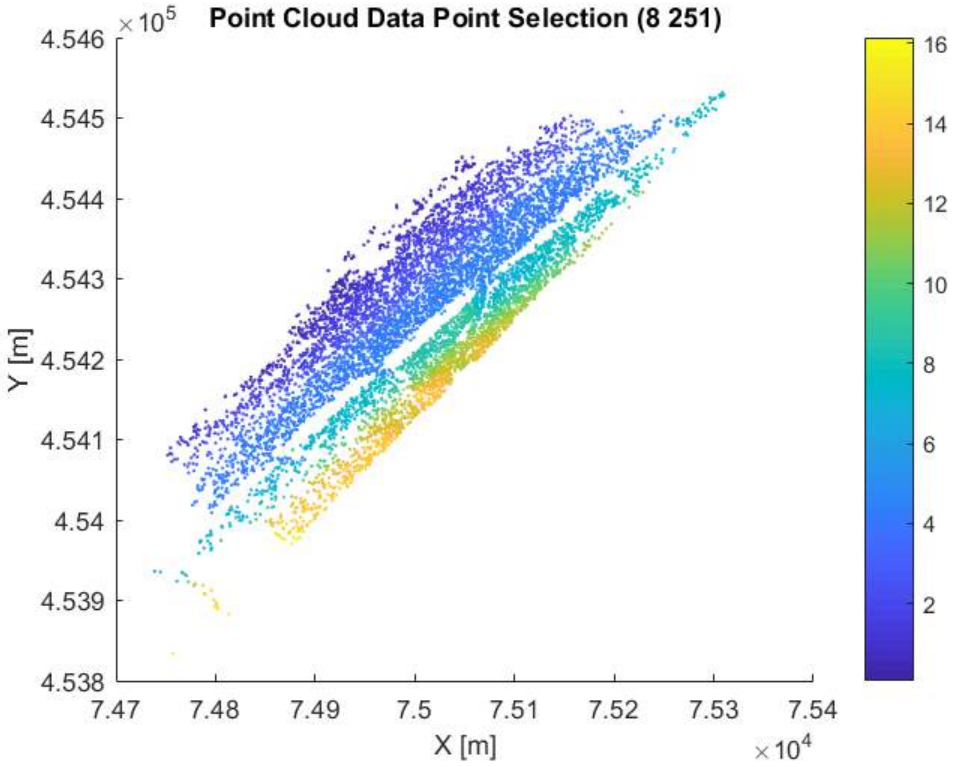


Figure 12: Point cloud of the 8251 remaining data points, the gradient is now gradual (dimension colourbar is in [m])

4.3 Classification

Point cloud classification is the process of assigning classes to individual points (Geography 2018). There are two main types of classification methods, namely pixel-based and object-based. Pixel-based classification methods can then be split into supervised and unsupervised classification. The classification method used in this thesis is unsupervised classification, but supervised clustering will also be briefly explained to highlight the differences.

4.3.1 Supervised Classification

In supervised classification, the user supervises the pixel classification process by providing training data. This means that before creating a classification, different classes are created. For example, in this beach setting, classes for asphalt, wet sand, dry sand, and vegetation may be specified. A number of representative sample points are then specified to belong to a certain class, these points are called training sites. This shows the program the kind of data points that belong to the class. The program then analyses the unspecified data points and subdivides these points into one of the classes based on their spectral signatures to classify the entire image. Common supervised classification algorithms are maximum likelihood and minimum-distance classification (Geography 2018).

Figure 13 shows the steps taken in supervised classification.

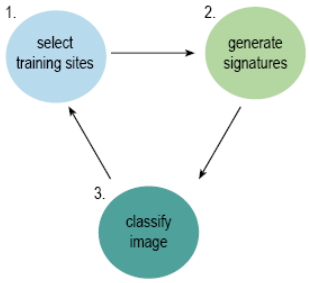


Figure 13: Supervised classification diagram (based on (Geography 2018))

4.3.2 Unsupervised Classification

Unsupervised classification, also referred to as clustering, is an effective method of dividing data in groups of pixels sharing the same characteristics. These groups of pixels are also called clusters. In unsupervised learning, no training data is provided. Instead, the program uses an algorithm to find similarities in the data and to group similar points together into clusters. After the clusters have been made, the user can manually identify each cluster with land cover classes (Geography 2018).

The user can choose the number of groups the program should generate. The user could choose to create a certain number of groups to create a classification, and then run the program again with another number of groups, and then use the difference between the two classifications to see which classification is best.

Unsupervised classification is an unbiased method and may therefore show unexpected features of the data that would not have been found in a supervised classification. Figure 14 shows the steps taken in unsupervised classification.

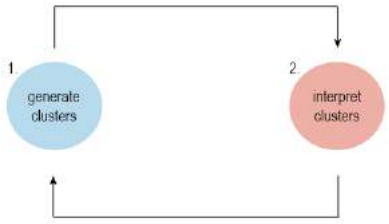


Figure 14: Unsupervised classification diagram (based on (Geography 2018))

4.4 K-means Clustering

K-means clustering is a type of unsupervised learning, which is used when the data set is unlabeled, so without previously defined groups. It is a simple, unbiased method that provides information on similar data points within a point cloud (Poortengen 2017). K-means clustering aims to separate n observations into k number of clusters S, in which each observation belongs to the cluster with the nearest mean. The variance within the cluster S is minimal (Poortengen 2017).

K-means clustering is an iterative procedure where each observation is assigned to the cluster which mean has the least squared Euclidean distance. With these new clusters the new mean of those clusters is then calculated and observations are re-assigned to the closest cluster. K-means clustering is finished when the observations no longer change cluster (Poortengen 2017), see figure 15.

The results of the k-means clustering are (Trevino 2016):

- The centroids (mean) of the k clusters, which can then be used to label new data
- Labels for the training data (each data point is assigned to a single cluster)

After the k-means clustering algorithm has finished and all data points have been assigned to a cluster, the user can identify what kind of process or land cover each cluster represents (Trevino 2016).

The algorithm inputs are the k number of clusters S and the data set consisting of n observations. The algorithm starts with initial estimates for the k centroids, either randomly generated or randomly selected from the data. The algorithm then iterates between two steps:

- Data assignment
- Centroid update

In the data assignment step, each data point is assigned to a cluster based on the squared Euclidean distance. The nearest mean is the cluster whose mean has the least squared Euclidean distance. Mathematically speaking: where c_i is the collection of centroids in set C, each data point x is assigned to exactly one cluster S_i based on where dist is the standard Euclidean distance (L2) (Trevino 2016).

$$argmin_{c_i \in C} dist(c_i, x)^2 \tag{4.1}$$

In the centroid update step the centroids are computed again by taking the mean of all data points assigned to that centroid's cluster. Mathematically speaking: (Trevino 2016).

$$c_i = \frac{1}{|S_i|} \sum_{x_i \in S_i} x_i \tag{4.2}$$

The algorithm then iterates between these two steps until data points stop changing clusters. The algorithm always converges to a result, but a new run might give another (better) result, as the algorithm starts with randomised starting centroids (Trevino 2016). The algorithm finds the clusters

for a particular pre-chosen k. To find the number of clusters in the data, the k-means clustering algorithm should be run for a range of different k values and the results should then be compared to determine the best k value. The algorithm is generally very fast, so it is not a problem to run it multiple times. By default, Matlab uses the squared Euclidean distance metric when performing k-means clustering (MATLAB 2018).

Figure 15 shows a simple example of how the k-means clustering algorithm is applied on a data set with k=2. The algorithm converges to a solution after a number of iterations, splitting the data in two clusters based on location.

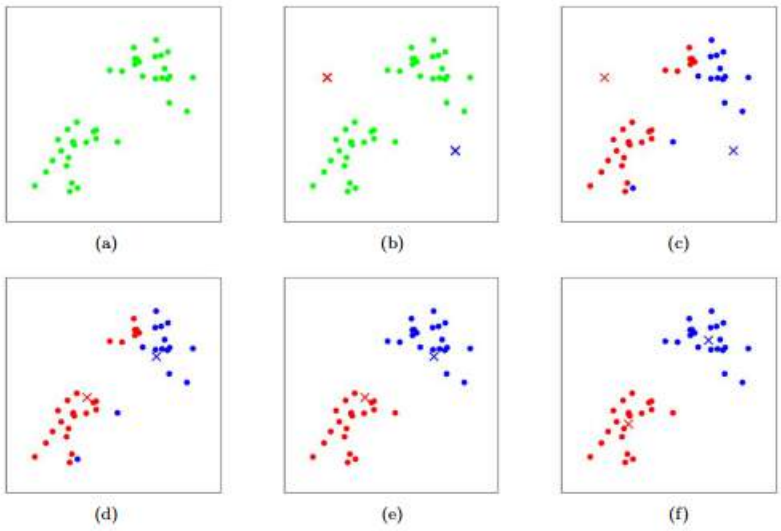


Figure 15: K-means clustering. Training examples are shown as dots, and cluster centroids are shown as crosses. (a) shows the original dataset, (b) shows the random initial centroid locations, (c)-(f) show two iterations (after (Piech 2013))

4.4.1 Advantages and Disadvantages

The advantage of k-means clustering is that the method is unbiased, and therefore the clustering allows the user to find and analyse the clusters without prejudice. Because of this, unexpected groups can be found of which the existence was not expected. Another interesting possibility is checking if a data point changes groups over time. However, k-means clustering also has disadvantages. If the wrong k is chosen, the results may be very poor. Also, k-means clustering tends to produce equal-sized clusters, which may not be the true case (Trevino 2016). K-means clustering works especially well if the clusters are spherical and the spread of the clusters is similar.

The biggest problem is that if the wrong initial values are chosen, the result can be very inaccurate. In order to mitigate this, k-means++ can be used. K-means++ is an algorithm for choosing the initial values for the k-means clustering algorithm. Although the initial selection in the algorithm takes extra time, the k-means part itself converges very quickly and then the algorithm actually lowers computation time (Arthur and Vassilvitskii 2007). By default, Matlab's k-means implementation uses the k-means++ algorithm for cluster center initialisation (MATLAB 2018).

4.4.2 Input

K-means clustering is often explained in a 2D-setting, but it can be applied on data containing more dimensions.

In this thesis, the input exists of a set of time series, where each time series consists of 30 z-values. The goal is that the k-means algorithm then divides these time series into clusters based on their height and shape. The location of the time series is not part of the input to avoid the time series being clustered based on their location.

4.4.3 Replicates

Matlab allows the user to specify a number of times to repeat the clustering process using new initial cluster centroid positions. The number of times to repeat the process is specified as the number of replicates to make. The `kmeans` command then returns the solution where the within-cluster sums of point-to-centroid distances is the smallest (MATLAB 2018). This may be the first solution, but it does not have to be. In this thesis, the script is run with the default replicate value of 1 and with a replicate value of 5.

4.4.4 Choosing K

In order to give a good result, a proper value for k should be used. Theoretically, a higher amount of clusters will reduce the amount of error in the clustering; as there are more clusters, an observation is more likely to be a good fit for one of those clusters instead of an intermediate between clusters. However, after a certain point, adding more clusters does not give a much better result.

The ideal k can be found using Matlab using the `eval` command. `Eval` evaluates clustering solutions based on different criteria. The criterion used in this thesis is the Davies Bouldin criterion.

The Davies-Bouldin criterion computes for each cluster how similar that cluster is to all other clusters. The cluster similarity is then defined to be the highest of these values. Ideally, this cluster similarity is low, because that would mean that even the most similar other cluster is not similar to the investigated cluster. The Davies Bouldin index is found by averaging all cluster similarities. The smaller this index, the better the clustering result, as that means that the clusters are less similar to each other (Liu et al. 2010).

This criterion is chosen because it uses the similarities of the clusters, which is a good measure to determine the quality for time series.

In this thesis, the theoretical ideal k is determined through Matlab's `eval` function using the silhouette criterion, but the resulting clustered time series are also evaluated with the eye.

4.5 Interpreting Time Series

The K-means clustering results are interpreted with the aim of recognising different deformation regimes in the time series. In order to simplify the interpretation, the beach is divided in three main sections (see figure 16): the intertidal zone, the backshore and the dunes. Figure 16 also shows the envelope around the mean profile, and thus the degree of irregularity that is to be expected in the time series. The heights in this image per section can be disregarded.

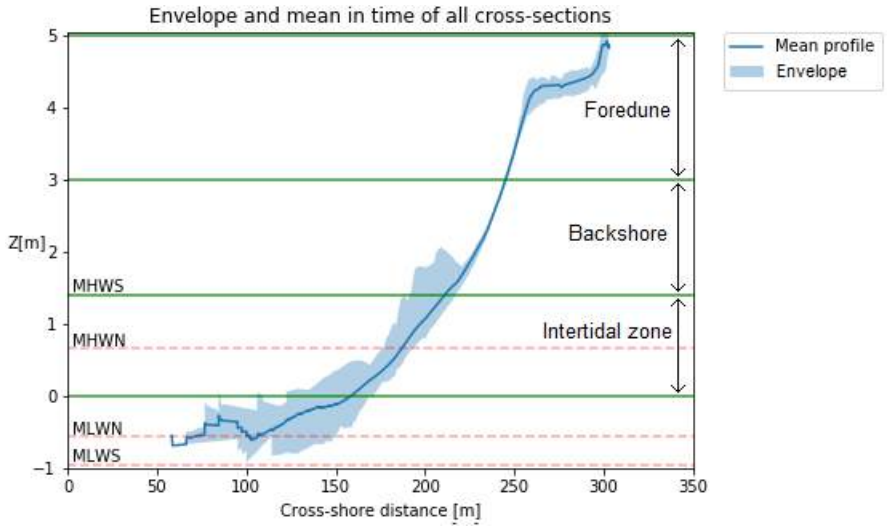


Figure 16: Mean in time and envelope of all cross-sections. This image shows the different zones and the degree of irregularity that is to be expected. (After (Hobbelen 2018))

4.6 Height-based Clustering vs. Shape-based Clustering

The time series that were created can be clustered including the heights of the time series or excluding the height.

To remove the heights, the mean height of every time series was calculated and subsequently subtracted from every data point of the corresponding time series. An example is shown in figure 17, demonstrating that the shape of the time series remains the same.

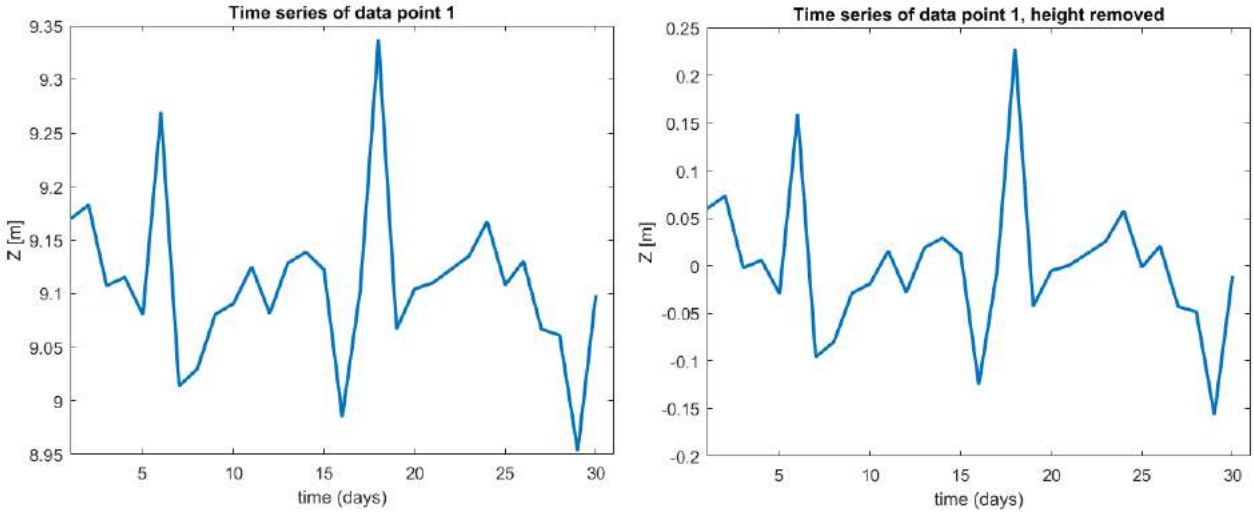


Figure 17: Left: Time series of data point 1 at its original height. Right: Time series of data point 1 with height-removed.

The k-means clustering solution is expected to be different. In the height-based clustering, the k-means algorithm input contains both height and shape information, and the algorithm is expected

to create clusters largely based on height. This is expected because the height-data drowns out the shape, especially if the number of clusters k is low. If k is higher, the clustering solution will still be partly based on height, but the shape will play a bigger role: two clusters may form with time series of the same height, but of different shapes.

If the time series are clustered excluding height, the relative height difference throughout the time series is the most important factor. This can also be seen as the shape of the time series, which is why it is referenced as shape-based clustering. Shape-based clustering is interesting because it may lead to different and unexpected conclusions than the height-based clustering solution. For example, time series in the inter-tidal zone may belong to the same cluster as time series located in the dunes because a common process governs the height differences throughout the month. Such an occurrence would likely not be found in height-based clustering clustering.

4.7 Software

4.7.1 CloudCompare

CloudCompare is an open source 3D point cloud editing and processing software. CloudCompare has several functions and plugins that can be used for visualising and editing point cloud data. The data provided was saved in a .xyz format, but Matlab can only read .pcd or .ply files into point clouds. Therefore, the data had to be converted, which was done using CloudCompare.

4.7.2 Matlab

Matlab is used to prepare the point cloud data for analysing the data. The data is further prepared, then time-series are created, and then k-means interpolating is applied and assessed. The Matlab script used for this thesis can be found in Appendix A.

5 Results

This chapter elaborates on the results found using the methods described in chapter 4. It should be noted that due to the algorithm of k-means clustering the results will never be exactly the same. All the figures in this chapter were created using the same clustering results, and should therefore be representative.

5.1 Height-based clustering

First, k-means clustering is applied to the data set including heights. Therefore, the clustering is largely based on the z-values of the points.

5.1.1 Number of Clusters & Point Distribution

All k values ranging from 2 to 10 have been tried and assessed. It can be assumed that the results will be of a higher quality using 5 replicates rather than only 1 replicate, therefore all computations will be performed using 5 replicates. However, the groups will not be the same because in some cases the first replicate may be optimal, while in other cases the second or the third etc. Therefore, group 1 for k=5 may suddenly be group 4 for k=6, and subsequently be plotted in a different colour. This is not a problem, but it must be taken into account when evaluating the resulting images.

Using the Matlab command `evalclusters` to evaluate clustering solutions based on the Davies Bouldin criterion, an optimal number of four clusters was found. Using only four clusters, not a lot of detail can be seen, and the clusters seem to be mainly based on altitude rather than how a point behaves over time. The time series can be seen in figure 18. The resulting clustered image can be seen in figure 19, along with the rough location in satellite image for comparison. Although this image looks as expected, different deformation regimes cannot be identified. Therefore this k is not deemed to be the most ideal for this thesis.

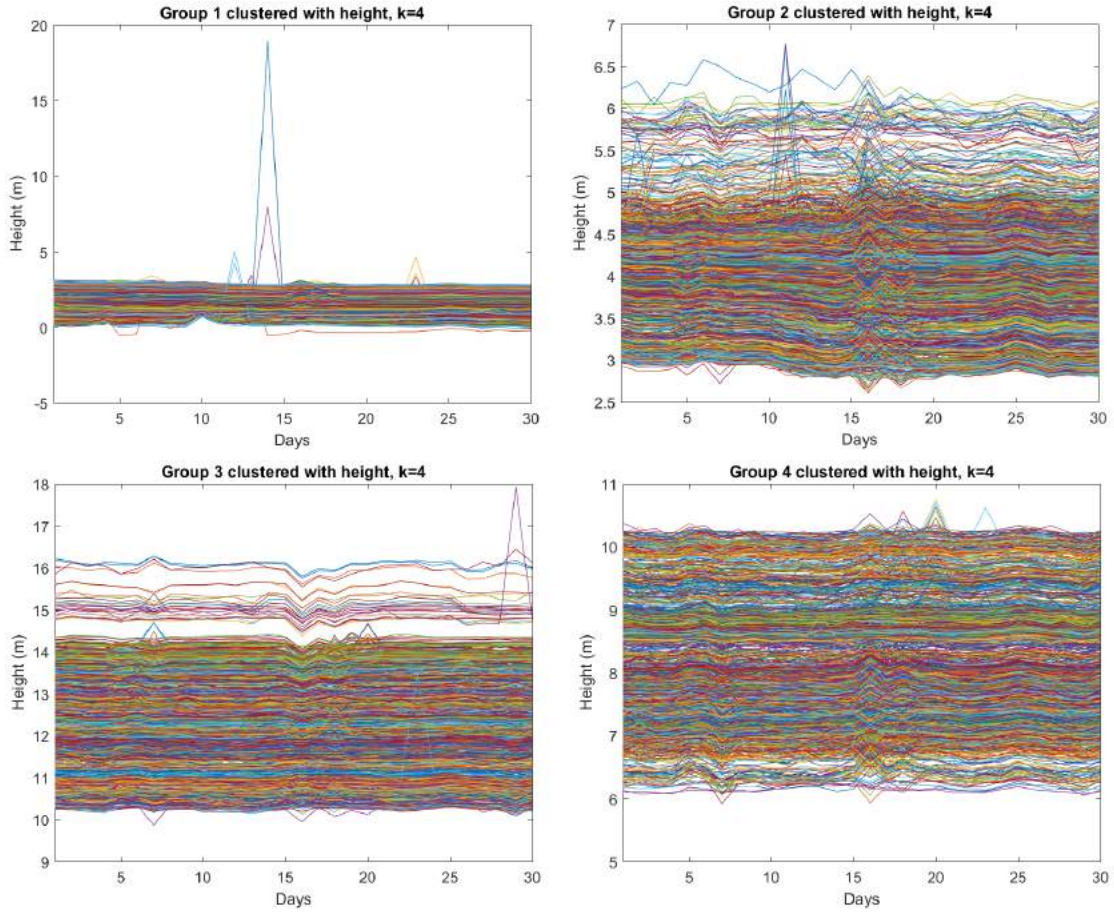


Figure 18: The time series of clusters for k=4 show that the clustering is only based on height, different deformation regimes are very hard to distinguish

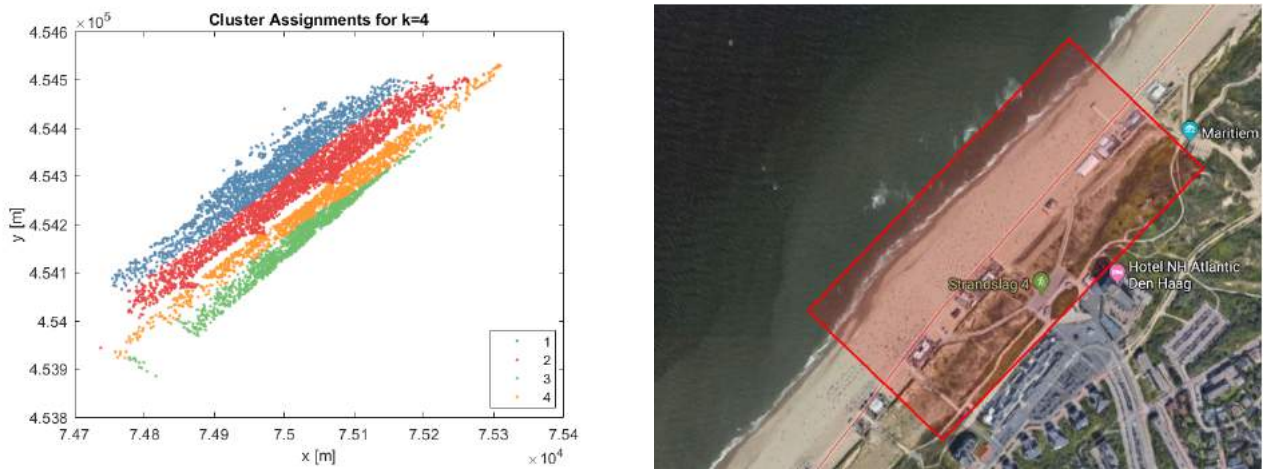


Figure 19: Left: Visualisation of the data clustered into 4 groups. Right: Satellite image of Kijkduin beach, the location of the scanned area is highlighted (via Google Maps).

Figure 20 shows the course of the Davies Bouldin value. The Davies Bouldin value is optimal for k=4 and the quality then declines. However, as the results using k=4 are deemed not to be ideal for this thesis, the results computed using a range of k-values were visually compared. The results found using k=10 were deemed to be the best for this thesis, and as such k=10 was used.

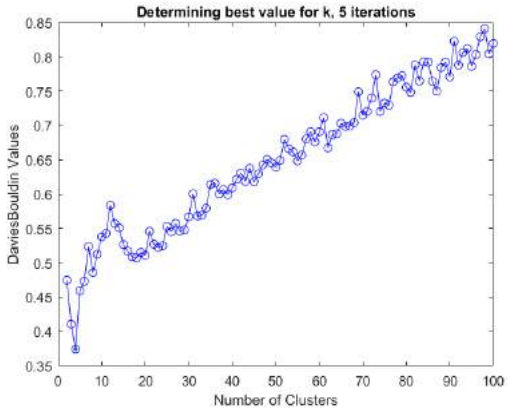


Figure 20: The Davies Bouldin criterion returns an ideal k value of 4.

The distribution of points for a k of 10 and 5 replicates is as shown in table 2.

Table 2: Point distribution per group for k=20, 5 replicates

	no. of points
group 1	2355
group 2	667
group 3	424
group 4	952
group 5	423
group 6	786
group 7	601
group 8	925
group 9	676
group 10	442

5.1.2 Clustered Beach

The data points were plotted according to their assigned clusters, resulting in figure 21. The beach is divided in strokes parallel to the coast, until the dunes are reached, from where the groups are more scattered, and not necessarily connected. Compared to figure 19, more detail can be seen in the dune area especially.

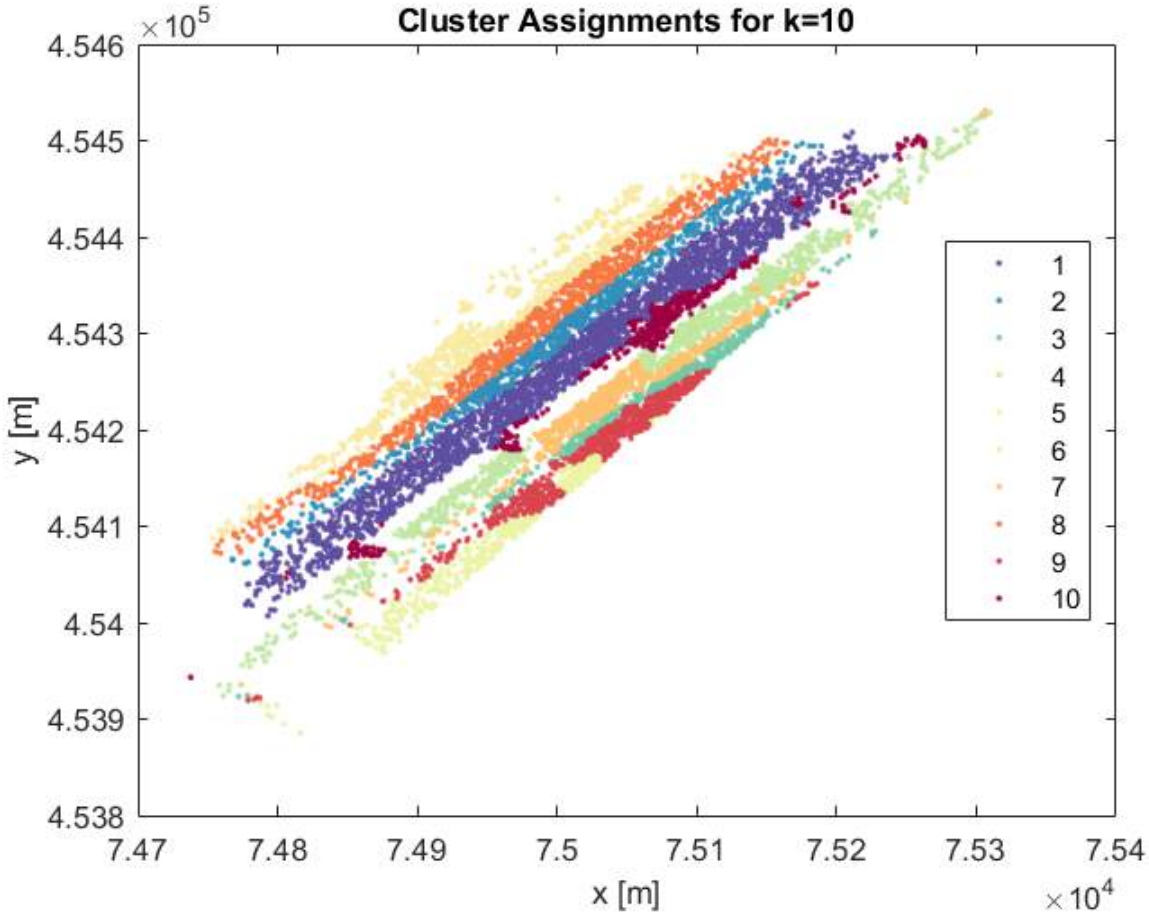


Figure 21: Visualisation of the data clustered into 10 groups

5.1.3 Mean Time Series Per Cluster

In order to better visualise the difference between the different groups, the mean time series of every group is computed and all mean time series are plotted in one window. It is apparent that the clusters are largely based on height.

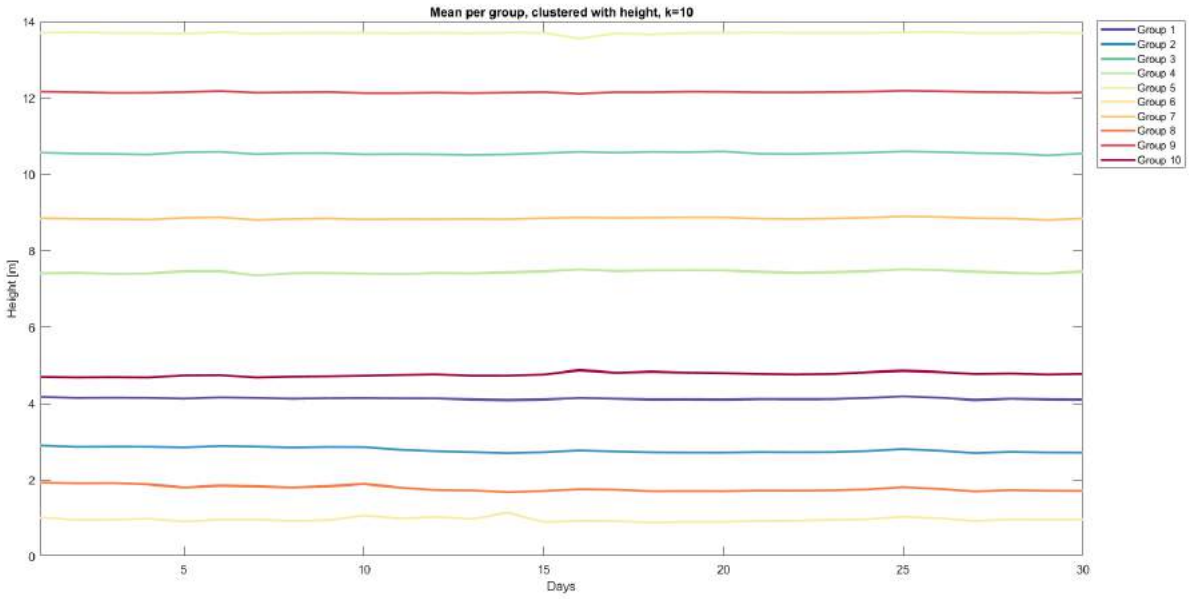


Figure 22: Mean of all 10 groups plotted in one window. Please refer to appendix B for more detailed figures of the mean time series.

5.1.4 Cluster Interpretation

In order to interpret the clusters the mean time series as well as the combined time series were analysed (see Appendix B) and the clustered image of the beach was compared with Google Maps satellite images and field observations (see Appendix D). The beach is first divided in different sections, using Rik van Hobbelen's work on the same beach and Google Maps. This division can be seen in figure 23. In each of these sections different deformation regimes are to be expected, and predefining these sections makes it easier to know what to look for. The divisions were made based on Google Map satellite images and largely follows the shape of the clusters.

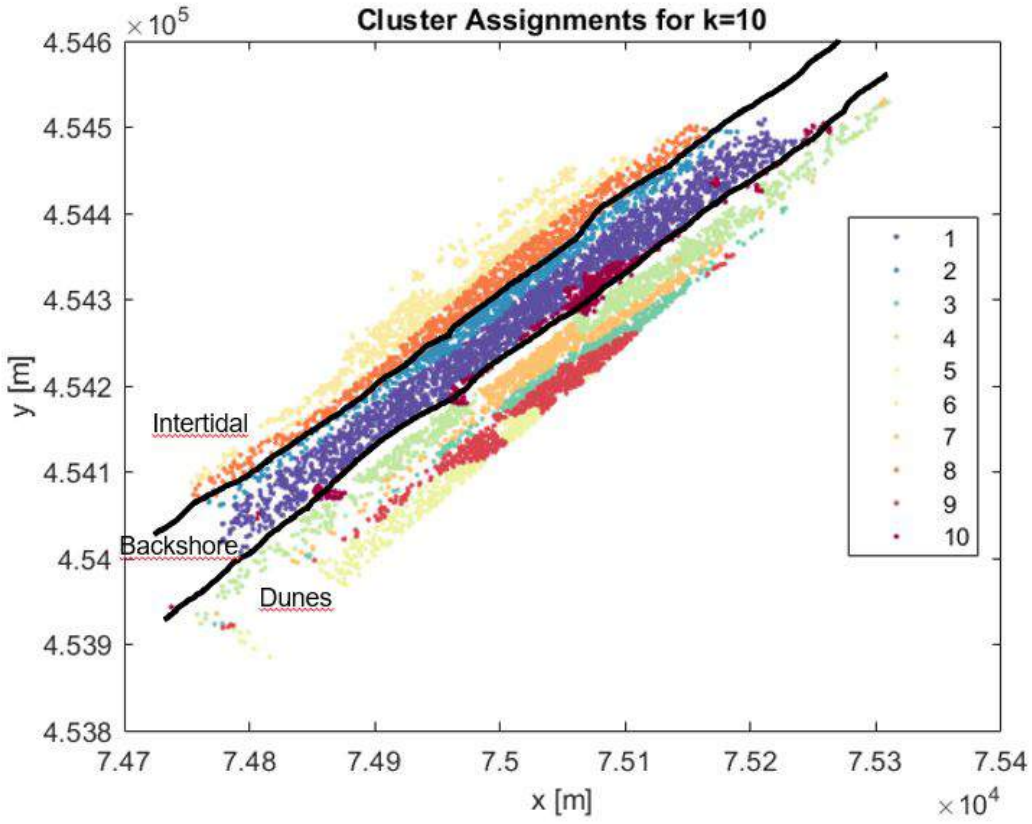


Figure 23: The clustered image of the beach has been divided in three sections: intertidal area, backshore, and dunes. In each of these sections different deformation regimes are to be expected.

This led to the following interpretations, starting closest to the sea and then moving land inwards:

Intertidal zone

In the intertidal zone the highest degree of irregularity is expected using figure 16.

- Group 6: This group is located closest to the sea. The height difference is very small, the height is mostly constant. Due to its location in the intertidal zone, a lot of activity was expected. A possible explanation of the lack thereof is that this area is deeper underwater at high-tide, and the surface may therefore be less affected by waves.

- Group 8: This group is located slightly further from the sea and is interpreted to be the end of the intertidal zone. The overall trend is downwards, but the height varies a lot, moving up and down. The height differences are interpreted to be caused by a combination of waves and tides working together to move sand bars towards the dunes. Another explanation could be aeolian processes blowing dried sand further land inwards during low-tide.

Backshore

The degree of irregularity is expected to be low in the backshore, although the beginning of the backshore may still show quite a lot of variability using figure 16.

- Group 2: This group's behaviour is similar to group 8, but the peaks are smaller. This group may

still be slightly affected by the tides, but less so than group 8. Aeolian processes are expected to be the main cause of deformation.

- Group 1: This group is the most stable group located on the beach. Deformation is mostly caused by aeolian processes or through human intervention (ie footsteps, tire tracks).

- Group 10: This group is located mainly around the beach entrances, and is interpreted to be concrete slabs. The overall height increases during the month, more gradually than the previous groups. This is interpreted to be due to aeolian processes blowing sand land inwards, and due to pedestrians that (inadvertedly) throw sand on the concrete when they exit the beach. The combined time series shows a lot of peaks and irregularities, which are interpreted to be temporary objects.

Dunes

The dunes are a mix of vegetation, sand and paths, and therefore more variation between the different time series is expected than in the other zones.

- Group 4: This group is located in the dunes and is closest to the sea of this zone. The overall height remains largely constant, but the combined time series show that the height varies throughout the month, more so than in group 7. This is interpreted to be vegetation moving because of the wind.

- Group 7: This group is located in between group 4. The overall height remains largely constant, and although the shape of the mean time series is very similar to group 4, the height differences are a lot smaller. This is interpreted to be vegetation. The mean height is higher; the dunes are higher and the vegetation is bigger. The height differences are smaller than in group 4 because the vegetation is less susceptible to wind.

- Group 3: This group is located close to the scanner. The overall height is largely constant, but varies throughout the month. Interpreted to be vegetation.

- Group 9: The overall height is constant, but the combined time series show a lot of irregularities. The group is interpreted to be asphalt, then these irregularities are most likely caused by pedestrians.

- Group 5: Height decreases suddenly halfway through the month, therefore this cannot be asphalt or another constant ground type. Other than that the time series is very constant. Because of this fact as well as the clusters location this group is interpreted to be the berm next to the asphalt.

Overall it was apparent that the second half of the month was more irregular than the first half of the month on the beach. This may be explained by a change in wind velocity, a storm occurred on the 13th/14th of January (Hobbelen 2018).

5.2 Shape-based Clustering

The preceding results are all based on the data set including heights, and therefore the height is also included during clustering. This leads to the aforementioned fact that different clusters behave the same, but are assigned to another cluster based on the height of the points. Therefore the height was taken out of the time series by finding the mean per time series and subtracting the mean from every data point per time series, resulting in time series around a height of 0. The k-means clustering algorithm was then applied again, now clustering mainly based on shape of the time series.

5.2.1 Number of Clusters & Point Distribution

The number of clusters was again evaluated using Matlab's evalclusters function. The resulting ideal k value was 2, with evalclusters judging the results to be considerably poorer than in height-based clustering, with an average silhouette value of

This can be explained by the method evalclusters uses to judge the quality of a clustering solution. As the height of the data points has been removed, the different time series are more similar to each other. This makes it harder for Matlab to distinguish between different clusters, and points assigned to one cluster are also very similar to another cluster.

As evalclusters did not give a good indication for k, the value used for height-based clustering will also be used for shape-based clustering: k=10.

Table 3 shows the point distribution for 10 clusters. Compared to 2 the group sizes vary more: there are now groups smaller than 100 locations, and the other groups are a lot bigger.

Table 3: Point distribution per group for k=10, 5 replicates

	no. of points
group 1	1486
group 2	10
group 3	2
group 4	1053
group 5	1199
group 6	692
group 7	2608
group 8	67
group 9	1113
group 10	21

5.2.2 Clustered Beach

The data points were again plotted according to their assigned clusters, resulting in figure 24. In comparison to figure 21, the groups are not parallel to the coast. Instead, they are more compact. Another difference is the fewer and larger groups that dominate the image, and the few small groups within these larger groups.

Because the clusters are no longer bound by their absolute height but by their relative height, other unexpected deformation regimes may be found when analysing and interpreting the clusters.

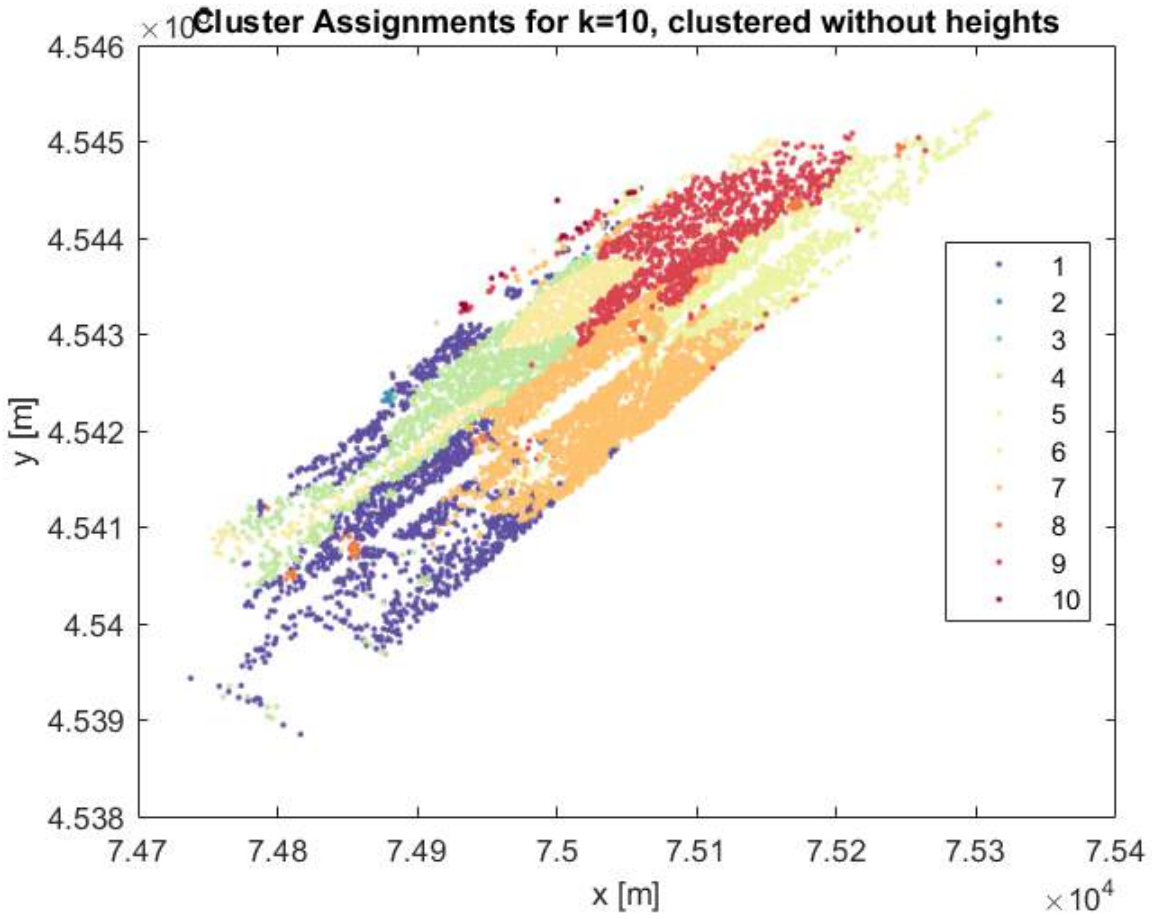


Figure 24: Visualisation of the data clustered into 10 groups

5.2.3 Mean Time Series Per Cluster

The mean time series per group were computed and plotted, resulting in figures 25 & 26. Figure 25 shows that the clusters are still somewhat based on height, but significantly less than before. Figure 26 shows the difference between the mean time series. The time series with higher peaks may be manually removed in Matlab to better see the difference between the mean time series, or the user can zoom in on the lower time series.

There is one group (group 2) with a peak at 18.96 m on day 15 (as day 9 was removed previously). This is remarkable, because the same data points were used to compute the height-based clustering solution, in which these data points were not present.

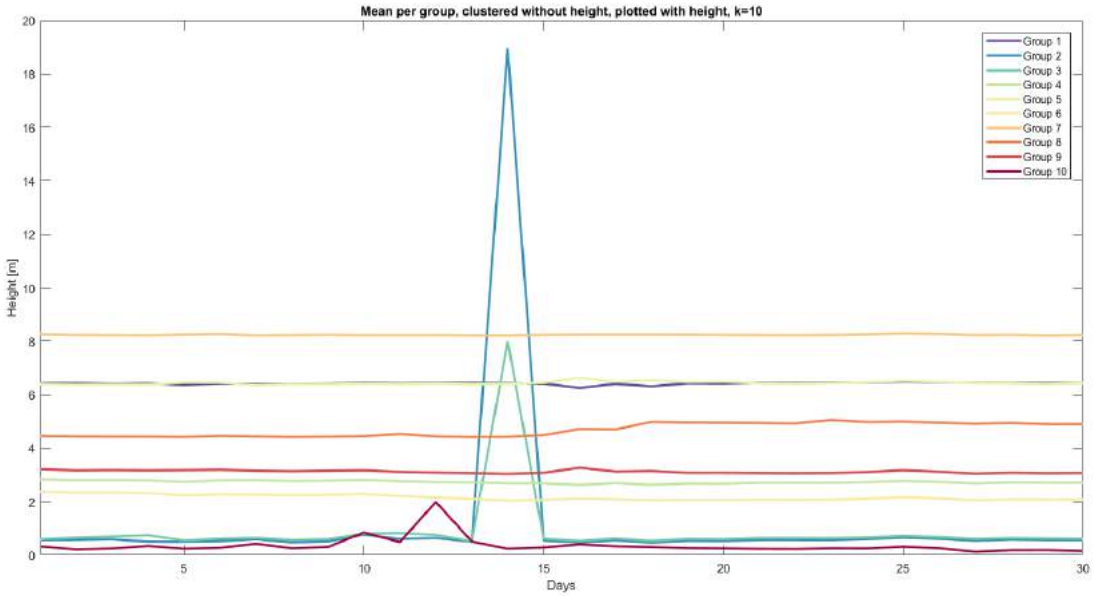


Figure 25: Mean of every group plotted in one window, at their original height. Please refer to appendix C for more detailed figures of the mean time series.

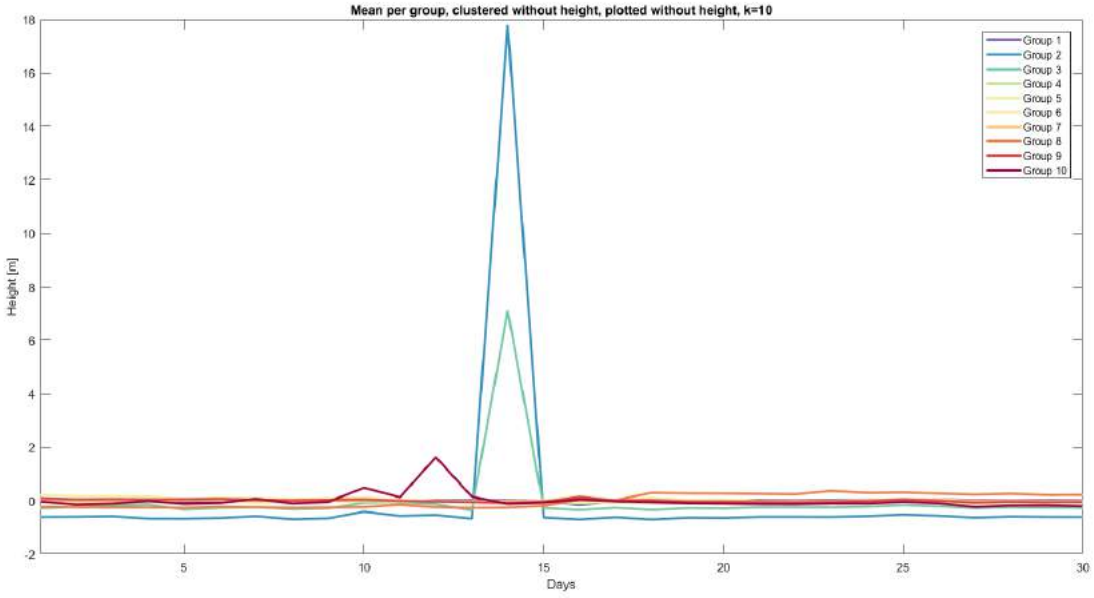


Figure 26: Mean of every group plotted in one window. Please refer to appendix C for more detailed figures of the mean time series.

5.2.4 Cluster Interpretation

In order to interpret the clusters the mean time series were analysed and the clustered image of the beach was compared with Google Maps satellite images and field observations (see Appendix D). The shape-based clustered image was divided into the same three sections previously used in the height-based clustering, see figure 27. It is clear that the different clusters are more spread out over the different zones, and the deformation regimes governing the different groups are less likely to be related to height. Therefore, new deformation regimes may be found.

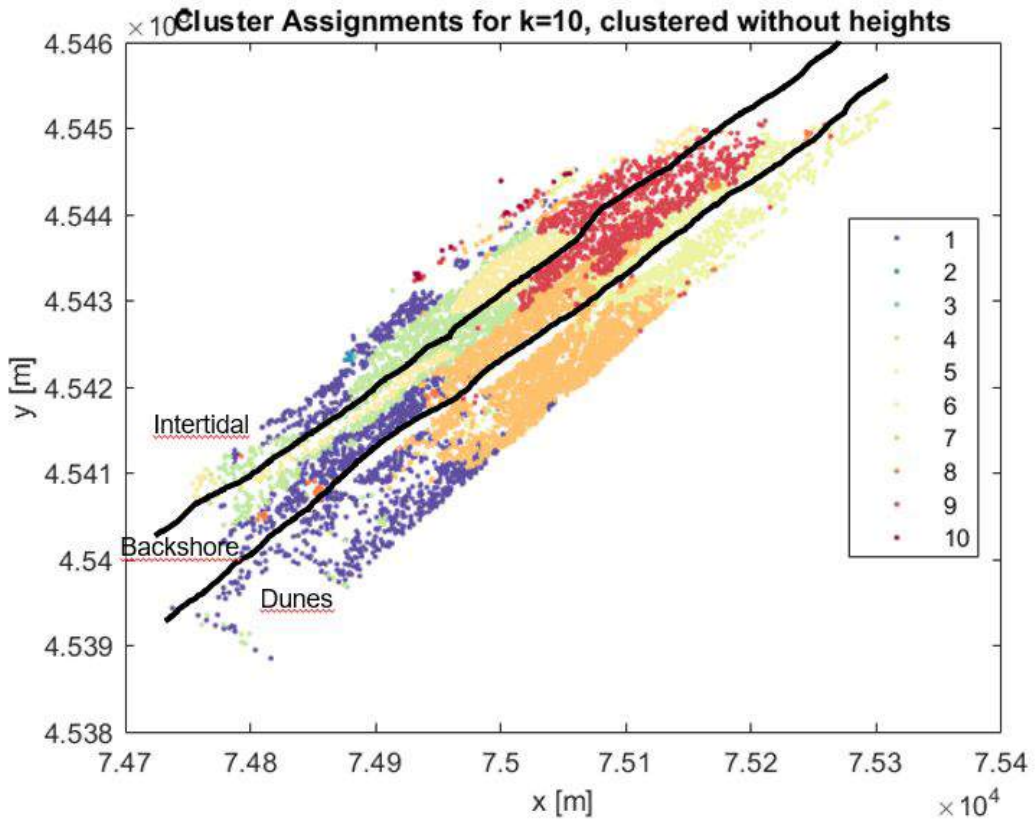


Figure 27: The shape-based clustered image of the beach divided into the same three sections: intertidal area, backshore, and dunes. Groups are no longer bound to one of these zones.

The clusters were divided into zones based on the shape of the mean time series: zone 1 is almost constant with one peak, zone 2 has a downward trend, and zone 3 has an upward trend. Group 1 could be interpreted to belong to two of these zones, because the shape of the time series is most similar to zone 2, but the overall trend is upwards. This can be explained by the location of the group, as it is present in the intertidal zone, in the backshore and in the dunes. Ultimately zone 3 was chosen because the combined time series fit better in this zone.

Zone 1

This zone consists of small clusters that are largely constant, except for one much higher peak. All groups are located close to the sea in the intertidal zone.

- Group 2: This group is located at the beginning of the intertidal zone, its height is constant, with the exception of one peak. This peak occurs on the same day as in group 3, but the peak is approximately 10 meters higher. The clusters are right next to each other. No explanation

for this peak could be found when examining the point cloud in CloudCompare. As such, the peak is interpreted to be caused by a temporary object such as a flying bird, or a kite, and the peak in group 3 is thought to be caused by the same object (or another bird).

- Group 3: See group 2.
- Group 10: The overall height is largely constant, but the time series is very irregular throughout the month. This is thought to be caused by swash.

Zone 2

This zone consists of clusters that have a downward trend throughout the month.

- Group 4: Group is located on the left side of the beach, close to the sea. Time series is irregular throughout the month, the total height difference is approximately equal to that of group 9, but the shape is different. This indicates that there are different processes at work. Due to the irregularity, the time series is interpreted to be part of the intertidal zone. Height differences are approximately 5 cm per day. The deformation regime at play is swash. The combined time series show a number of random peaks (ie they only occur in a few time series on that day), these could be caused by smaller deformation regimes, but in this case they are interpreted to be footsteps or tire tracks.
- Group 6: This group is located on the same stretch of beach as groups 4 and 9, but patches are scattered from the left side to the right side of the beach. Height difference is large, the height drops 20 cm within 5 days. Time series is overall irregular. Deformation regimes are swash and erosion, but these processes are more extreme here than in group 4. These locations may be more vulnerable to wind, or the wave energy may be higher.
- Group 9: There is a big peak halfway through the month, the height increases 20 cm in 2 days. The combined time series show that the height increases quickly everywhere, then decreases very irregularly. Therefore the deformation regime at play can cause a sudden height increase, but not a sudden height decrease.

Zone 3

This zone consists of clusters that have an upward trend throughout the month.

- Group 1: This group is found at the left side of the beach, with planes in the intertidal zone, on the beach and in the dunes. The shape of the mean time series is more similar to those of zone 2, but the overall trend is upwards, and as such it is assigned to zone 3. This is explained by the location of the group, as part of the group lies in the intertidal zone and is therefore expected to behave more like zone 2. However, its trend is upwards instead of downwards, and because of this it is interpreted to be a sand bar migrating towards the coast. This is caused by the waves and tides. The deformation regime changing the part of the cluster on the upper beach and in the dunes is thought to be aeolian and similar to group 7, based on the combined time series.
- Group 5: This group is located in the dunes at the right side. Both the mean time series and the combined time series are very variable, mainly throughout the second half of the month. This is interpreted to be the aftermath of a storm that occured on the 13th and 14th of January (Hobbelen 2018). The changes are likely caused by aeolian processes, which transports sand and moves vegetation.
- Group 7: This group is located in the dunes in the middle, between the middle beach entrance and the right beach entrance. Both the mean time series and the combined time series are very variable, throughout the entire month. The changes are likely caused by aeolian processes, which transports sand and moves vegetation. The group may be more variable than group 7 because its location is more susceptible to wind.

- Group 8: This group is located at the foot of the dunes, around the beach entrances, and the height is largely constant. As such, this group is interpreted to be concrete slabs (see figure 40 in appendix D). Height strongly increases around day 15, and then remains at that height. This is thought to be caused by strong winds blowing a lot of sand from the beach more land inwards, which then settled down at the foot of the dunes. This can only be seen around the beach entrances because of the scan shadow. The smaller height differences are likely to be caused by wind blowing sand onto or off of the concrete, and by pedestrians kicking sand on or off as they walk over the concrete.

These zones were then layered over figure 27, which resulted in figure 28. This image shows that although the predefined zones do not entirely correspond to the shape-based clustered image, the zones may still be relevant if they are moved somewhat. The backshore disappears and the image is divided into an intertidal zone and a dune zone. The two red circles corresponding to zone 1 are interpreted to be part of the intertidal zone, but clustered into a different zone because of temporary objects or errors. The part of group 1 that is in the intertidal zone is interpreted to be a migrating sand bar. Migrating sand bars could also be seen in the point cloud visualisations in CloudCompare, so it is deemed to be a plausible explanation.

Then, the predefined zones were removed and the image was coloured again following the interpretation, resulting in figure 29.

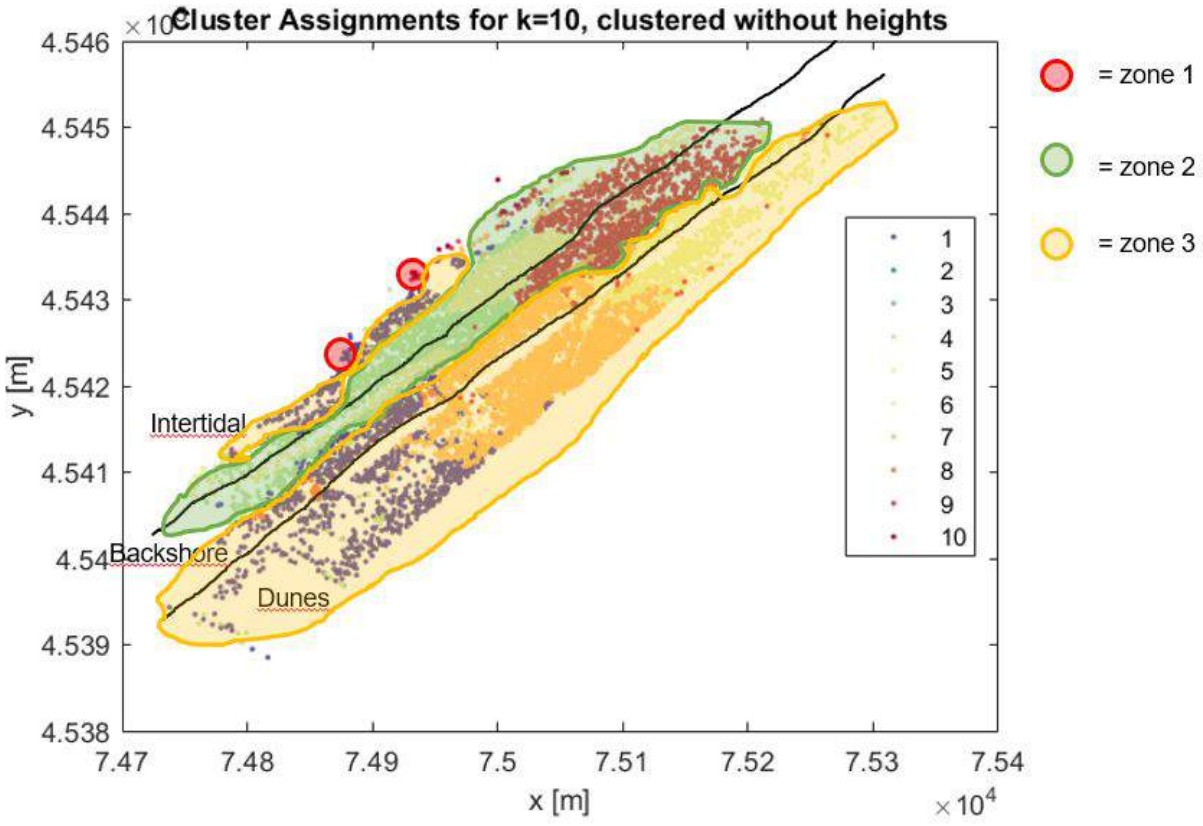


Figure 28: Figure 27 with the three zones resulting from the shape-based clustering plotted on top of it. This shows that the predefined zones are somewhat accurate.

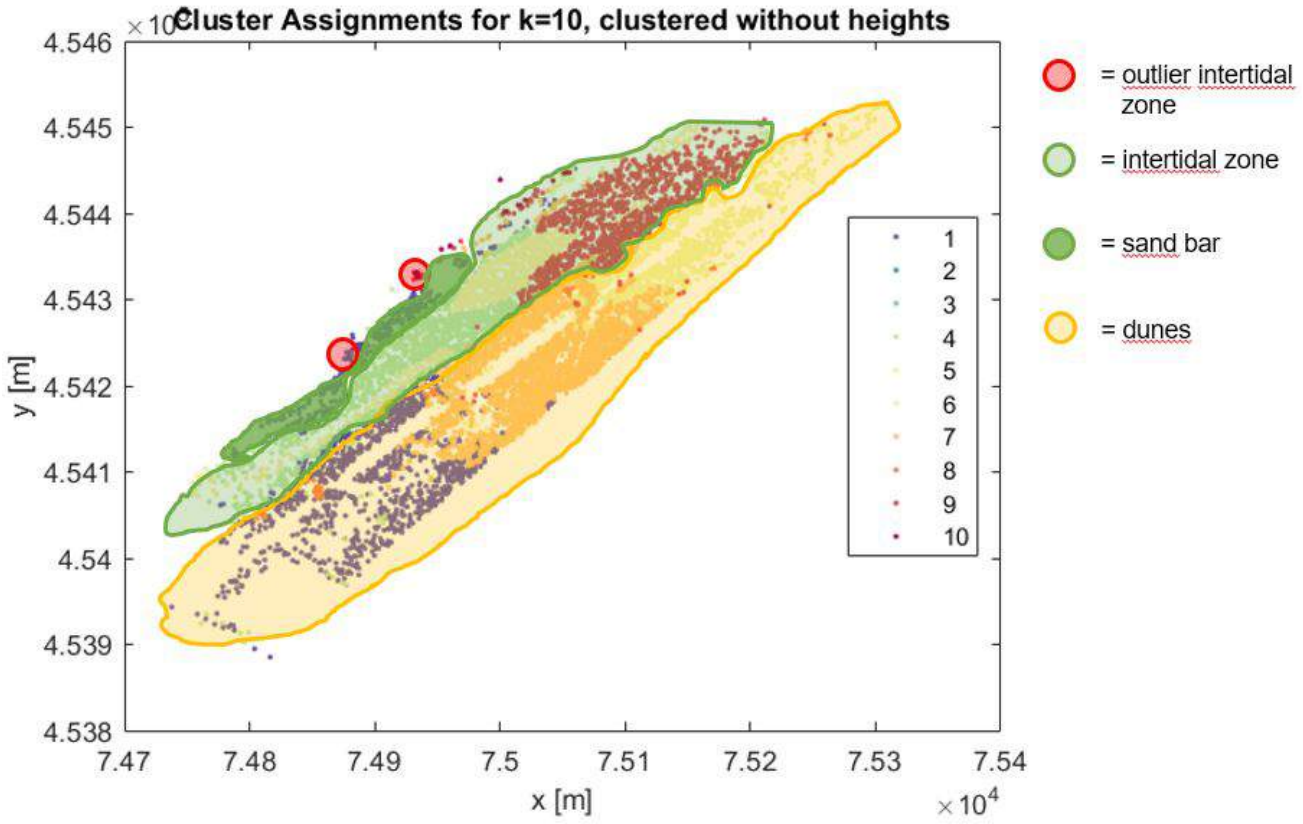


Figure 29: The interpretation of the clustered image after comparing the height-based and the shape-based results.

6 Discussion

6.1 Preprocessing

Multiple choices had to be made during the preprocessing of the data in Matlab, all affecting the quality of the results. These choices are highlighted here.

6.1.1 Morphological Filtering & Outliers

Morphological filtering was not applied in this thesis. There is a chance of using a point with a person or some other temporary obstacle. The time series will then be interrupted at that day and show a peak where a peak would not normally be expected. Variances of up to approximately 2 m are therefore expected throughout the time series, varying per point.

The effects of these temporary objects seem to be small, meaning that the point is unlikely to be assigned to the wrong cluster. It is important to note that this is also related to the distance used when defining the region of interest.

There are a number of outliers in the data, which may either be caused by temporary objects, such as birds, or errors. This influences the shape-based clustering more so than the height-based clustering.

6.1.2 Downsampling

In this thesis downsampling with a grid step of 0.5 m was chosen. This size is reasonable based on the large area of the scan as well as the high point density. A smaller grid step may give slightly better results, but it would also slow down Matlab.

6.1.3 Data Point Selection

In this thesis a number of 10 000 data points were selected, sampled uniformly. This data point selection is not essential. In fact, the entire downsampled data set can be used, but it slows down further computations significantly.

Defining the ROIs and subsequently computing the positions used to plot time series becomes an especially slow process; the elapsed time for 10 000 data points is 94 seconds (1.5 minute), for 100 000 data points this is 745 seconds (12,5 minutes) and for the entire downsampled point cloud of 254 731 data points this is 1904 seconds (31 minutes). Furthermore, k-means clustering is also slowed down significantly.

There is a notable increase in quality of the results, especially in the dune area and intertidal zone. The increase in quality is bigger for the shape-based clustering solution than for the height-based. The results acquired when only using 10 000 data points is sufficient for this thesis, as only one scan per day was used and the beach as a whole was analysed rather than only a small area.

Using more data points would be interesting if one looks at changes between subsequent hourly scans at more detail. The visualisations of the resulting clustering solutions are shown in figures 31 until 32, computed with k=10 as well as k=20. When using this many data points, a larger value of k may lead to better results, especially in shape-based clustering.

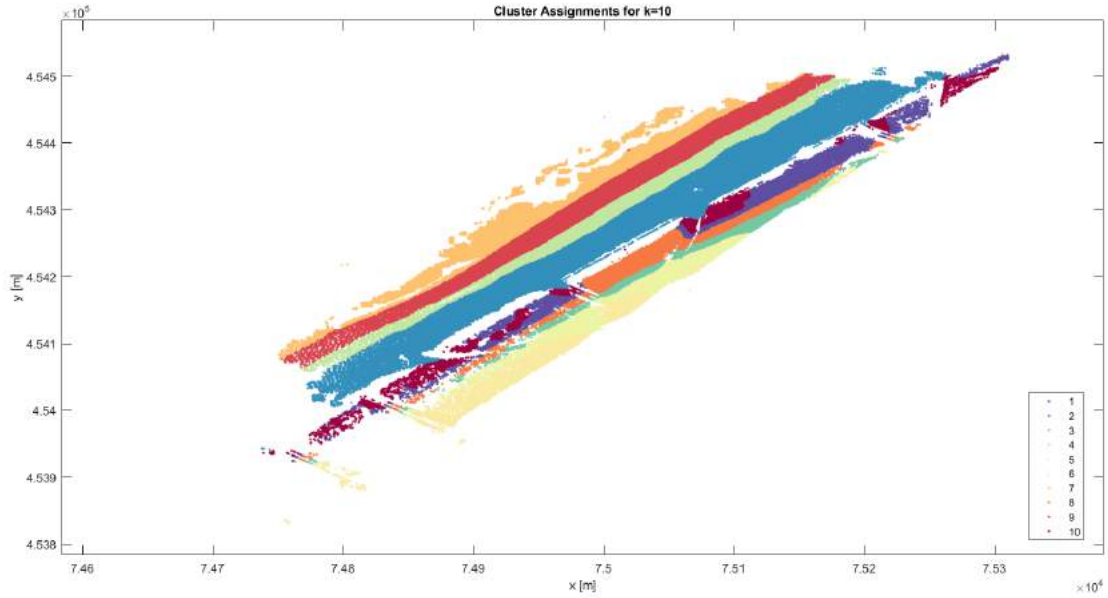


Figure 30: Visualisation of the 254 731 data points clustered using height-based k-means clustering, k=10

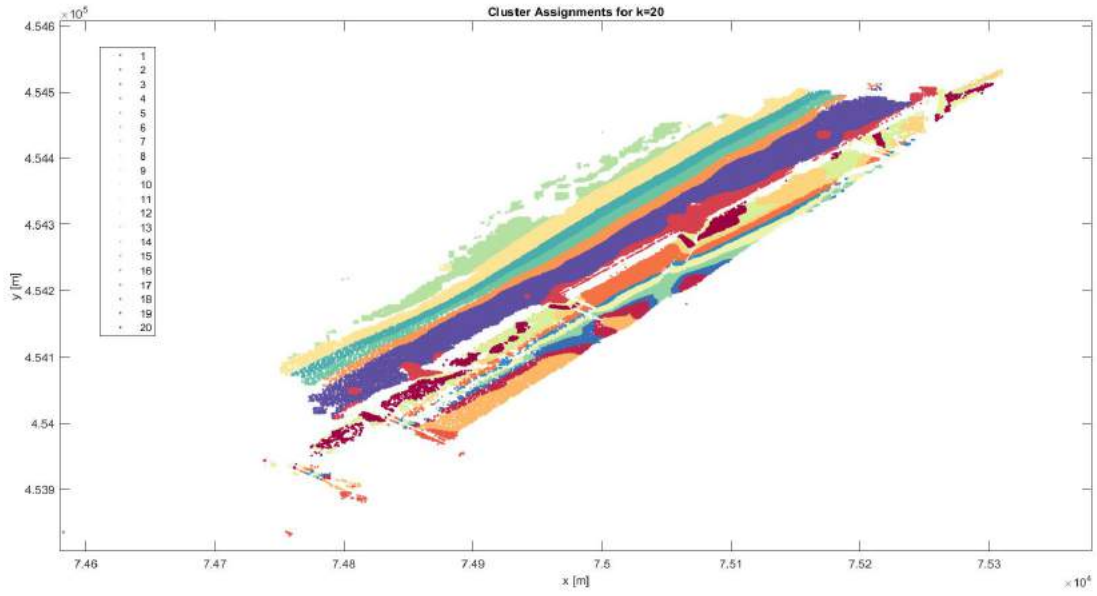


Figure 31: Visualisation of the 254 731 data points clustered using height-based k-means clustering, k=20

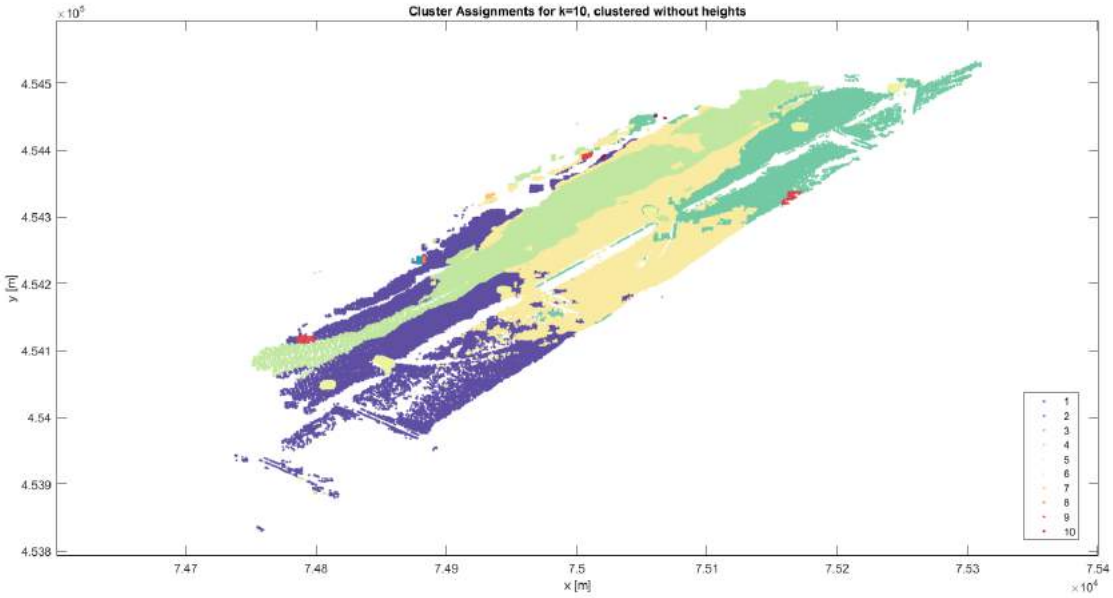


Figure 32: Visualisation of the 254 731 data points clustered using shape-based k-means clustering, k=10

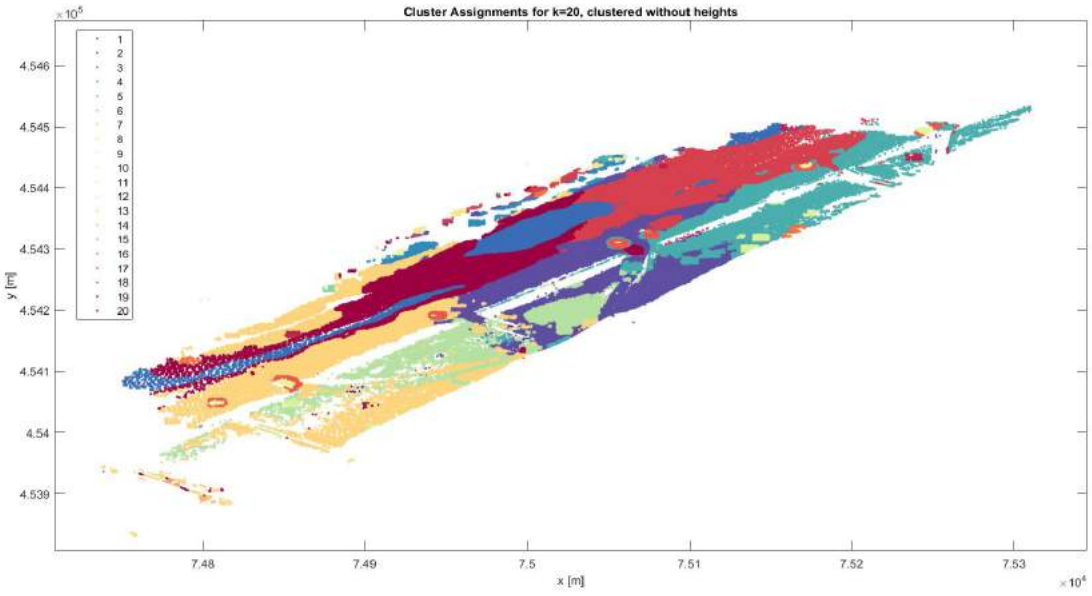


Figure 33: Visualisation of the 254 731 data points clustered using shape-based k-means clustering, k=20

6.1.4 Interpolation Methods

In this thesis, a data point was chosen on the first day of January, and a region of interest was defined around these data points. The region of interest used was the square around every data point, with the x and y values + or - 5 meters. For the subsequent days, the average of data points within the region of interest was then used to create a z-value for these days. If a person or other temporary object was within the region of interest, this average would be somewhat higher than the actual value.

Alternatively, a smaller radius could be used to produce better results. Especially when looking for small changes, such as ripples, as those are likely to be filtered out due to the large ROI. The downside of using a smaller distance when computing the ROI is that the number of complete time series decreases.

Figure 34 shows the height-based clustering solution computed using a ROI with x and y values +/- 5 m (as can be seen in the results chapter) and x and y values +/- 1.5 m. The results are very similar, and as such the results computed using the large ROI are deemed to be of sufficient quality for this thesis. Still, it is advised to use a smaller ROI in further research.

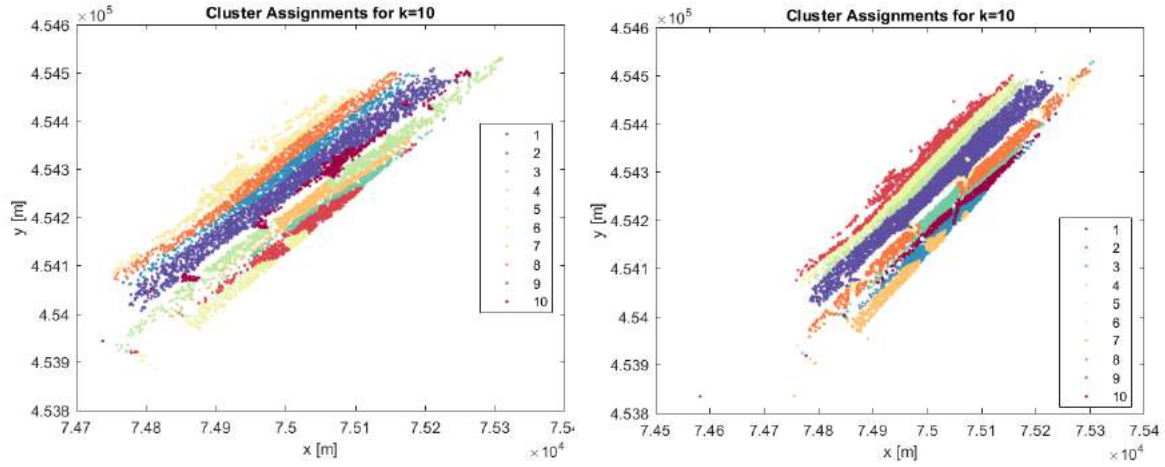


Figure 34: Comparison between the cluster assignment depending on the ROI defined. On the left the ROI was defined as the x and y values +/- 5 m, on the right as x and y values +/- 1.5 m

An alternative method to interpolate the data would be to assign a region of interest and take a random data point from this region. The risk would be that this random point is an outlier, affecting the accuracy of the results.

Another possibility is interpolating the area within the region of interest using the known data points, and then using the exact same location that was defined on the first day. This would give reliable results, but it would still have a larger risk of containing an outlier than the used method. It would also be slower.

6.2 K-means Clustering

6.2.1 Number of Iterations

If the number of iterations is not specified by the user, Matlab uses 100 iterations by default. However, sometimes the k-means clustering algorithm has failed to converge after 100 replicates, and the resulting clustering solution is not optimal. Therefore, the number of iterations was set to 150 in this thesis. This proved to be enough for a data point selection of 10 000 points, but if significantly more data points are used this value may need to be reconsidered.

6.2.2 Choosing K

There are alternative criteria, such as silhouette analysis or the Calinski Harabasz criterion, which may also be interesting to investigate.

In this thesis, the evaluation of the k-value was done using the Davies Bouldin criterion. Davies Bouldin evaluates the similarity of the points within the cluster, and it evaluates the differences between different clusters. This criterion was chosen because these characteristics are more valuable when clustering time series than the characteristics on which the other criteria evaluate a clustering solution.

The results were then computed using both k=10 and k=20. K=20 had the advantage that more detail could be distinguished, but the disadvantage that the results were less clear, and interpreting the results was difficult. Therefore, k=10 was used.

A higher value of k may however lead to interesting results when using more data locations, as can be seen in figures 31 until 32.

6.3 Missing Data

6.3.1 Weather Influence: Fog

A lot of the locations analysed in this thesis had missing data on the 9th of January. Interpolating did not improve the results, meaning that these points did not have neighbouring points within the specified distance. Upon opening the point cloud acquired on the 9th of January, it became apparent that although the number of data points was not extremely large (2.69 million vs. 2.22 million). However, the range of the scanner was significantly lower. The difference between an average scan and the scan of the 9th of January can be seen in figures 35 and 36. The cause of this was dense fog (NU 2017).

The influence of fog cannot be avoided, and it is important to recognise the impact on the data. In this thesis the decision was made to remove the data from this day completely, because the k-means clustering algorithm does not work on non-continuous time series. This decision was made to ensure that a large area of the beach would be included in the analysis, rather than only the part of the beach of which data was captured on the 9th of January. When looking at time series, it is important to take into account that the 9th of January is missing, and therefore day 9 in figures is actually day 10, and day 10 is actually day 11 and so on.

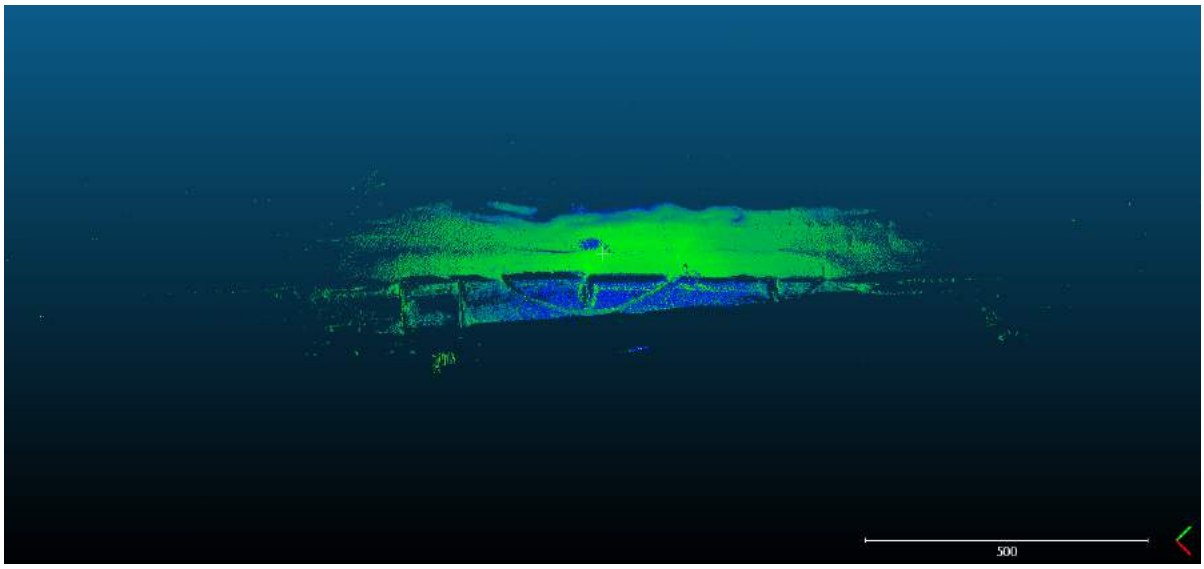


Figure 35: Visualisation of the data acquired on the 1st of January (2692693 data points)

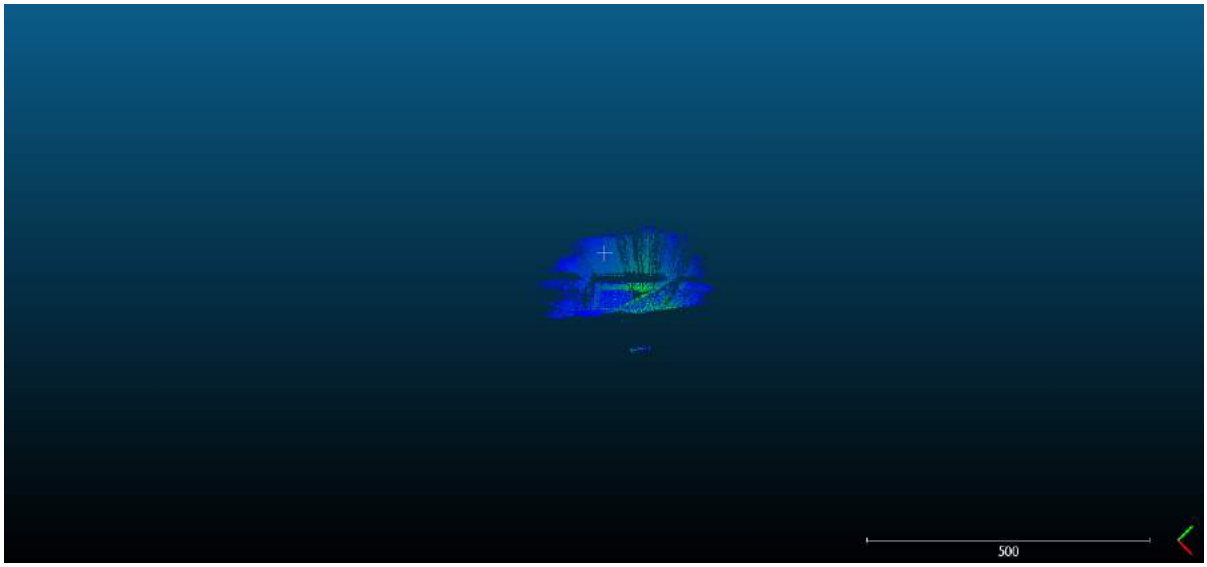


Figure 36: Visualisation of the data acquired on the 9th of January (223452 data points)

6.3.2 Weather Influence: Snow

Chapter 2 stated that upon analysing weather reports, it was found that it had snowed on the 12th and 13th of January. However, the effects of snow were not visible in the data; the snow probably melted immediately this close to the coast. Therefore, there was no problem with missing data on these days.

6.3.3 Scan Shadow

The scanner cannot capture data points directly behind other objects. Due to the sudden height difference between the dunes and the beach, a scan shadow can be observed parallel to the coastline. This is also the case for the paths towards the beach and some parts of the dune area. This problem cannot be solved in permanent laser scanning; in order to fill these missing areas the scanner would have to be moved.

6.3.4 Data Points in the Sea

CoastScan has difficulty capturing data points in the sea, as explained previously. However, the scanner does capture a couple of data points in the sea. These data points could be ships sailing past, or buoys. These data points are likely to have a lot of missing data on the other days of the month, and because of these incomplete time series they are not suitable for k-means clustering.

These data points were filtered out during preprocessing. This is not expected to affect the results, as this thesis focuses on deformation regimes on the beach, not in the sea.

6.3.5 Edges of the Scanned Area

The edges of the scanned area vary depending on the scanning conditions. On a good day without precipitation, the area captured by the scanner is likely to be bigger than on a gray misty day. Therefore, data points around the edges are also likely to have non-continuous time series. These data points were also filtered out during preprocessing, reducing the area studied.

7 Conclusion & Recommendations

7.1 Conclusion

It is concluded that it is possible to identify deformation regimes at Kijkduin using k-means clustering of time series. These time series were created using point clouds acquired by the Permanent Laser Scanning system CoastScan. K-means clustering is an unsupervised classification method, and therefore the results are unbiased. The clustered image of the beach was then interpreted using the cluster locations and the clustered time series. It was then discovered that immediately clustering the time series led to a largely height-based clustering solution. By taking the height out of the time series and clustering again, a shaped-based clustering solution was found.

The time series belonging to each cluster were analysed, and from this analysis different deformation regimes were interpreted. This was done using the time series belonging to each cluster, by judging the overall trend of the time series and by looking at the variability of the time series throughout the month. The resulting interpretations were verified using Google Maps satellite images and field observations.

The shape-based clustering results are more promising than the height-based clustering results, because the results are not influenced by the location of the data point. Instead, only the trend is used to cluster the time series. As such, the results are less predictable and deformation regimes are more distinguishable, because the same cluster may be found at different heights.

The height-based solution led to an interpretation of the beach in three areas: the intertidal zone, the backshore and the dunes. The shape-based solution led to a slightly different interpretation, and the boundaries between the zones were shifted a bit. The resulting image was the scanned section of beach divided into the intertidal zone, the backshore and the dunes, and what was interpreted to be a migrating sandbar was found.

7.2 Recommendations

There are recommendations so as to improve the current results, and recommendations for further research.

For this thesis, the .xyz data was manually converted to .ply data. For further research it is recommended to convert the files automatically, especially if more scans are used (ie hourly scans, or a longer time period).

A number of variables were used to compute the results described in this research, such as the downsampling grid step and the size of the region of interest. The focus of this research was mainly towards investigating whether and how k-means interpolating could be used to identify deformation regimes, and as such these variables were not chosen with a lot of research. Therefore, these variables may not lead to the best results. It is recommended to further investigate which values for these variables give the best results. If the size of the ROI is set smaller, the importance of morphological filtering increases, and as such it is advisable to perform morphological filtering to avoid outliers.

The results in this thesis were acquired using 10 000 data locations. The results are more detailed if more data points are used, such as the entire downsampled pointcloud (approximately 250 000 data points). This was already shortly mentioned in the discussion. If more data points are used, adding more clusters will allow for more detailed classification, and smaller deformation regimes can be identified. It is recommended to analyse the results obtained using more data points, and it is recommended to add more clusters. If hourly scans are used rather than daily scans, smaller deformation regimes may be found as well.

In this thesis, unsupervised classification was done using k-means clustering. However, other unsupervised classification methods may also be useful in trying to locate and identify different deformation

regimes. It is therefore recommended to investigate whether a different classification method leads to better results. K-means clustering was performed twice, height-based and shape-based. The shape-based results were found to be more promising when trying to identify different deformation regimes, and as such it is recommended to focus on the shape-based method in further research.

K-means clustering of time series can only be done using complete time series. In this thesis, incomplete time series were simply removed. One day (January 9th) was removed as a whole because of the poor scanning results. Although this approach is simple and effective, it is not the most elegant approach and may not lead to the best results. It is recommended that further research investigates whether there are better methods to deal with incomplete time series.

8 Bibliography

References

Arthur, David and Sergei Vassilvitskii (2007). "k-means++: The advantages of careful seeding". In: *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics, pp. 1027–1035.

Boehler, Wolfgang, M Bordas Vicent, Andreas Marbs, et al. (2003). "Investigating laser scanner accuracy". In: *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences* 34.Part 5, pp. 696–701.

Cowell, Peter J et al. (2003). "The coastal-tract (part 1): a conceptual approach to aggregated modeling of low-order coastal change". In: *Journal of Coastal Research*, pp. 812–827.

Geography, GIS (2018). *Image Classification Techniques in Remote Sensing*. `https://gisgeography.com/image-classification-techniques-remote-sensing/`. Accessed: 2018-09-12.

Grilli, E, F Menna, and F Remondino (2017). "A review of point clouds segmentation and classification algorithms". In: *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences* 42, p. 339.

Hejbudzka, K et al. (2010). "Influence of atmospheric conditions on the range distance and number of returned points in Leica ScanStation 2 point clouds". In: *Proceedings ISPRS Commission V Mid-Term Symposium? Close Range Image Measurement Techniques?, Newcastle upon Tyne, UK, 21-24 June 2010; IAPRS, XXXVIII (5), 2010*. International Society of Photogrammetry and Remote Sensing (ISPRS).

Hobbelen, Rik van (2018). "Mapping beach growth using terrestrial laser scanning". MA thesis. http://repository.tudelft.nl/: TU Delft.

Hsu, Tian-Jian, Steve Elgar, and RT Guza (2006). "Wave-induced sediment transport and onshore sandbar migration". In: *Coastal Engineering* 53.10, pp. 817–824.

Huang, Xiaohui et al. (2016). "Time series k-means: A new k-means type smooth subspace clustering for time series data". In: *Information Sciences* 367, pp. 1–13.

KNMI (2017). *Januari 2017*. `https://www.knmi.nl/nederland-nu/klimatologie/maand-en-seizoensoverzichten/2017/januari`. Accessed: 2018-09-11.

Lemmens, Mathias (2011). "Terrestrial laser scanning". In: *Geo-information*. Springer, pp. 101–121.

Lindenbergh, Roderik C et al. (2011). "Aeolian beach sand transport monitored by terrestrial laser scanning". In: *The Photogrammetric Record* 26.136, pp. 384–399.

Liu, Yanchi et al. (2010). "Understanding of internal clustering validation measures". In: *Data Mining (ICDM), 2010 IEEE 10th International Conference on*. IEEE, pp. 911–916.

Masselink, Gerd and Roland Gehrels (2014). *Coastal environments and global change*. John Wiley & Sons.

MATLAB (2018). *Matlab Documentation*. `https://nl.mathworks.com/help`. Accessed: 2018-09-27.

Nichols, Gary (2009). *Sedimentology and stratigraphy*. John Wiley & Sons.

NU (2017). *Schiphol rekent op vertragingen door dichte mist*. `https://www.nu.nl/reizen/4376160/schiphol-rekent-vertragingen-dichte-mist.html`. Accessed: 2018-09-21.

Piech, Chris (2013). *K Means*. `http://stanford.edu/~cpiech/cs221/handouts/kmeans.html`. Accessed: 2018-10-21.

Poortengen, E.G. (2017). "Classifying Mangroves in Vietnam using Radar and Optical Satellite Remote Sensing". MA thesis. http://repository.tudelft.nl/: TU Delft.

Reshetyuk, Yuriy (2006). "Investigation and calibration of pulsed time-of-flight terrestrial laser scanners". PhD thesis. KTH.

RIEGL (2017). *RIEGL VZ-2000*. `http://www.riegl.com/nc/products/terrestrial-scanning/produktdetail/product/scanner/58/`. Accessed: 2018-09-09.

Rijkswaterstaat (2018). *Kustonderhoud*. https://www.rijkswaterstaat.nl/water/waterbeheer/bescherming-tegen-het-water/maatregelen-om-overstromingen-te-voorkomen/kustonderhoud/index.aspx. Accessed: 2018-09-11.

Slob, Siefko (2010). "Automated rock mass characterisation using 3-D terrestrial laser scanning". PhD thesis. TU Delft, Delft University of Technology.

Soudarissanane, Sylvie et al. (2011). "Scanning geometry: Influencing factor on the quality of terrestrial laser scanning points". In: *ISPRS Journal of Photogrammetry and Remote Sensing* 66.4, pp. 389–399.

Stive, Marcel JF et al. (2013). "A new alternative to saving our beaches from sea-level rise: The sand engine". In: *Journal of Coastal Research* 29.5, pp. 1001–1008.

Trevino, Andrea (2016). *Introduction to K-means Clustering*. https://www.datascience.com/blog/k-means-clustering. Accessed: 2018-09-14.

Virtanen, Juho-Pekka et al. (2014). "Rapid prototyping—A tool for presenting 3-dimensional digital models produced by terrestrial laser scanning". In: *ISPRS International Journal of Geo-Information* 3.3, pp. 871–890.

Vos, Sander et al. (2017). "Coastscan: Continuous monitoring of coastal change using terrestrial laser scanning". In: *Proceedings of the Coastal Dynamics*, pp. 1518–1528.

Vosselman, George and Hans-Gerd Maas (2010). *Airborne and terrestrial laser scanning*. CRC.

9 Appendix

9.1 Appendix A - Script

9.1.1 Creating Time Series & K-Means Clustering

```
clear all
close all
%% load data into matlab
tic
PointCloudsPLY = dir('*.ply'); %tells MATLAB which type of file to import
numfiles = length(PointCloudsPLY);
ptCloudsPLY = cell(1, numfiles); %creates a cell array for the point clouds imported

for k = 1:numfiles
  ptCloudsPLY{k} = pcread(PointCloudsPLY(k).name); %fills the mydata cell with the ...
      specified point clouds
end
toc %Elapsed time is 10.590046 seconds
%% Lower Resolution!!
%blokje van de resolutieverlaging aantal punten comprimeren naar maar 1 punt
tic
for i = 1:numfiles
    gridStep = 0.5; % in 0.5 m
    ptCloudsPLY{i} = pcdownsample(ptCloudsPLY{i}, 'gridAverage', gridStep);
    %pcdownsample downsamples 3D point cloud using box grid filter. gridStep ...
        specifies the size of a 3D box, points within the
    %same box are merged to a single point in the output. gridAverage preserves the ...
        shape of the point cloud
end
toc %Elapsed time is 15.453562 seconds.
%% visualise downsampled point cloud
figure
pcshow(ptCloudsPLY{1})
title('Point Cloud Downsampled with Grid Step 0.5')
xlabel('X [m]')
ylabel('Y [m]')
zlabel('Z [m]')
colorbar

%% pick one point
tic
no_points = 10000; %specify how many points to select
positions = datasample(ptCloudsPLY{1}.Location,no_points); %select no_points random ...
    points from ptCloudsPLY{1} and saves them to positions

results = zeros(numfiles,no_points); %reserves space for the results
 for j = 1:no_points
    for i = 1:numfiles
        i %prints on which i the for loop is in the command window
        roi = [positions(j,1)-5, positions(j,1)+5 ; positions(j,2)-5 , ...
            positions(j,2)+5 ; -inf, inf]; %roi is region of interest, the x and y ...
            values - or + 5, z can have every value. j indicates the row of xyz
        indices = findPointsInROI(ptCloudsPLY{i}, roi); %returns the points within a ...
            region of interest, different for every point!
        results(i,j) = mean(ptCloudsPLY{i}.Location(indices,3)); %takes the mean of ...
            the points within the region of interest. column 1 is selected point 1, ...
```

```
                column 2 selected point 2 etc
    end
 end
 toc %Elapsed time is 94.182600 seconds.
positions = positions';
%% further preparing results dataset
tic
results_1 = results;
results_1(9,:) = []; %removes 9th of january alltogether because of mist
results_1(results_1 > 21) = NaN; %set all points higher than 30 to Not a Number: ...
    these points are from the roof!
results_2 = results_1(:,all(¬isnan(results_1))); %removes all NaN values
toc %Elapsed time is 0.015411 seconds.
%%
cell = {positions(:,1),positions(:,2), results}; %cell array where 1x1 is x ...
    positions of all points, 1x2 is y positions of all points, 1x3 is z poisitions ...
    of all points for all days
xy = [positions(1,:); positions(2,:)]; %row 1 is x value of all points, row 2 is y ...
    value of all points
array = vertcat(xy, results_1); %row 1 is x value of all points, row 2 is y value of ...
    all points, row 3-33 is z values
array_1 = array(:,all(¬isnan(array))); %removes all NaN values
%% create point cloud out of data points, compare with original point cloud
ptCloud = pointCloud(positions');
figure %point cloud 10000 data points
pcshow(ptCloud)
title('Point Cloud Data Point Selection (10000)')
xlabel('X [m]')
ylabel('Y [m]')
zlabel('Z [m]')
colorbar

ptCloud2 = pointCloud(array_1(1:3,:)');
figure %point cloud 8251 data points: height <21m and NaN removed
pcshow(ptCloud2)
title('Point Cloud Data Point Selection (8251)')
xlabel('X [m]')
ylabel('Y [m]')
zlabel('Z [m]')
colorbar
%% plotting time series for no_points and for remaining points
%results_new(isnan(results_new))=[]; set all nan values to 0
tic
figure
plot(results) %plot z value over time, 9th of January missing!
title(['Time series of z-value at ' num2str(no_points) ' selected locations'])
xlabel('time (days)')
ylabel('Z [m]')
xlim([1 31]) %because data runs from day 1 until day 31

figure
plot(results_2) %plot z value over time, 9th of January missing!
title(['Time series of z-value, remaining 8251 locations'])
xlabel('time (days)')
ylabel('Z [m]')
xlim([1 31]) %because data runs from day 1 until day 31
toc %Elapsed time is 4.770509 seconds.

%% subtracting average height per time series
X = array_1'; %transposes results matrix for k means clustering
X2 = X(:,3:32); %only z values
X3 = zeros(8251, 30);
for i = 1:8251
    X3(i,:) = X2(i,:) - mean(X2(i,:));
end
```

```
X3 = horzcat(X(:,1:2), X3); %put back x and y location for plotting
X4 = X3(:,3:32);

%% use k means to create an input matrix of proposed clustering solutions
tic
clust= zeros(size(X2,1),25); %clustering with height
for i=1:25
clust(:,i) = kmeans(X2,i, 'MaxIter', 150, 'replicate',5);
end
toc

tic
clust2 = zeros(size(X4,1),25); %clustering without height
for i=1:25
clust2(:,i) = kmeans(X4,i, 'MaxIter', 150, 'replicate',5);
end
toc

%for X2 Elapsed time is 69.753996 seconds. (with heights). clust. k t/m 100
%for X4 Elapsed time is 127.995280 seconds. (without heights). clust2.k t/m 100

%% create groups in order to visualise time series per group
tic
k = 20; %20 clusters
for i=1:k
  groups_clust{i} = X2(clust(:,20) == i , : ); %groups with height
end

for i=1:k
  groups_clust2{i} = X2(clust2(:,20) == i , : ); %groups clustered without height, ...
      but shown at their original height
end

for i=1:k
  groups_clust4{i} = X4(clust2(:,20) == i , : ); %groups clustered without height, ...
      and shown without height
end
toc %Elapsed time is 0.012168 seconds.

%% standard deviation
tic
S = std(X2,0,2); %for every time series

%calculate mean per group
for i = 1:k
    groups_mean{i} = mean(groups_clust{i}); %clustered with height
    groups_mean2{i} = mean(groups_clust2{i}); %clustered without height, at original ...
        height
    groups_mean4{i} = mean(groups_clust4{i}); %clustered without height, at height 0
end

%standard deviation
for i = 1:k
    S_groups{i}=std(groups_clust{i});
    S_groups2{i}=std(groups_clust2{i});
    S_groups4{i}=std(groups_clust4{i});
end

%calculate mean - standard deviation
for i = 1:k
    groups_mean_min{i} = groups_mean{i} - S_groups{i};
    groups_mean2_min{i} = groups_mean2{i} - S_groups2{i};
    groups_mean4_min{i} = groups_mean4{i} - S_groups4{i};
end

```

```
%calculate mean + standard deviation
for i = 1:k
    groups_mean_max{i} = groups_mean{i} + S_groups{i};
    groups_mean2_max{i} = groups_mean2{i} + S_groups2{i};
    groups_mean4_max{i} = groups_mean4{i} + S_groups4{i};
end
toc %Elapsed time is 0.010814 seconds.
```

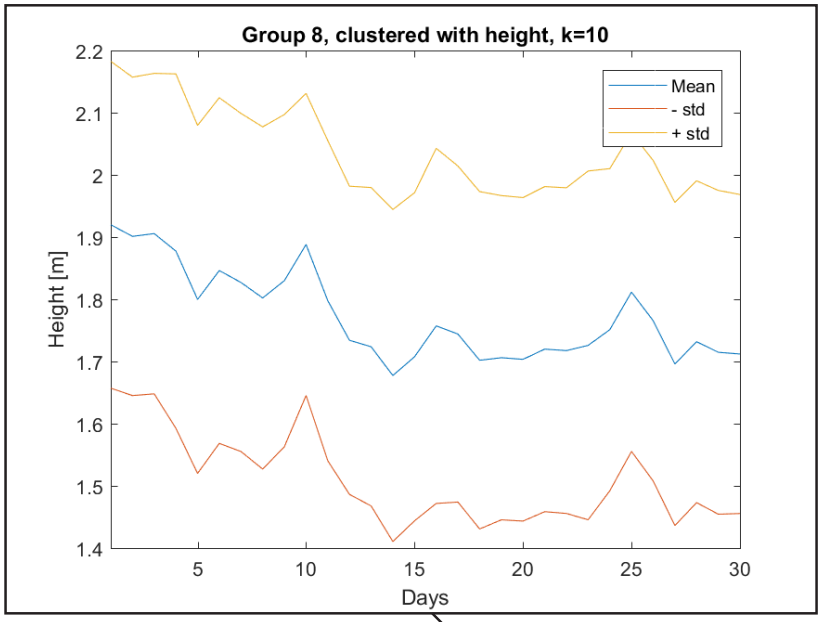
9.1.2 Creating Figures

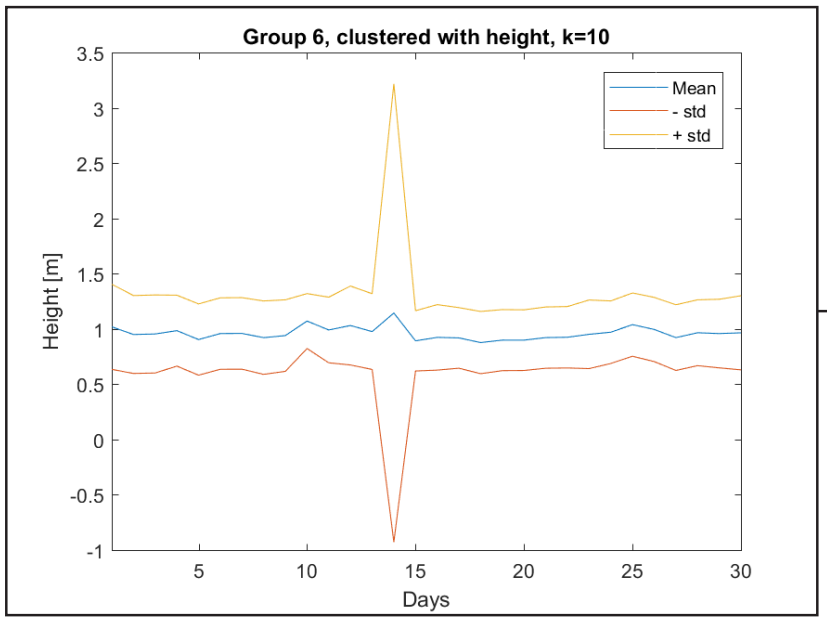
```
%% CREATING FIGURES

%% visualise beach
for k=1:20
 h=figure
 colors = linspecer(k);
 gscatter(X(:,1), X(:,2), clust2(:,k),colors)
 title(['Cluster Assignments for k=' num2str(k) ' clustered without height'])
 xlabel('Days')
 ylabel('Height [m]')
 hold off
 saveas(h,sprintf('vorm_clusters_%d.png',k)); %directly saves the figures to the ...
     current map
end

for k=1:20
 h=figure
 colors = linspecer(k);
 gscatter(X(:,1), X(:,2), clust(:,k),colors)
 title(['Cluster Assignments for k=' num2str(k) ' clustered with height'])
 xlabel('Days')
 ylabel('Height [m]')
 hold off
 saveas(h,sprintf('hoogte_clusters_%d.png',k));
end
%% creating and saving images of the time series per group
for k=1:20 %groups_clust (clustered with height)
 h=figure
 plot(groups_clust{k}')
 xlim([1, 30])
 title(['Group ' num2str(k) ' clustered with height, k=20']) %, plotted with height
 xlabel('Days')
 ylabel('Height (m)')
 saveas(h,sprintf('k20_height_%d.png',k));
end
%%
for k=1:20 %groups_clust2 (clustered without height, plotted with height)
 h=figure
 plot(groups_clust2{k}')
 xlim([1, 30])
 title(['Group ' num2str(k) ' clustered without height, plotted with height, k=20'])
 xlabel('Days')
 ylabel('Height (m)')
 saveas(h,sprintf('k20_noheight_yes_%d.png',k));
end
%%
for k=1:20 %groups_clust4 (clustered without height, plotted without height)
 h=figure
 plot(groups_clust4{k}')
 xlim([1, 30])
 title(['Group ' num2str(k) ' clustered without height, plotted without height, k=20'])
 xlabel('Days')
```

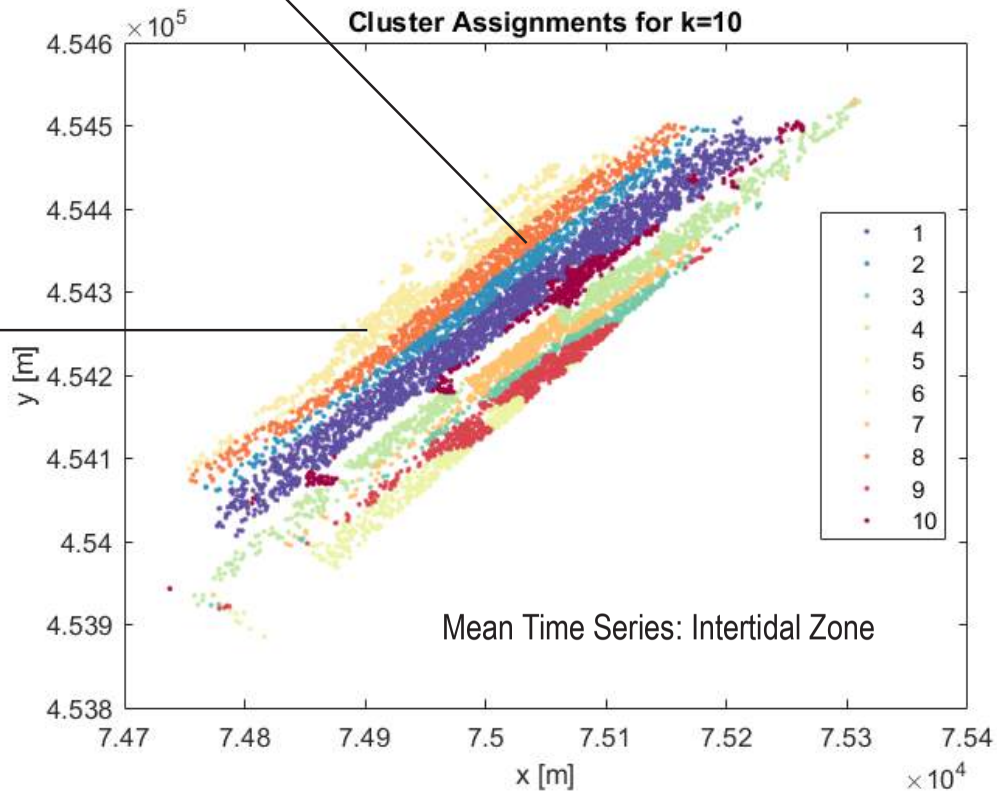
```
 ylabel('Height (m)')
 saveas(h,sprintf('k20_noheight_no_%d.png',k));
end

%% show mean per group and standard deviation

%groups_clust (clustered with height)
for k=1:20
 h=figure
 plot(groups_mean{k})
 hold on
 plot(groups_mean_min{k})
 hold on
 plot(groups_mean_max{k})
 legend('Mean','- std','+ std')
 xlim([1,30])
 title(['Group ' num2str(k) ', clustered with height, k=20'])
 xlabel('Days')
 ylabel('Height [m]')
 hold off
 saveas(h,sprintf('std_height_%d.png',k));
end

%groups_clust2 (clustered without height, plotted with height)
for k=1:20
 h=figure
 plot(groups_mean2{k})
 hold on
 plot(groups_mean2_min{k})
 hold on
 plot(groups_mean2_max{k})
 legend('Mean','- std','+ std')
 xlim([1,30])
 title(['Group ' num2str(k) ', clustered without height, plotted with height, k=20'])
 xlabel('Days')
 ylabel('Height [m]')
 hold off
 saveas(h,sprintf('std_noheight_yes_%d.png',k));
end

%groups_clust4 (clustered without height, plotted without height)
for k=1:20
 h=figure
 plot(groups_mean4{k})
 hold on
 plot(groups_mean4_min{k})
 hold on
 plot(groups_mean4_max{k})
 legend('Mean','- std','+ std')
 xlim([1,30])
 title(['Group ' num2str(k) ', clustered without height, plotted without height, k=20'])
 xlabel('Days')
 ylabel('Height [m]')
 hold off
 saveas(h,sprintf('std_noheight_no_%d.png',k));
end
%%
figure
plot(groups_clust{9}')

figure
plot(groups_mean{9})
%%
figure %plot mean of groups clustered with height
for i = 1:k
```

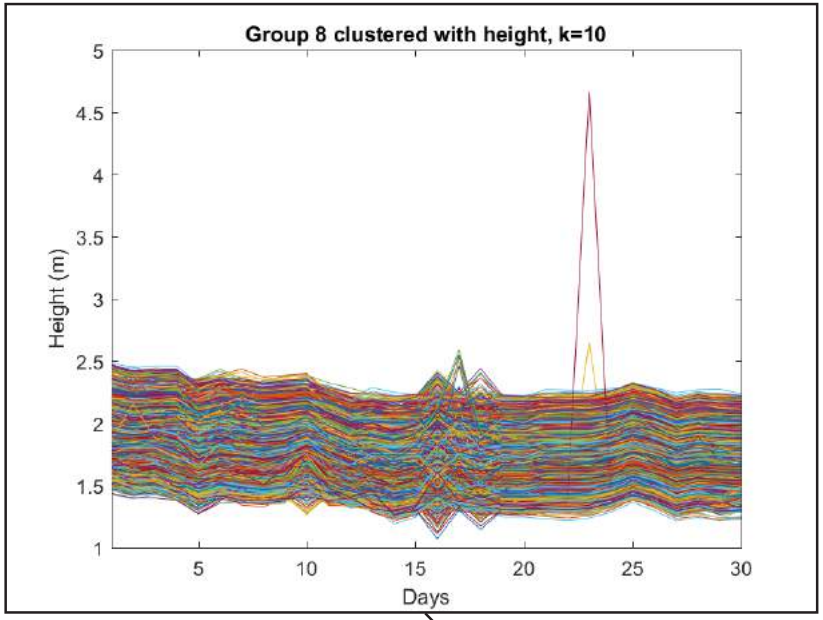
```
117     h(i) = plot(groups_mean{i}','color',c(i,:))
118     hold on
119     xlim([1, 30])
120 end
121 title('Mean per group, clustered with height, k=20')
122 ylabel('Height [m]')
123 xlabel('Days')
124 legend(h, 'Group 1', 'Group 2', 'Group 3', 'Group 4', 'Group 5', 'Group 6', 'Group ...
        7', 'Group 8', 'Group 9',...
125     'Group 10', 'Group 11', 'Group 12', 'Group 13', 'Group 14', 'Group 15', 'Group ...
            16', 'Group 17', 'Group 18', 'Group 19', 'Group 20')
126 %%
127 figure %plot mean of groups clustered without height, plotted with height
128 for i = 1:k
129     h(i) = plot(groups_mean2{i}','color',c(i,:))
130     hold on
131     xlim([1, 30])
132 end
133 title('Mean per group, clustered without height, plotted with height, k=20')
134 ylabel('Height [m]')
135 xlabel('Days')
136 legend(h, 'Group 1', 'Group 2', 'Group 3', 'Group 4', 'Group 5', 'Group 6', 'Group ...
        7', 'Group 8', 'Group 9',...
137     'Group 10', 'Group 11', 'Group 12', 'Group 13', 'Group 14', 'Group 15', 'Group ...
            16', 'Group 17', 'Group 18', 'Group 19', 'Group 20')
138
139 figure %plot mean of groups clustered without height, plotted without height
140 for i = 1:k
141     h(i) = plot(groups_mean4{i}','color',c(i,:))
142     hold on
143     xlim([1, 30])
144 end
145 title('Mean per group, clustered without height, plotted without height, k=20')
146 ylabel('Height [m]')
147 xlabel('Days')
148 legend(h, 'Group 1', 'Group 2', 'Group 3', 'Group 4', 'Group 5', 'Group 6', 'Group ...
        7', 'Group 8', 'Group 9',...
149     'Group 10', 'Group 11', 'Group 12', 'Group 13', 'Group 14', 'Group 15', 'Group ...
            16', 'Group 17', 'Group 18', 'Group 19', 'Group 20')
```

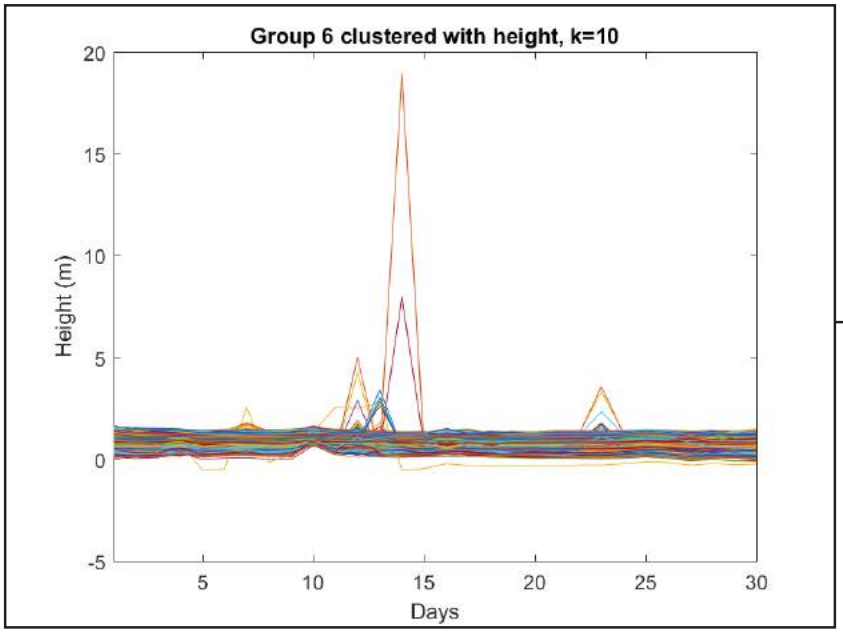
9.2 Appendix B - Height-Based Clustering

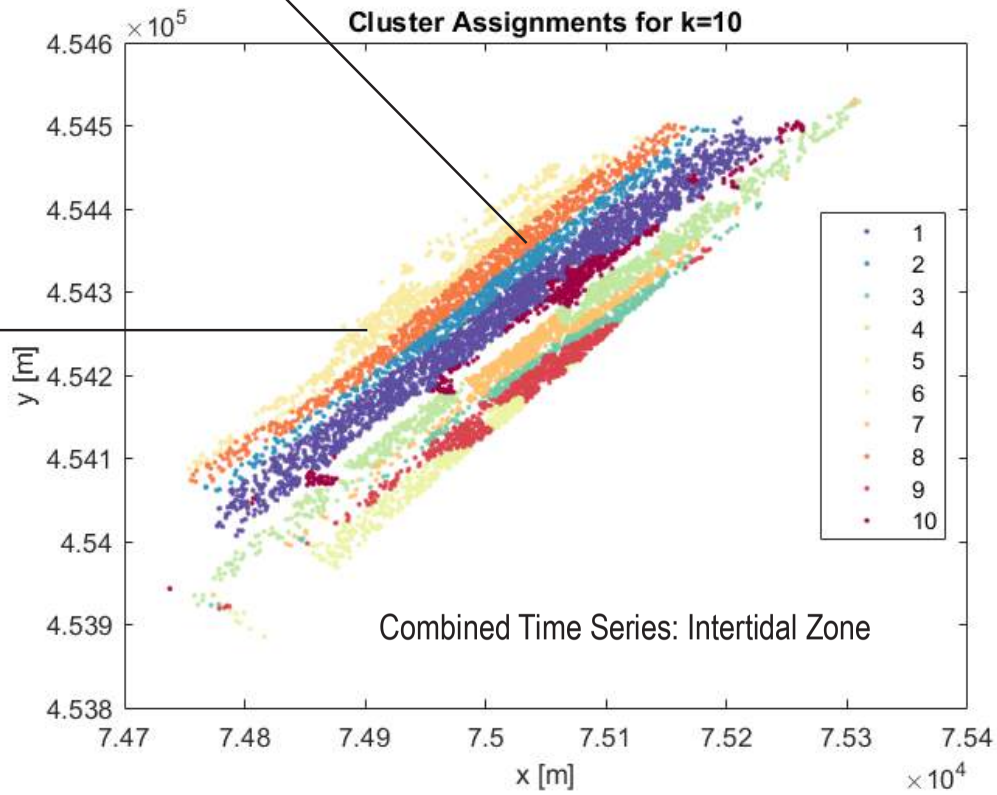

Group 8, clustered with height, k=10
Mean
- std
+ std
Height [m]
Days

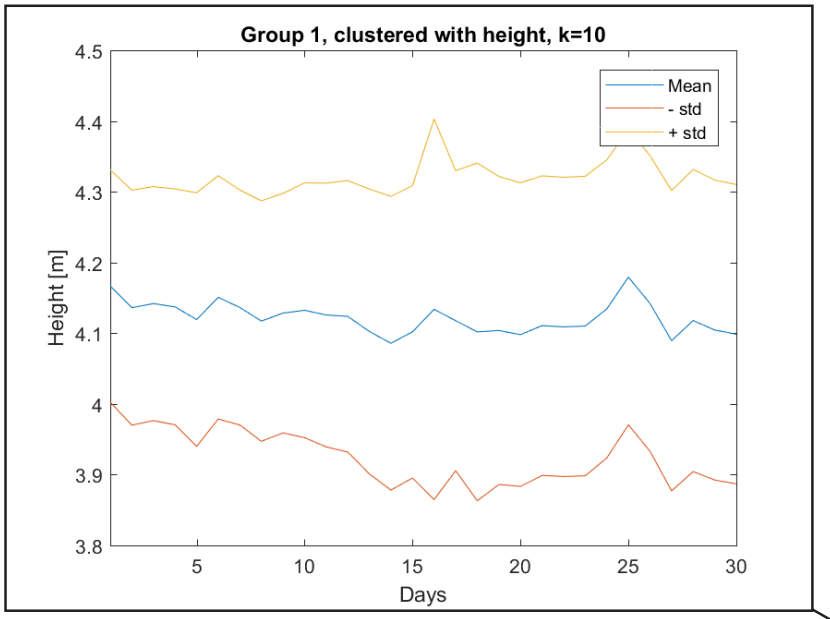

Group 6, clustered with height, k=10
Mean
- std
+ std
Height [m]
Days

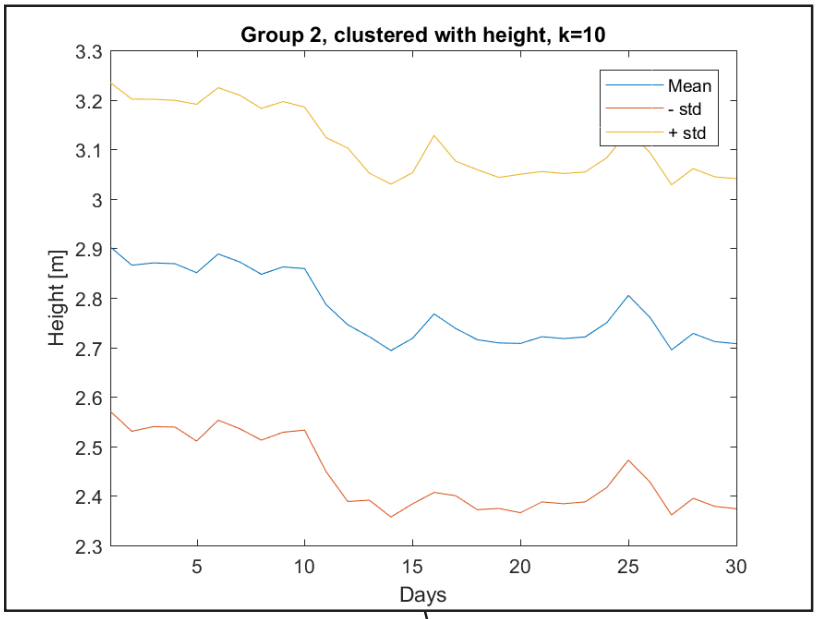

Cluster Assignments for k=10
×10^5
y [m]
x [m]
×10^4
1
2
3
4
5
6
7
8
9
10
Mean Time Series: Intertidal Zone

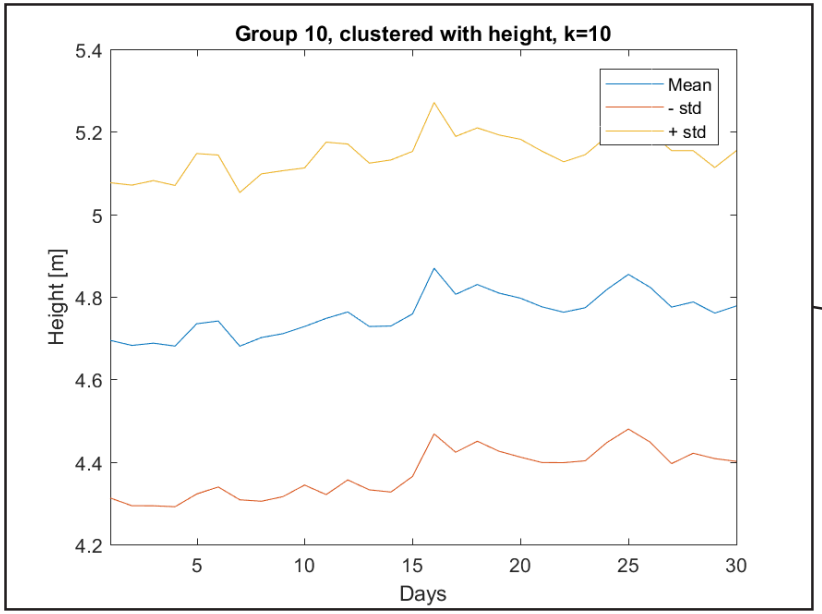

Group 8 clustered with height, k=10
Height (m)
Days

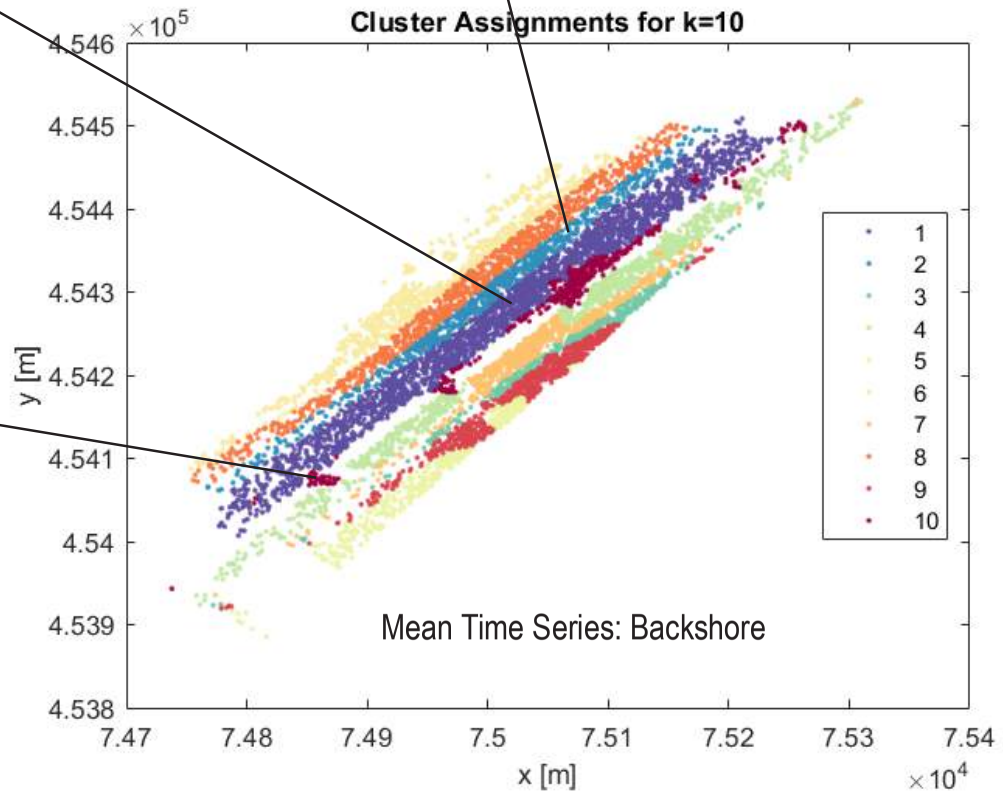

Group 6 clustered with height, k=10
Height (m)
Days

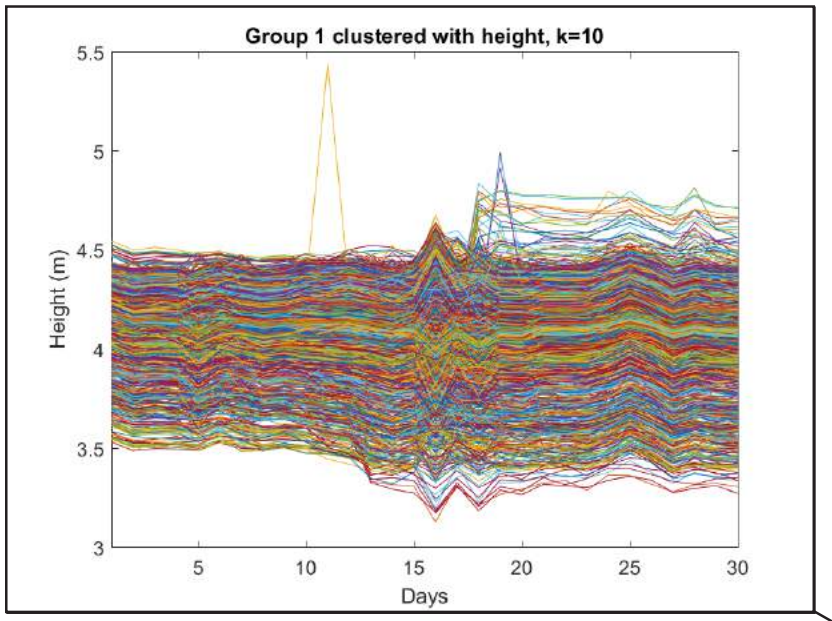

Cluster Assignments for k=10
y [m]
x [m]
Combined Time Series: Intertidal Zone

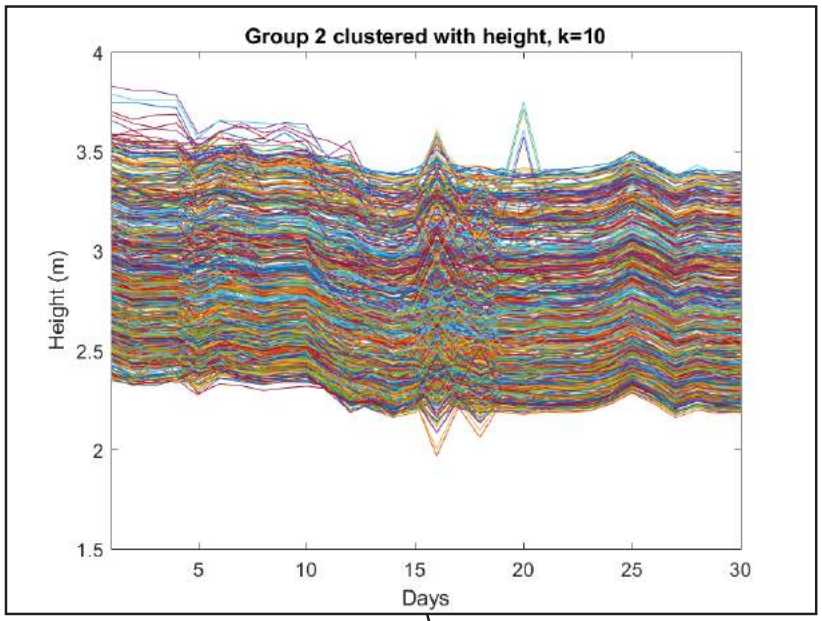

Group 1, clustered with height, k=10
Mean
- std
+ std
Height [m]
Days

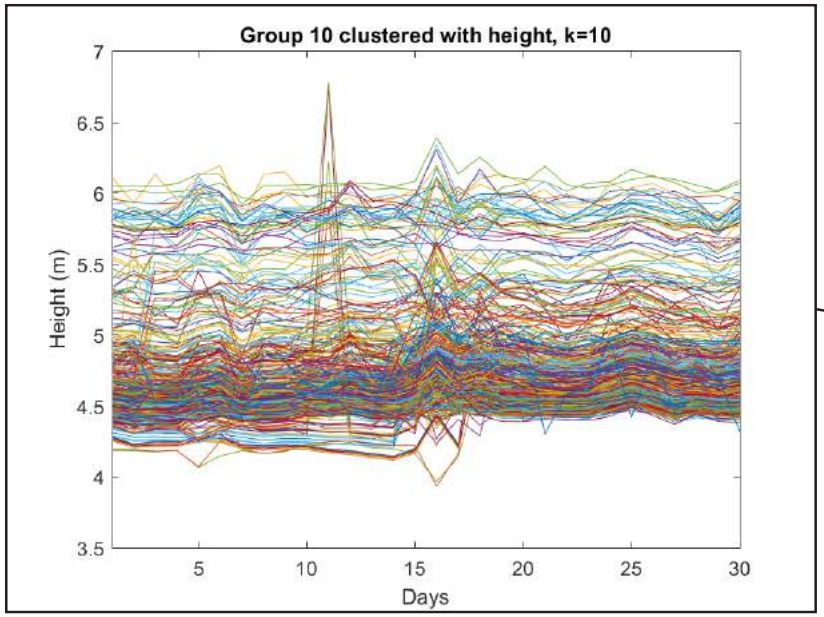

Group 2, clustered with height, k=10
Mean
- std
+ std
Height [m]
Days

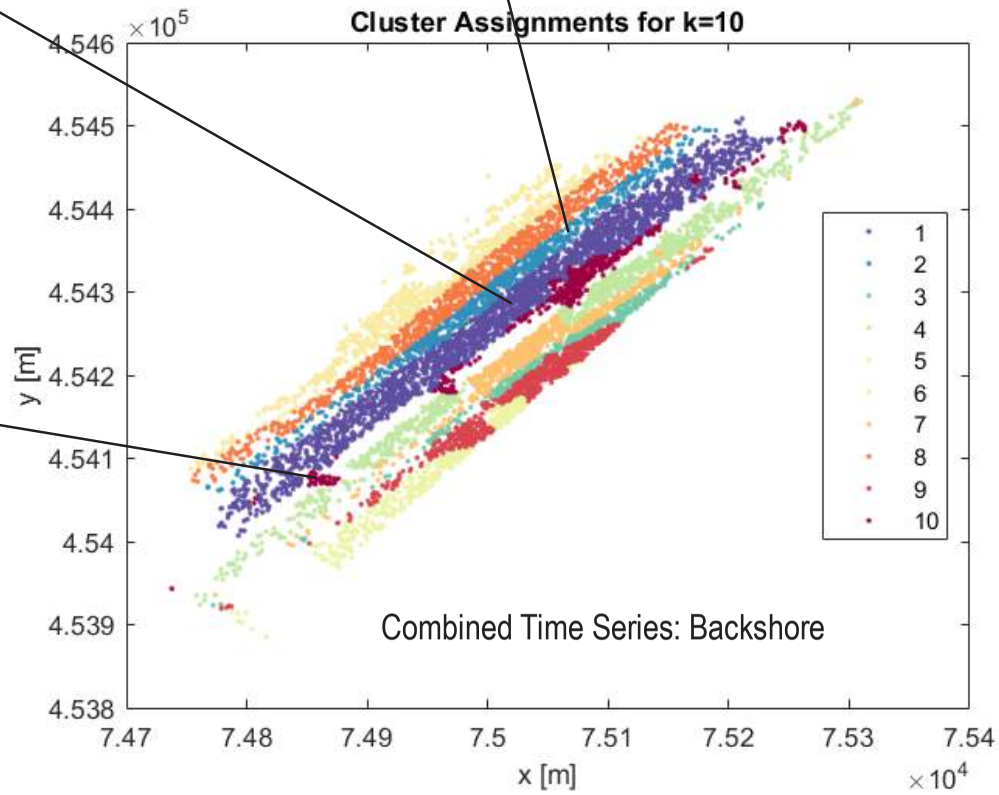

Group 10, clustered with height, k=10
Mean
- std
+ std
Height [m]
Days

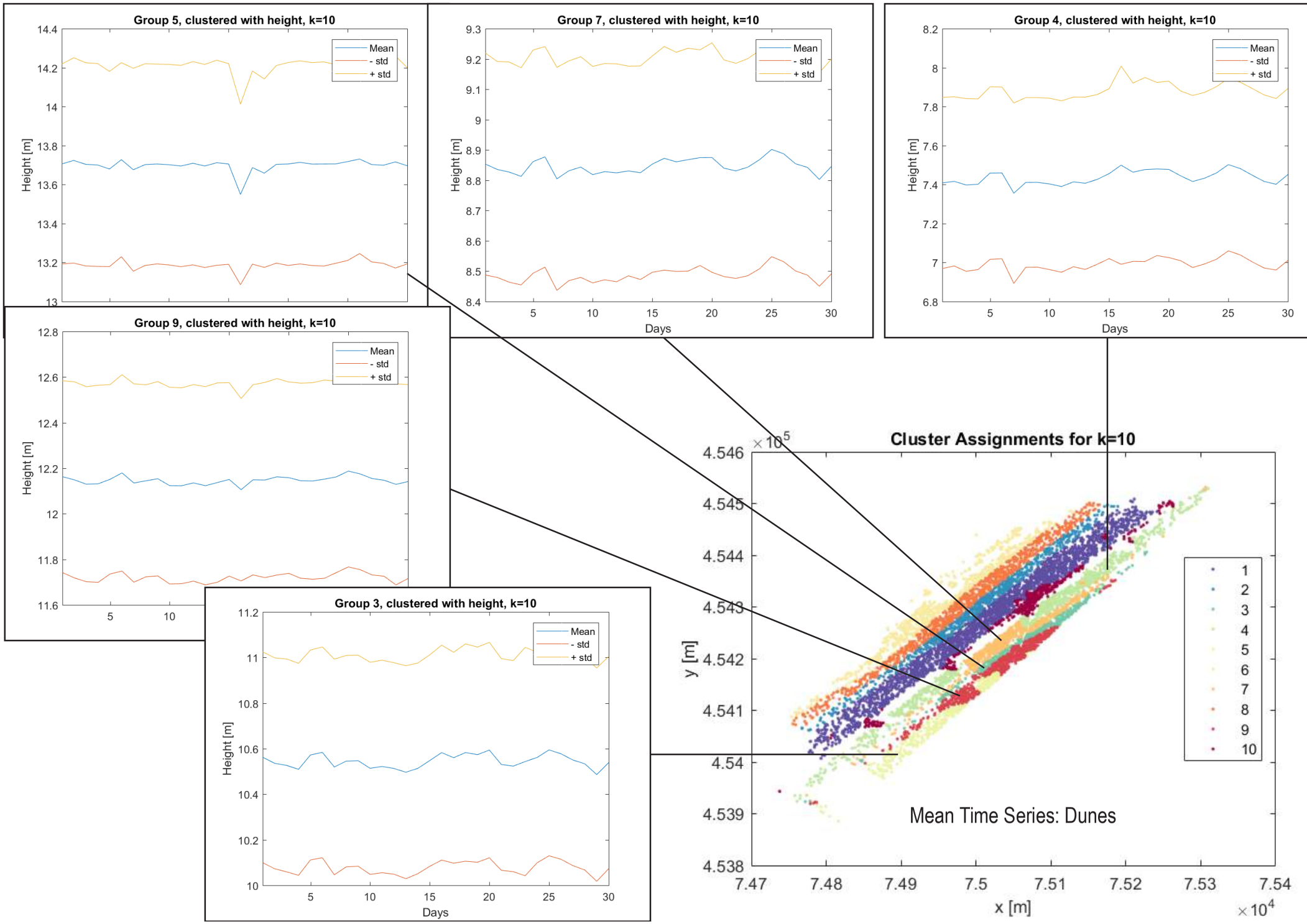

Cluster Assignments for k=10
Mean Time Series: Backshore
y [m]
x [m]

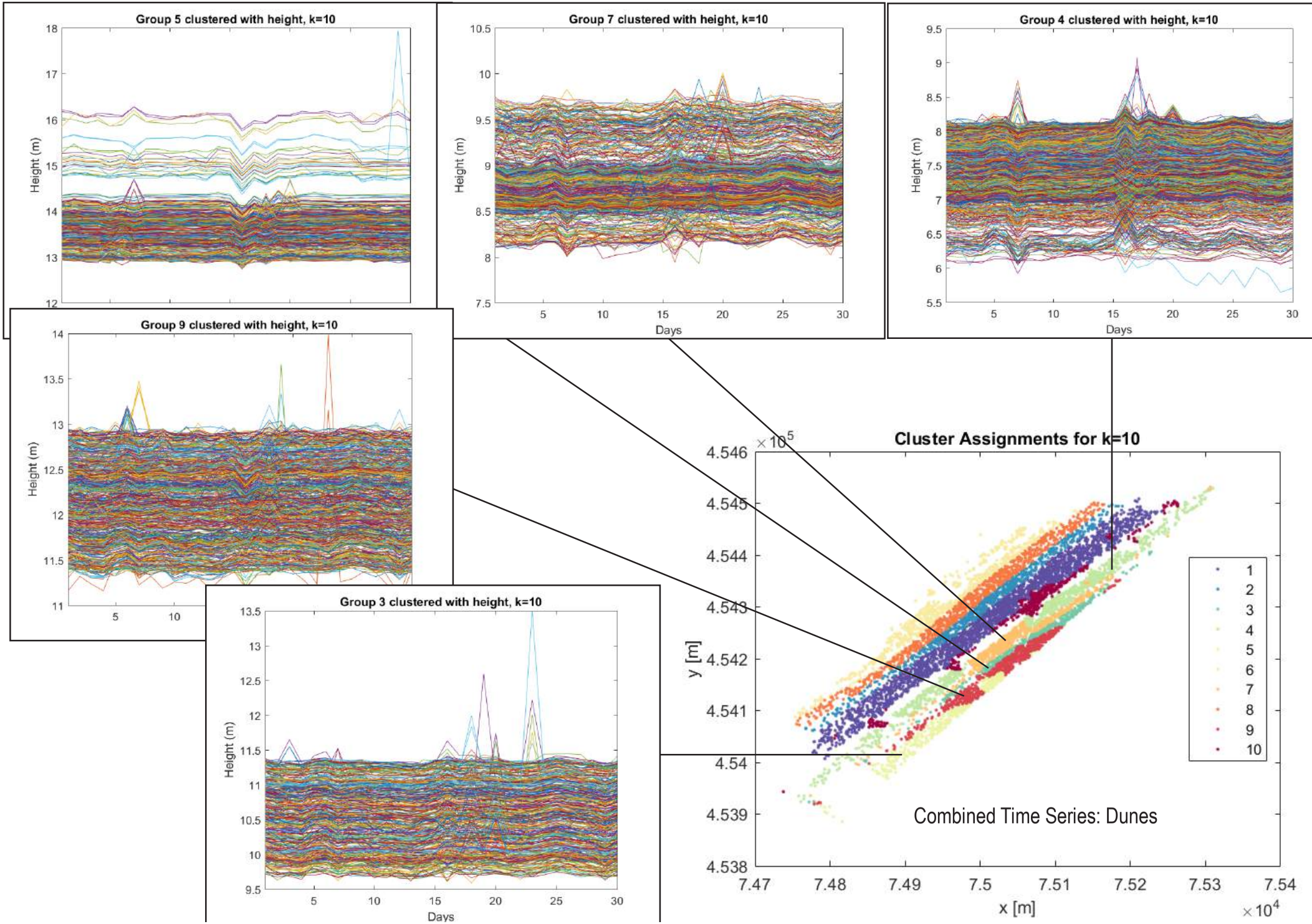

Group 1 clustered with height, k=10
Height (m)
Days


Group 2 clustered with height, k=10
Height (m)
Days

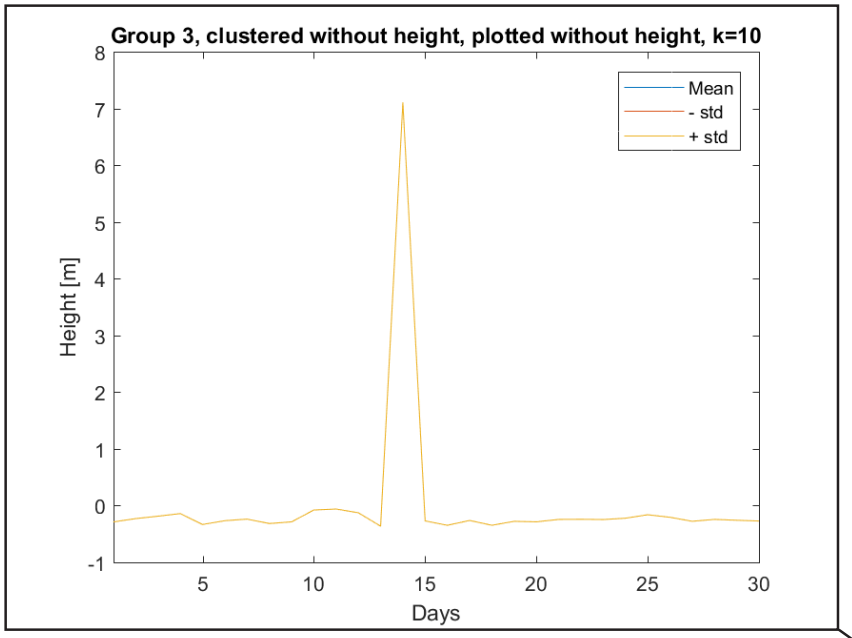

Group 10 clustered with height, k=10
Height (m)
Days

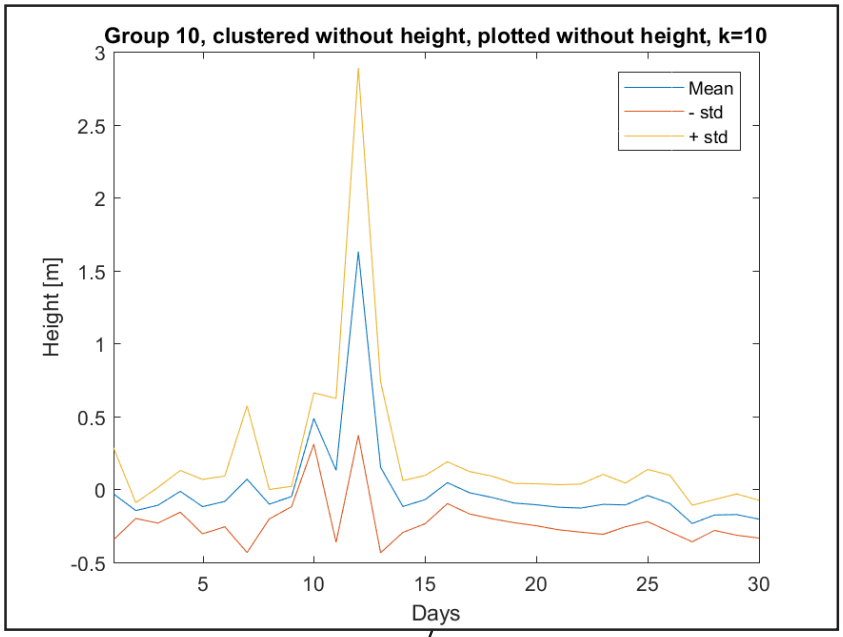

Cluster Assignments for k=10
y [m]
x [m]
Combined Time Series: Backshore
1
2
3
4
5
6
7
8
9
10

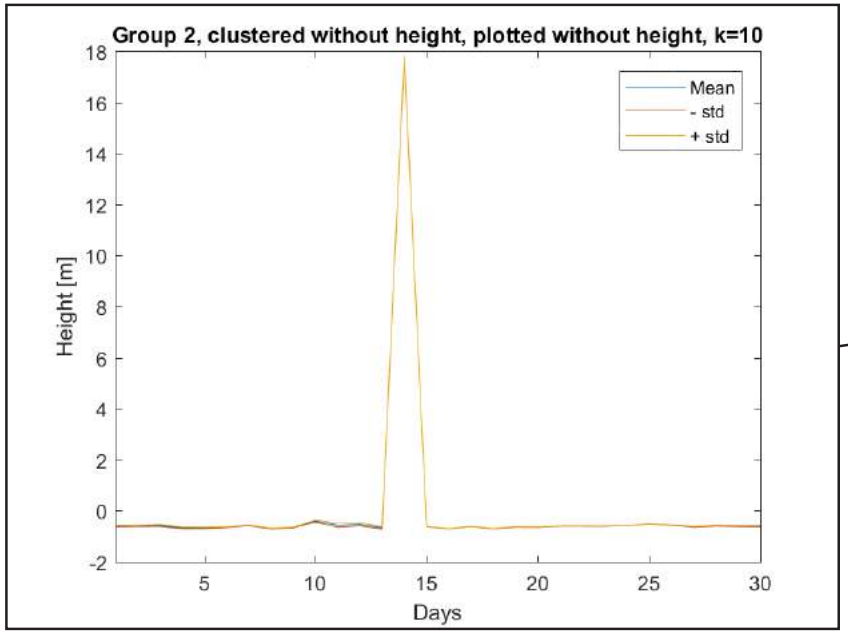

Group 5, clustered with height, k=10
Group 7, clustered with height, k=10
Group 4, clustered with height, k=10
Group 9, clustered with height, k=10
Group 3, clustered with height, k=10
Mean
- std
+ std
Height [m]
Days
Cluster Assignments for k=10
y [m]
x [m]
Mean Time Series: Dunes

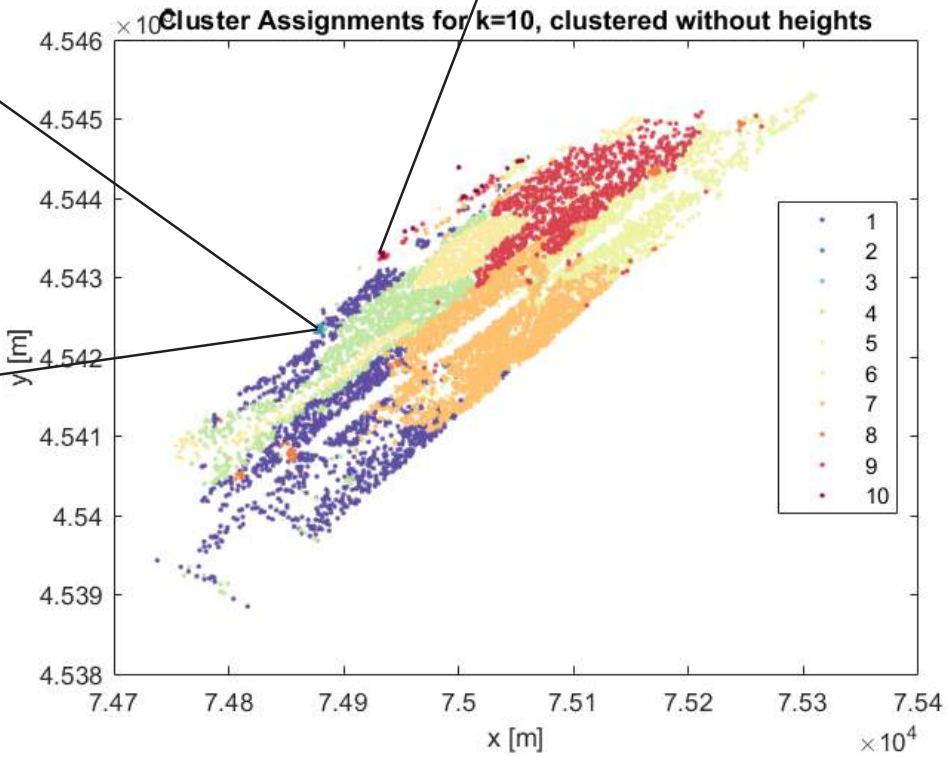

Group 5 clustered with height, k=10
Height (m)
Group 7 clustered with height, k=10
Height (m)
Days
Group 4 clustered with height, k=10
Height (m)
Days
Group 9 clustered with height, k=10
Height (m)
Group 3 clustered with height, k=10
Height (m)
Days
Cluster Assignments for k=10
y [m]
x [m]
Combined Time Series: Dunes
1
2
3
4
5
6
7
8
9
10

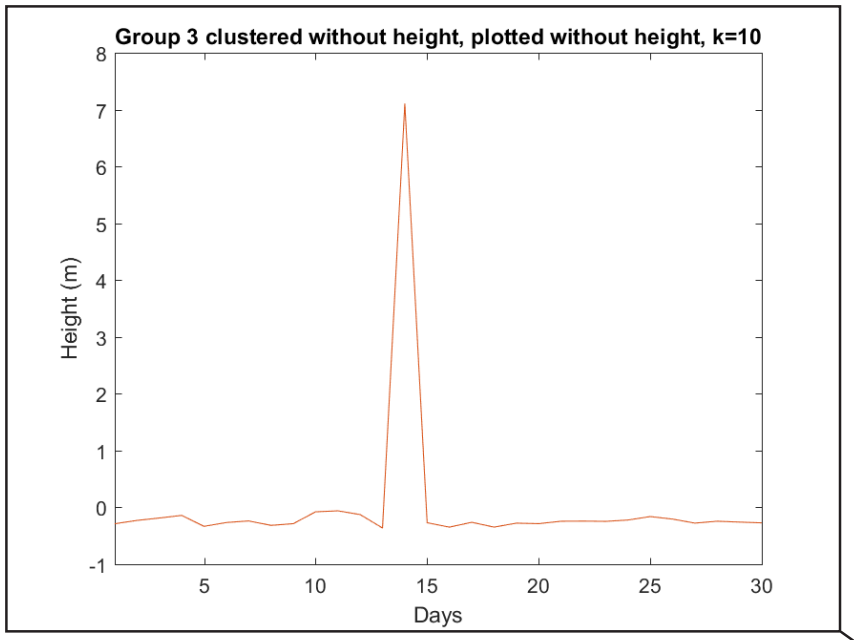
9.3 Appendix C - Shape-Based Clustering

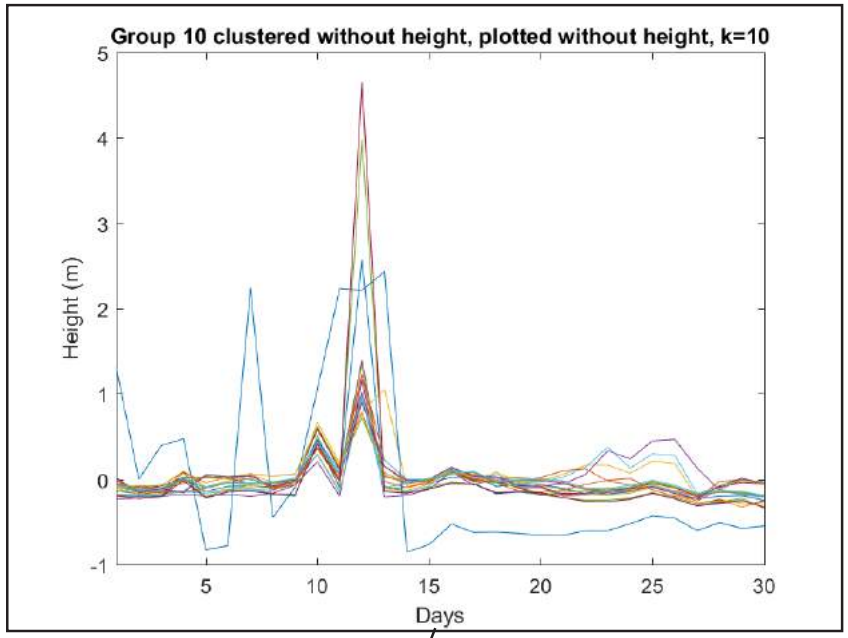

Group 3, clustered without height, plotted without height, k=10
Mean
- std
+ std
Height [m]
Days

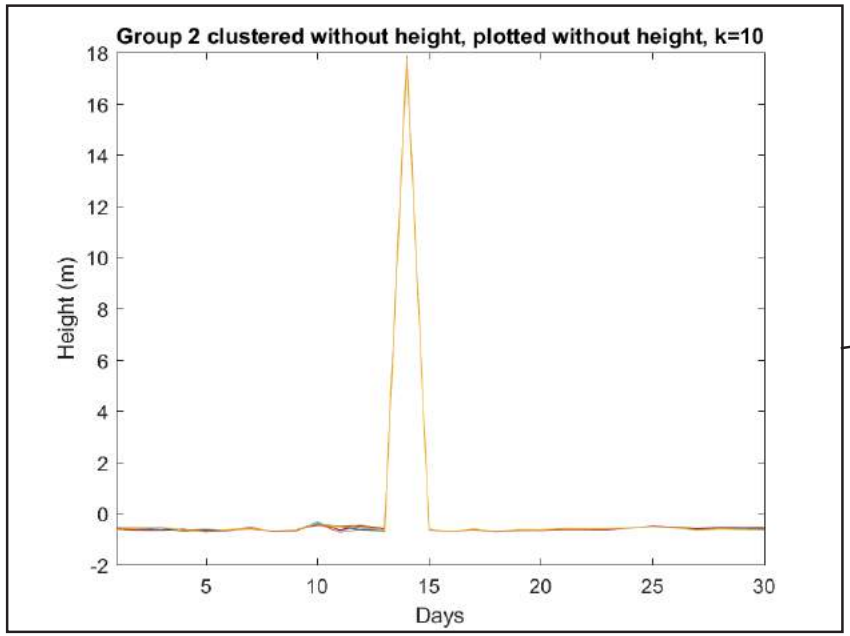

Group 10, clustered without height, plotted without height, k=10
Mean
- std
+ std
Height [m]
Days

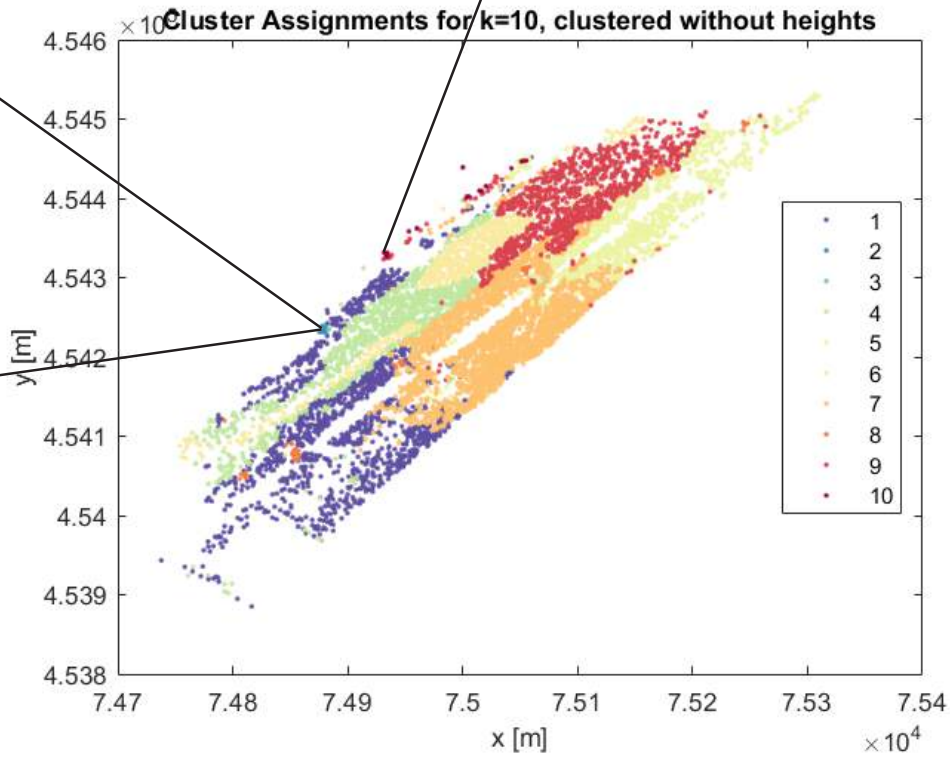

Group 2, clustered without height, plotted without height, k=10
Mean
- std
+ std
Height [m]
Days

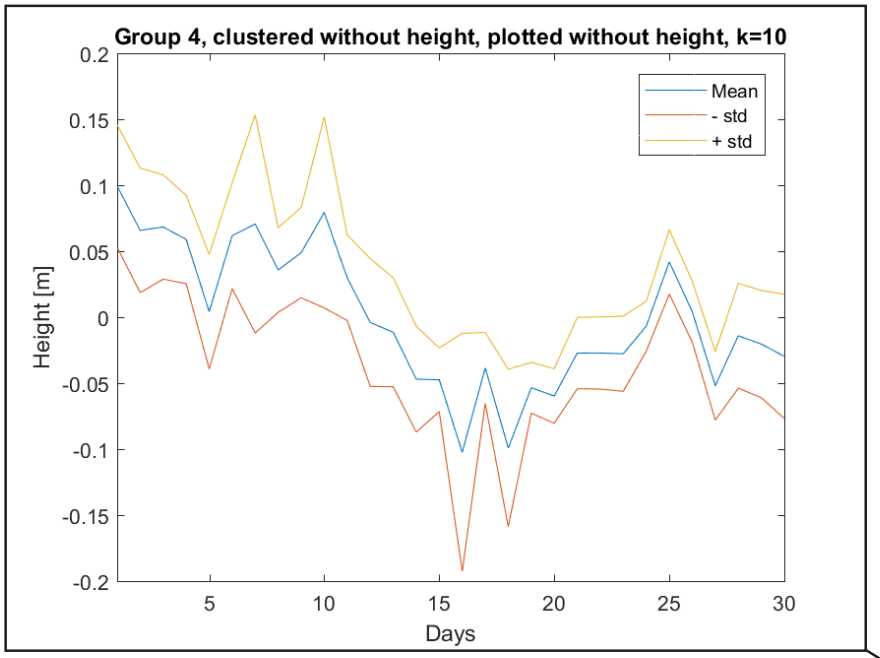

Cluster Assignments for k=10, clustered without heights
y [m]
x [m]
1
2
3
4
5
6
7
8
9
10

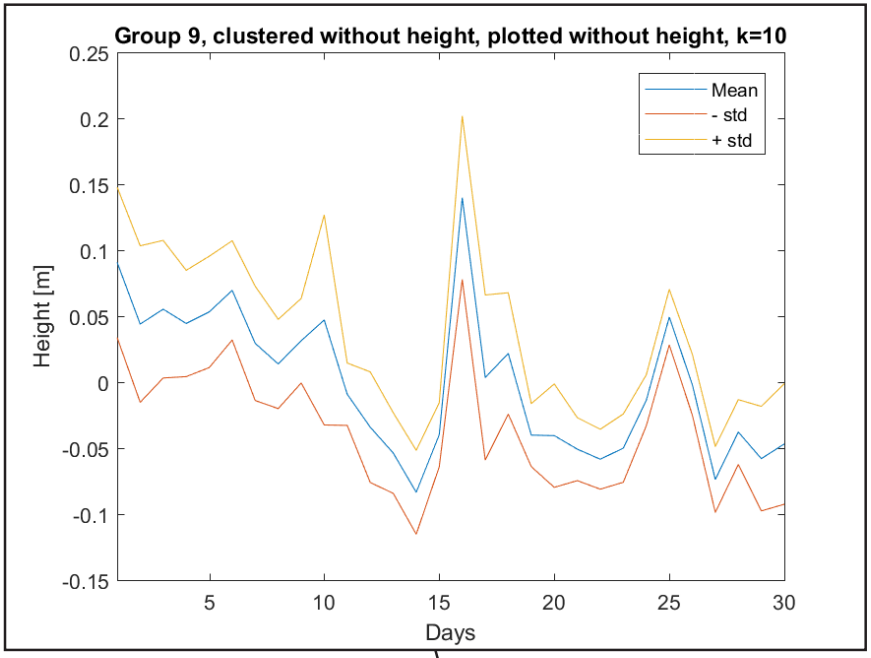

Group 3 clustered without height, plotted without height, k=10
Height (m)
Days

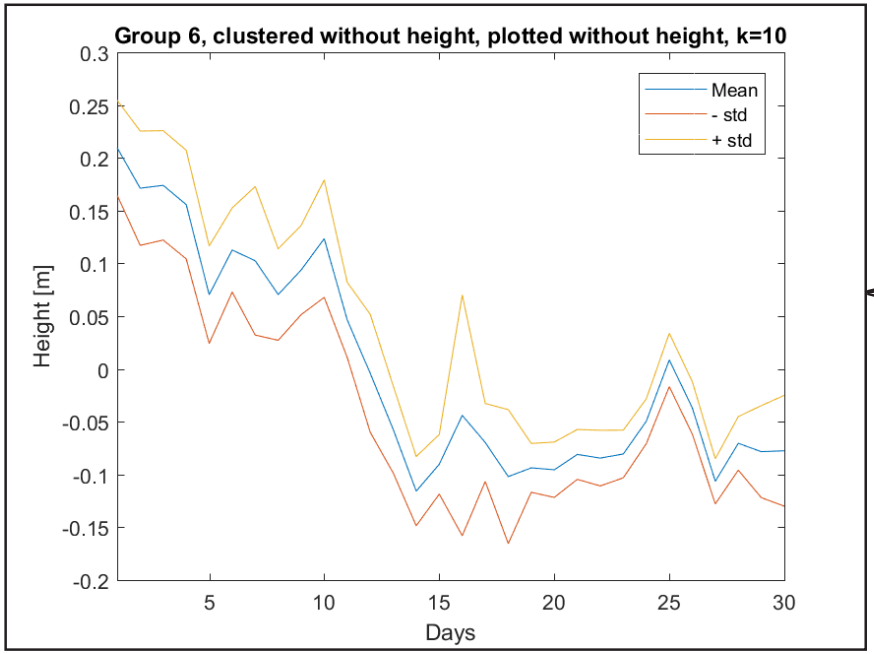

Group 10 clustered without height, plotted without height, k=10
Height (m)
Days

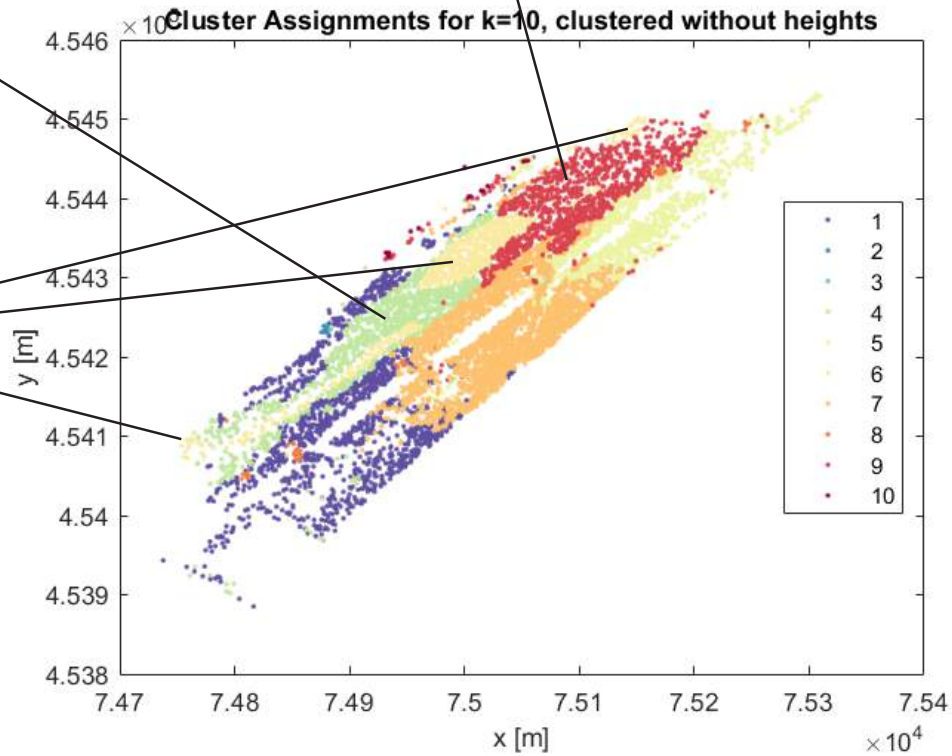

Group 2 clustered without height, plotted without height, k=10
Height (m)
Days

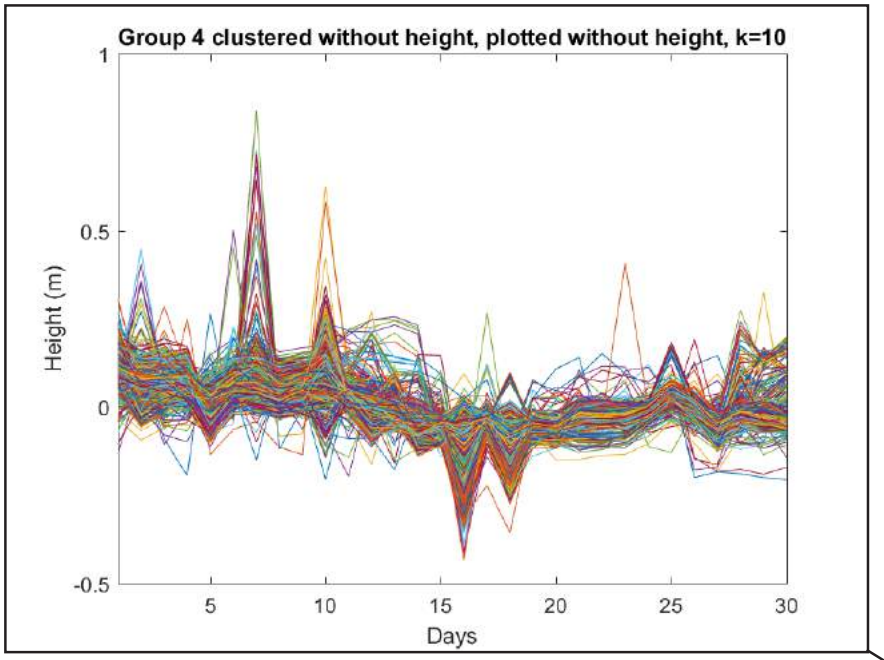

Cluster Assignments for k=10, clustered without heights
y [m]
x [m]
1
2
3
4
5
6
7
8
9
10

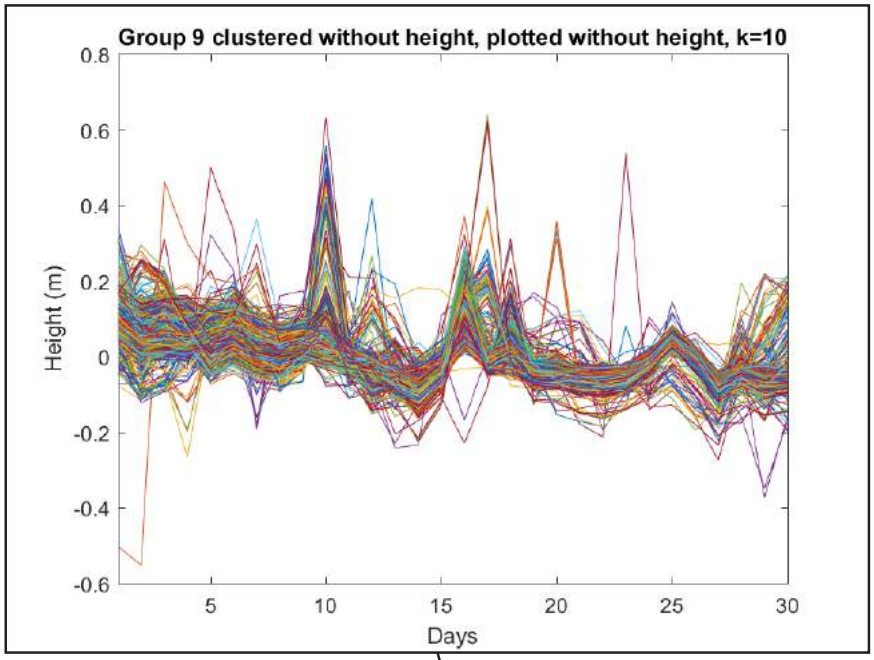

Group 4, clustered without height, plotted without height, k=10
Mean
- std
+ std
Height [m]
Days

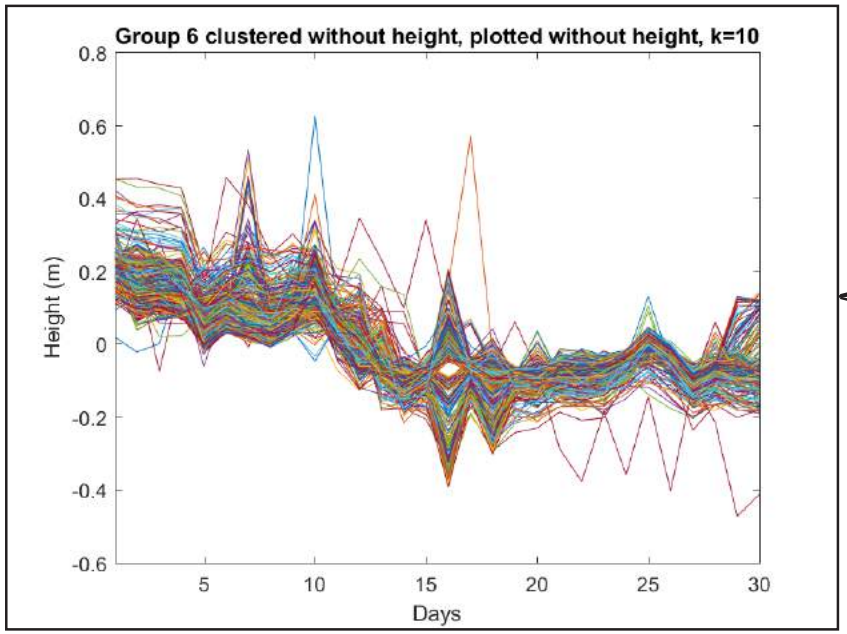

Group 9, clustered without height, plotted without height, k=10
Mean
- std
+ std
Height [m]
Days

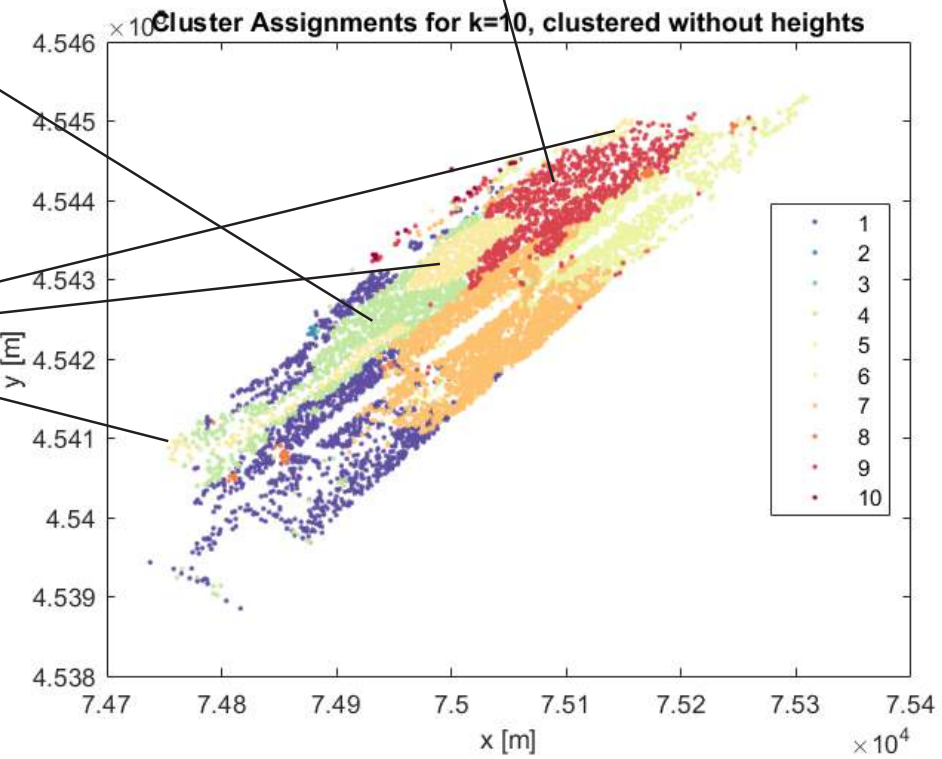

Group 6, clustered without height, plotted without height, k=10
Mean
- std
+ std
Height [m]
Days

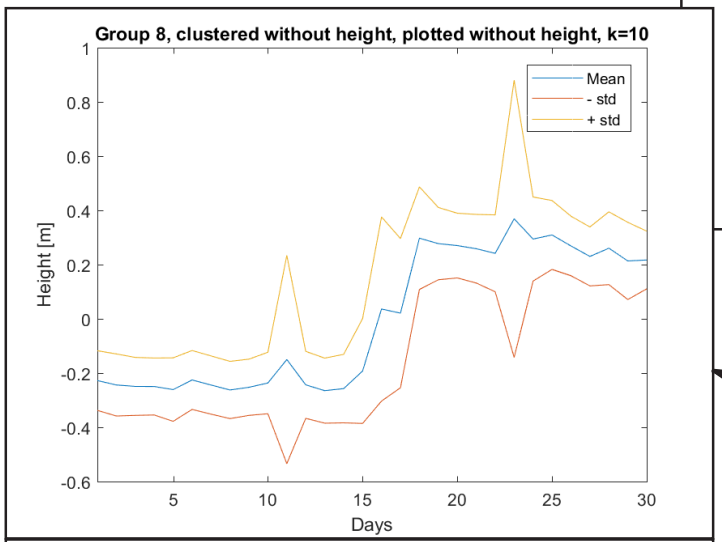

Cluster Assignments for k=10, clustered without heights
y [m]
x [m]
1
2
3
4
5
6
7
8
9
10

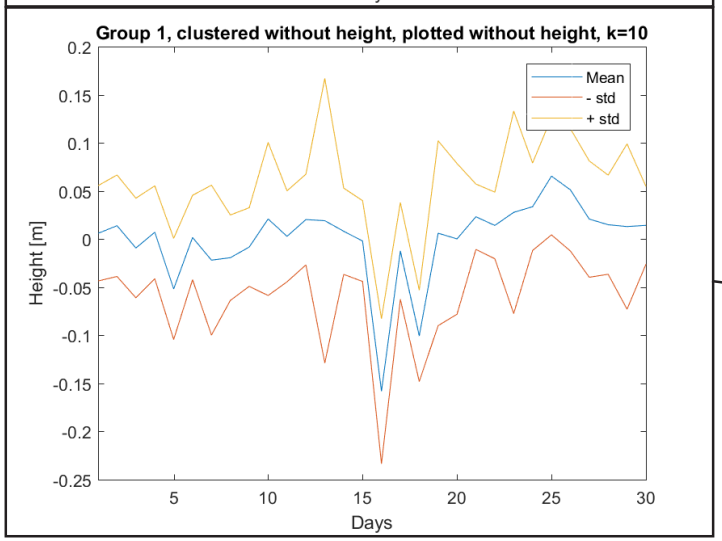

Group 4 clustered without height, plotted without height, k=10
Height (m)
Days

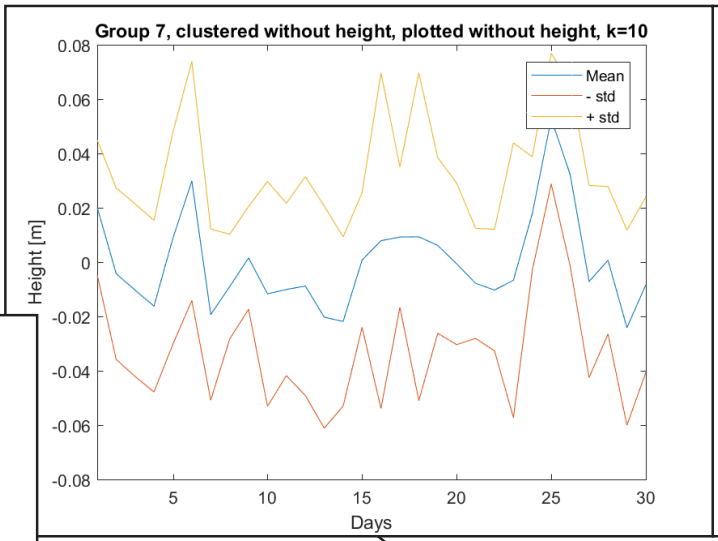

Group 9 clustered without height, plotted without height, k=10
Height (m)
Days

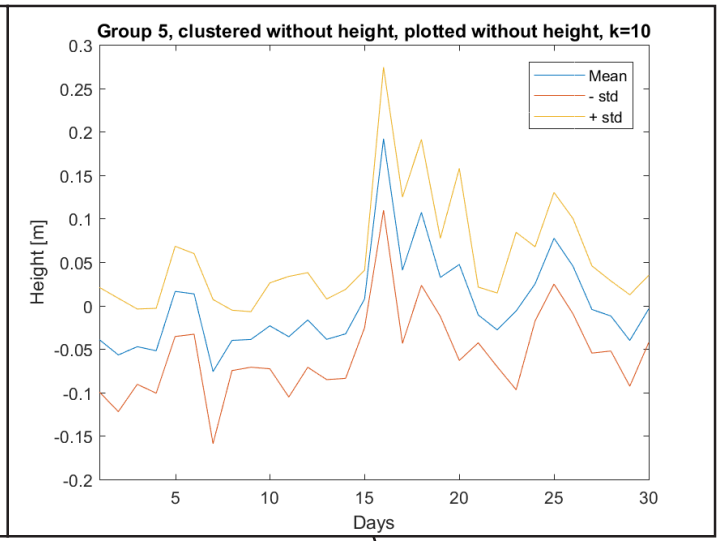

Group 6 clustered without height, plotted without height, k=10
Height (m)
Days

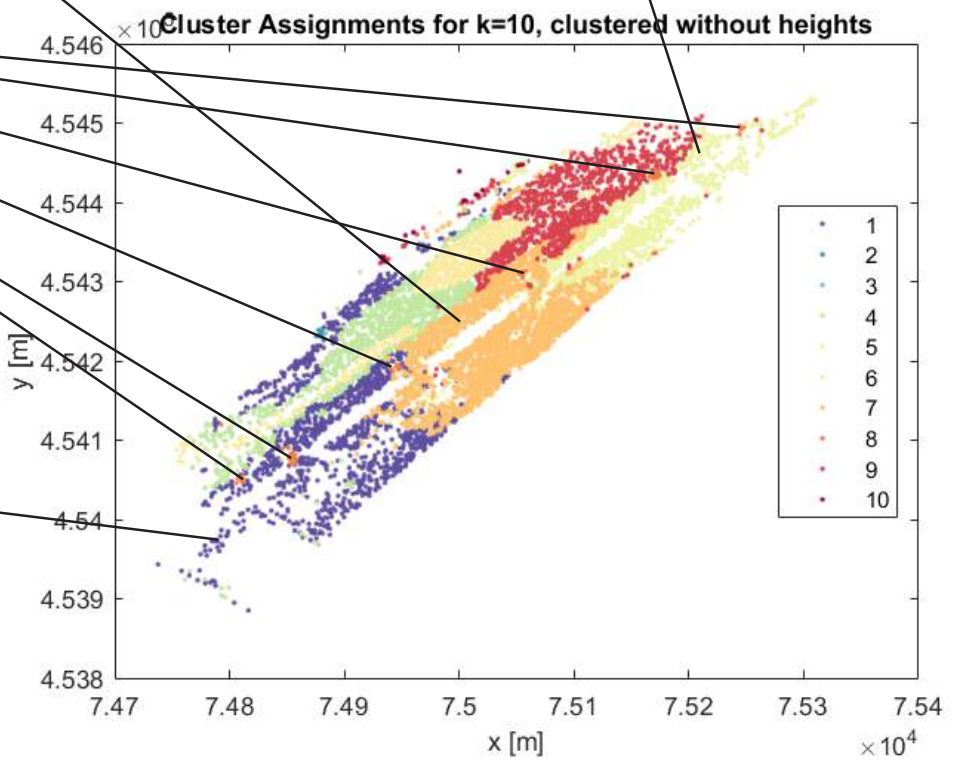

Cluster Assignments for k=10, clustered without heights
y [m]
x [m]
1
2
3
4
5
6
7
8
9
10

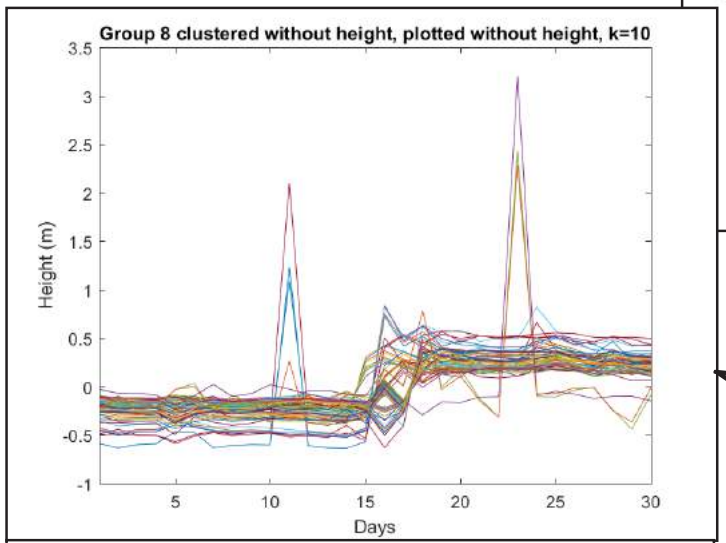

Group 8, clustered without height, plotted without height, k=10
Mean
- std
+ std
Height [m]
Days

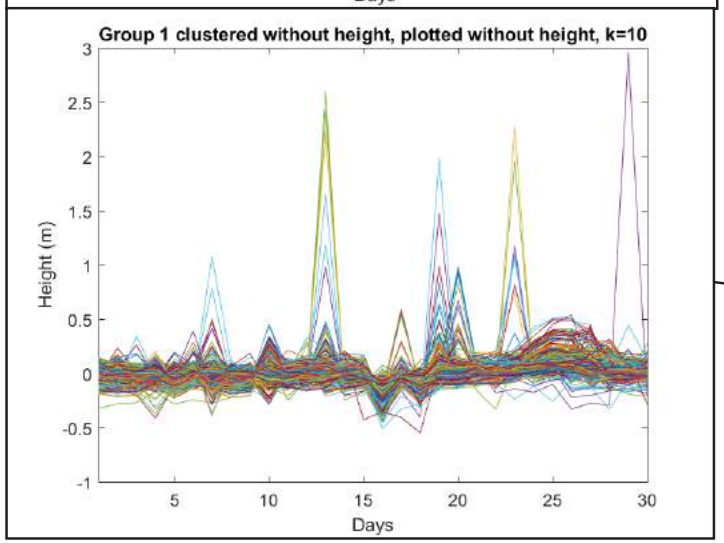

Group 1, clustered without height, plotted without height, k=10
Mean
- std
+ std
Height [m]
Days

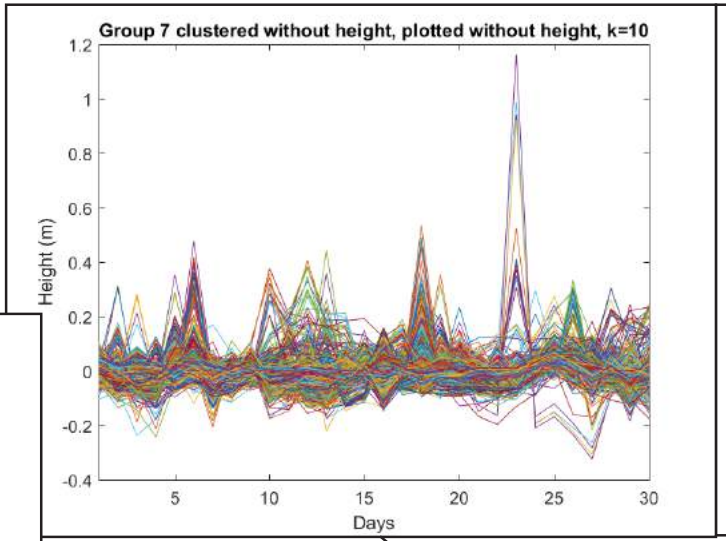

Group 7, clustered without height, plotted without height, k=10
Mean
- std
+ std
Height [m]
Days

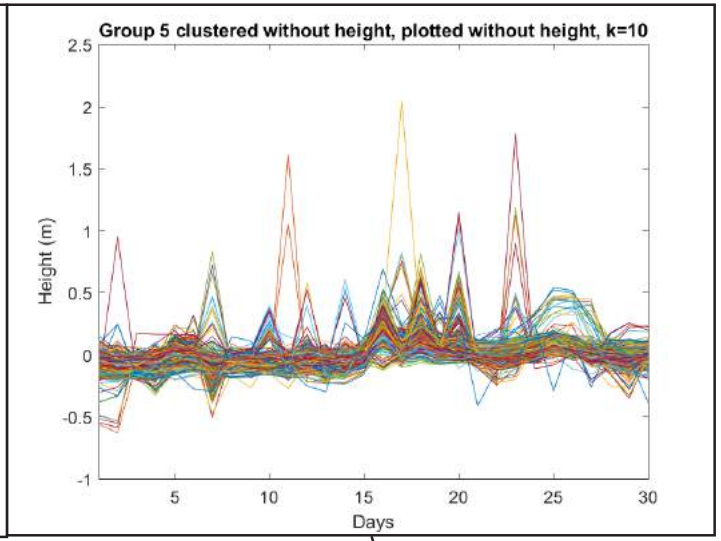

Group 5, clustered without height, plotted without height, k=10
Mean
- std
+ std
Height [m]
Days

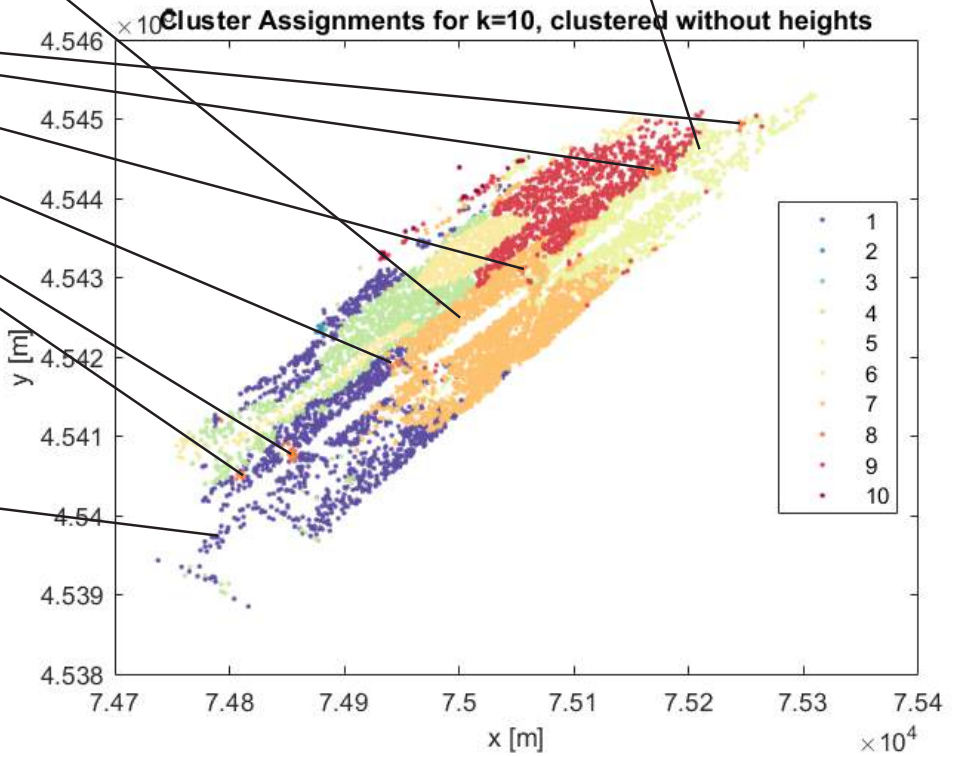

Cluster Assignments for k=10, clustered without heights
y [m]
x [m]
1
2
3
4
5
6
7
8
9
10


Group 8 clustered without height, plotted without height, k=10
Height (m)
Days


Group 1 clustered without height, plotted without height, k=10
Height (m)
Days


Group 7 clustered without height, plotted without height, k=10
Height (m)
Days


Group 5 clustered without height, plotted without height, k=10
Height (m)
Days


Cluster Assignments for k=10, clustered without heights
y [m]
x [m]
1
2
3
4
5
6
7
8
9
10

9.4 Appendix D - Photographs Taken On Site

Figure 37: Photograph overlooking the scanned beach, taken from the top of the middle beach entrance. Photograph was taken at Kijkduin beach on the 16th of October.

Figure 38: This photograph shows that in parts of the beach there is a large height difference between the end of the intertidal zone and the rest of the beach. This photograph was taken from the edge of the sea. Photograph was taken at Kijkduin beach on the 16th of October.

Figure 39: This photograph shows that in parts of the beach there is a large height difference between the end of the intertidal zone and the rest of the beach. This photograph was taken from the top of the berm. Photograph was taken at Kijkduin beach on the 16th of October.

Figure 40: This photograph shows the concrete slabs located around the beach entrances. These have been mostly cleared of sand. Photograph was taken at Kijkduin beach on the 16th of October.

Figure 41: This photograph shows that there can also be a lot of sand covering the concrete slabs located around the beach entrances. Photograph was taken at Kijkduin beach on the 16th of October.

Figure 42: Sudden height differences may also be caused through human intervention. Photograph was taken at Kijkduin beach on the 16th of October.

Figure 43: This photograph shows ripples on the beach. Photograph was taken at Kijkduin beach on the 16th of October.

Figure 44: This photograph shows tire tracks on the beach. Photograph was taken at Kijkduin beach on the 16th of October.

Figure 45: This photograph shows that parts of the dunes have been covered with sand, and as such the vegetation that was present during scanning cannot be seen any more. Photograph was taken at Kijkduin beach on the 16th of October.