Coastal Change Patterns from Time Series Clustering of Permanent Laser Scan Data

Mieke Kuschnerus[1], Roderik Lindenbergh[1], and Sander Vos[2]

[1]Department of Geoscience and Remote Sensing, Delft University of Technology
[2]Department of Hydraulic Engineering, Deflt University of Technology

Correspondence: M. Kuschnerus (m.kuschnerus@tudelft.nl)

Abstract. Sandy coasts are constantly changing environments governed by complex, interacting processes. Permanent laser scanning is a promising technique to monitor such coastal areas and to support analysis of geomorphological deformation processes. This novel technique delivers 3D representations of ~~a part of~~ the coast at hourly temporal and centimetre spatial resolution and allows to observe small scale changes in elevation over extended periods of time. These observations have the potential to improve understanding and modelling of coastal deformation processes. However, to be of use to coastal researchers and coastal management, an efficient way to find and extract deformation processes from the large spatio-temporal data set is needed. ~~In order to allow data miningin an automated way~~To enable automated data mining, we extract time series ~~in elevation or range~~ of surface elevation and use unsupervised learning algorithms to derive a partitioning of the observed area according to change patterns. We compare three well known clustering algorithms, k-means, agglomerative clustering and DBSCAN, apply them on the set of time series and identify areas that undergo similar evolution during one month. We test if ~~they~~ these algorithms fulfil our criteria for ~~a suitable clustering algorithm~~ suitable clustering on our exemplary data set. The three clustering methods are applied to time series ~~of~~ over 30 ~~epochs (during one month)~~ days extracted from a data set of daily scans covering ~~a part of the~~ about two km of coast at Kijkduin, the Netherlands. A small section of the beach, where a pile of sand was accumulated by a bulldozer is used to evaluate the performance of the algorithms against a ground truth. The k-means algorithm and agglomerative clustering deliver similar clusters, and both allow to identify a fixed number of dominant deformation processes in sandy coastal areas, such as sand accumulation by a bulldozer or erosion in the intertidal area. The DBSCAN algorithm finds clusters for only about 44% of the area and turns out to be more suitable for the detection of outliers, caused for example by temporary objects on the beach. Our study provides a methodology to efficiently mine a spatio-temporal data set for predominant deformation patterns with the associated regions, where they occur.

1 Introduction

Coasts are constantly changing environments that are essential to the protection of the hinterland from the effects of climate change and, at the same time, belong to the areas that are most affected by it. Especially long-term and small scale processes prove difficult to monitor but can have large impacts Aarninkhof et al. (2019). To improve coastal monitoring and knowledge of coastal deformation processes, a new technique called Permanent Laser Scanning (PLS) (also called continuous laser scanning) based on Light detection and ranging (LiDAR) measurements is available. For this purpose, a laser scanner is mounted on a high building close to the coast in a fixed location acquiring a 3D scan every hour during several months up to years.

The resulting spatio-temporal data set consists of a series of point cloud representations of a section of the coast. The high temporal resolution and long duration of data acquisition in combination with high spatial resolution (in the order of centimetres) provides a unique opportunity to capture a near continuous representation of ongoing deformation processes, like for example storm and subsequent recovery, on a section of the coast. As ~~analysed~~ reported by Lazarus and Goldstein (2019), the natural effects of a storm on a typical urban beach can rarely be analysed separately from anthropogenic activities, since in most cases work with bulldozers starts immediately after or even during severe storms. There is a need for the detection and quantification of change processes that influence the geomorphology of the coast, to allow understanding and modelling them, as the reaction of the coast to extreme weather events gains importance, Masselink and Lazarus (2019). More examples for potential use of such a data set are presented by O'Dea et al. (2019), who use data from a similar set-up in Duck, USA.

The PLS data set is large (in the order of hundreds of gigabytes), and to be relevant, the information on deformation processes has to be extracted concisely and efficiently. Currently there are no automated methods for this purpose and studies focus on one or a few two dimensional cross-sections through the data (for example O'Dea et al. (2019)). The high temporal resolution and long observation period lead to a high dimensional data set of ~~high dimensional~~ long time series (i.e. 30 ~~epochs~~ data points up to several thousands). Data mining on high dimensional data sets can be challenging as ~~concluded by Verleysen and François (2005)~~discussed for example by Zimek et al. (2012). In a first step towards extraction of interesting events and change patterns we build on the method introduced by Lindenbergh et al. (2019). We use clustering algorithms on time series representing the evolution of ~~the data set~~topography, to group ~~them~~ these time series according to their similarity in change pattern and then identify underlying processes. We use clustering (or unsupervised learning) to avoid having to specify the patterns and processes that we are looking for in advance.

One example of spatio-temporal segmentation on our data set from permanent laser scanning was recently developed by Anders et al. (2020). They detected seed points for deformation in time series from permanent laser scanning, to then grow a region affected by the detected change around the seed points with the use of dynamic time warping distance to spatial neighbours. Dynamic time warping is a distance measure between time series, that accounts for similarity in patterns even though they might be shifted in time (see for example Keogh and Ratanamahatana (2005)). One drawback of this approach is that temporal patterns of interest have to be defined before hand, and therefore only deformation patterns that are expected can be found. Another approach to model spatio-temporal deformations in point clouds from laser scanning, is presented by Harmening and Neuner (2020). Their model assumes that the deformation can be represented by a continuous B-spline surface.

This approach could potentially be used to further analyse some of the deformation patterns found in our study but does not allow the exploratory data mining, that we are aiming to accomplish. A more general overview of methods to find spatio-temporal patterns in earth science data was published by Tan et al. (2001) and a continuation of this study was presented by Steinbach et al. (2001). The study of Tan et al. deals with pre-processing of time series of different variables from satellite data including issues with auto-correlation and seasonality. Steinbach et al. successfully apply a novel clustering technique introduced by Ertöz et al. (2003) to explore spatio-temporal climate data. However, this technique only focuses on contiguous clusters, where all time series are in a close neighbourhood to each other, and does not allow to find general patterns independent of location.

Time series data sets are also used to asses patterns of agricultural land use by Recuero et al. (2019). For this study time series of Normalized Difference Vegetation Index (NDVI) data have been analysed using auto-correlation values and random forest classification. Benchmark data from an alternative source was needed to train the classifier. Such benchmark data is currently not available in our case. A study by Belgiu and Csillik (2018) used time series from Sentinel-2 satellite data for cropland mapping. They made use of dynamic time warping classification and showed that in areas with little available reference data for training a classifier, their approach delivers good results in segmentation based on time series' evolution. Also in this case pre-labelled training data is required. Another approach using expectation-based scan statistics was presented by Neill (2009): To detect spatial patterns in time series from public health data, a statistical method based on expectation values is used. Clusters are formed where the observed values significantly exceed the expectation. The results are promising but depend on the choice of time series analysis method, statistics used and the shape of the search region, which all have to be defined in advance specific to each data set and application. Generally there is a lack of studies on mining spatio-temporal data for deformation patterns, without using training data or predefined change patterns.

The goal of the present study is to evaluate the application of clustering algorithms on a high dimensional spatio-temporal data set without specifying deformation patterns in advance. Our objectives in particular are:

1. To analyse and compare the limits and advantages of three clustering algorithms for separating and identifying change patterns in high dimensional spatio-temporal data.

2. To detect specific deformation on sandy beaches by clustering time series from permanent laser scanning.

We compare the k-means algorithm, agglomerative clustering and the DBSCAN algorithm on a PLS data set ~~of~~ over 30 ~~epochs~~days, to investigate the effectiveness of the identification of coastal change patterns. All three algorithms are well established and represent three common but different approaches to data clustering. To determine if an algorithm is suitable, we expect that it fulfils the following criteria:

- A majority of the observation area is separated into distinct regions,
- each cluster shows a change pattern that can be associated with a geomorphic deformation process, and
- time series contained in each cluster roughly follow the mean change pattern.

~~Additionally, we compare two representations of time series: First we use time series extracted from a grid in Cartesian coordinates as elevation per grid cell and second time series from a grid in spherical coordinates represented by range per grid cell which is a more native way of representing laser scanner data.~~ We use the different clustering approaches on a small area of the beach at the bottom of a footpath, where sand accumulated after a storm, and a bulldozer subsequently cleared the path and formed a pile of sand. We determine the quality of the detection of this process for both algorithms and compare them in terms of standard deviation within the clusters and area of the beach covered by the clustering. We compare and evaluate the resulting clusters using these criteria as a first step towards the development of a method to mine the entire data set from permanent laser scanning for deformation processes.

2 The permanent laser scan data set

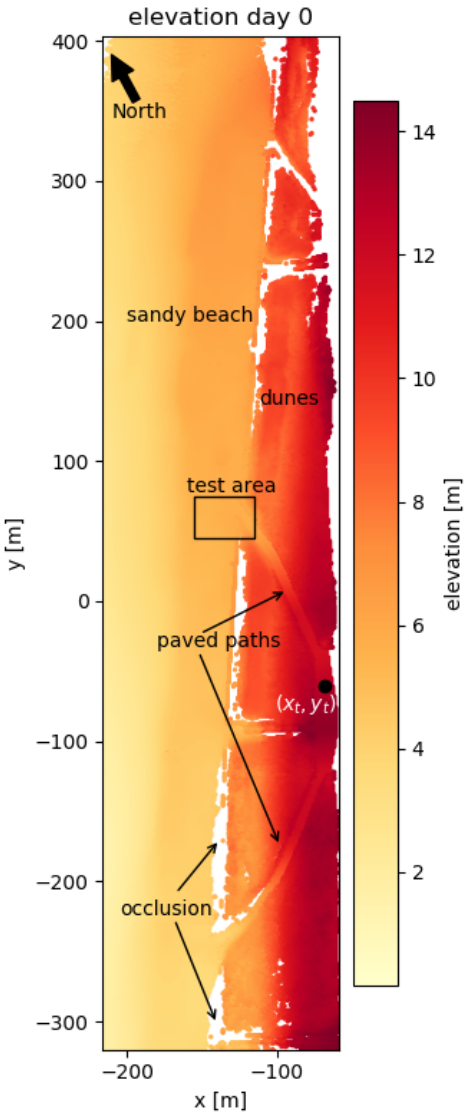

Figure 1. Top view of a point cloud representing the observation area at low tide on 1st January ~~2019. Colours represent elevation from low (yellow, about sea level) to high~~ 2017. The laser scanner is located at the origin of the coordinate system (~~grey, about 14 above sea level and higher~~not displayed). The point ~~(x_0, y_0)~~ (x_t, y_t) indicates the location of the time series shown as an example in Figure 3. The test area, which is discussed in Section 3.4, is indicated ~~in red~~ with a box at the end of the northern path leading to the beach. The paved paths leading to the beach are used as stable reference surface for the errors reported in Table 2. Parts that are white between the dunes and the sandy beach are gaps in the data due to occlusions caused by the dunes.

The data set from permanent laser scanning is acquired within the CoastScan project at a typical urban beach in Kijkduin, the Netherlands, Vos et al. (2017). For the acquisition a Riegl VZ-2000 laser scanner was used to scan over a period of six months from December 2016 to May 2017. The full data set consists of hourly scans of a section of sandy beach and dunes.

Figure 2. Riegl VZ2000 laser scanner mounted on the roof of a hotel facing the coast of Kijkduin, the Netherlands. The scanner is covered with a protective case to shield it from wind and rain.

For the present study, a subset of the available data is used to develop the methodology. This subset consists of 30 daily scans taken at low tide over a period of one month, January 2017. It covers a section of the beach and dunes in Kijkduin and is displayed in top view in Figure 1. The area contains a path and stairs leading down to the beach, a paved area in front of the dunes, a fenced in dune area and the sandy beach. It is about 950 m long, 250 m wide and the distance from the scanner to the farthest points on the beach is just below 500 m. For the duration of the experiment the scanner was mounted on the roof of a hotel just behind the dunes at a height of about 37 m above sea level (as shown in Figure 2).

The data is extracted from the laser scanner output format and converted into a file that contains xyz-coordinates and spherical coordinates for each point. The data is mapped into a local coordinate system, where the origin in x- and y-direction is at the location of the scanner and the height (z-coordinate) corresponds to height above sea level. Since we are interested in relative changes between consecutive scans, we do not transform the data into a geo-referenced coordinate system for this analysis.

Each point cloud is chosen to be at the time of lowest tide between 18:00 and 06:00, in order to avoid people and dogs on the beach, with the exception of two days where only very few scans were available due to maintenance activities. The data from 9th of January 2017 is entirely removed from the data set, because of poor visibility due to fog. This leads to the 30 day

data set, numbered from 0 to 29. Additionally all points above 14.5 m elevation are removed to filter out points representing the balcony of the hotel and flag posts along the paths~~that are interfering with the spherical coordinate grid extraction~~. In this way also a majority of reflections from particles in the air, birds or raindrops are removed. However, some of these particles might still be present at lower heights close to the beach.

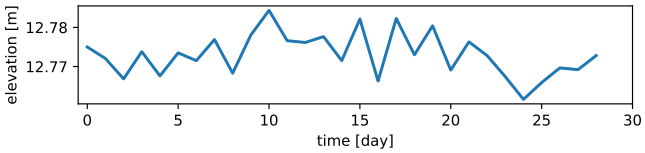

Figure 3. Time series ~~in~~ of elevation at location ~~(x_0, y_0)~~ (x_t, y_t) (marked in Figure 1) on the path that is assumed to be stable throughout the entire month. Elevation is varying within less than 2~~cm~~ cm.

Since the data is acquired from a fixed and stable position we assume that consecutive scans are aligned. Nevertheless, the orientation of the scanner may change slightly due to strong wind, sudden changes in temperature, or maintenance activities. The internal inclination sensor of the scanner measures these shifts while it is scanning and we apply a correction for large deviations (more than 0.01 degrees) from the median orientation.

The remaining error in elevation ~~and range~~ is estimated as the ~~average standard deviation of time series in locations that are assumed to be stable during the entire month~~standard error and the 95-percentile of deviations from the mean elevation over all grid cells included in the stable paved area. We chose ~~stable surfaces that are~~ the stable surface that is part of the paved paths on top of the dunes and leading to the beach in northern and southern direction ~~and derived the mean remaining errors shown in Table~~as indicated in Figure 1. This area includes 1653 grid cells with complete time series. The derived mean elevation, standard error and overall 95-percentile of deviations from the mean per time series averaged over the stable area are reported in Table 2. The elevation does on average not deviate more then ~~2~~ 1.4 cm from the mean elevation~~of the respective area and the standard deviation of the range is within 5 on the northern path and on top of the dunes, but around 11~~, and 95 % of deviations from the mean elevation are on average below 3.5 cm~~on the southern path~~. An example time series from the stable paved area on top of the dunes (at location ~~(x_0, y_0)~~ (x_t, y_t) marked in Figure 1) is shown in Figure 3.

3 Methods

To derive coastal deformation processes from clusters based on change patterns we follow three steps: Extraction of time series~~in two different coordinate systems~~, clustering of time series with three different algorithms, and derivation of geomorphological deformation processes. To cluster time series the definition of a distance between two time series (or the similarity) is not immediately obvious. We discuss two different options (Euclidean distance and correlation) to define distances between time series with different effects on the clustering results. The rest of this section is organized as follows: We focus on time

series extraction in subsection 3.1, discuss distance metrics for time series (3.2), introduce three clustering algorithms (3.3) and our evaluation criteria (3.4). The derivation of deformation processes will be discusses with the results (section 4).

3.1 Time Series Extraction

Time series of surface elevation are extracted from the PLS data set ~~in two different ways: First~~ by using a grid in Cartesian xy-coordinates~~and extracting time series in elevation and second by using a grid in spherical coordinates and extracting time series in range. For both methods, we~~ . We only use grid cells that contain at least one point for each of the scans.

~~Regular grid in Cartesian coordinates (A) and in spherical coordinates (B). In spherical coordinates (B) the size of the grid cells is larger, the further away they are from the scanner because of the large incidence angle between the line of sight of the scanner and the normal of the sloping surface of the beach.~~

3.1.1 ~~Cartesian Coordinates~~

Before defining a grid ~~in Cartesian coordinates~~on our observed area, we rotate the observation area to make sure that the coastline is parallel to the y-axis, as shown in Figure 1. This ensures that the grid covers the entire observation area efficiently and leaves as few empty cells as possible. Then we generate a regular grid ~~(as illustrated in Figure ??)~~ with grid cells of 1 $\mathrm{m} \times$ 1~~m~~ m. Time series are generated for each grid cell by taking the median elevation z_i for each grid cell and for each time stamp t_k. That means, per grid cell with center (x_i, y_i) we have a time series

$$\tilde{Z}_i = (z_i(t_1), \dots z_i(t_T)), \tag{1}$$

Table 1. ~~Average standard deviation of the gridded elevation and range in listed areas, which are each assumed to be stable throughout the observation period of one month.~~

~~mean error~~	~~elevation~~	~~range~~
~~paved area on top of dunes~~	~~1.47 cm~~	~~4.89 cm~~
~~path leading to the beach (north)~~	~~1.12 cm~~	~~4.03 cm~~
~~path leading to the beach (south)~~	~~1.65 cm~~	~~10.9 cm~~

Table 2. Test statistics of the gridded elevation values on the paved area, which is assumed to be stable throughout the observation period of one month. Values are calculated per time series and averaged over the entire stable area, which results in mean elevation, standard error and an average 95-percentile of deviations from the mean.

mean elevation	12.43 m
standard error	1.4 cm
95-percentile of deviation from mean (averaged over all grid cells)	3.5 cm

with the number of time stamps $T = 30$. To make the time series dependent on change patterns, rather than the absolute elevation values, we remove the mean elevation $\bar{z}_i$ of each time series $\tilde{Z}_i$. This leads to time series

$$Z_i = (\Delta z_i'(t_1), \ldots, \Delta z_i'(t_T), \tag{2}$$

with $\Delta z_i(t_k) := z_i(t_k) - \bar{z}_i z_i'(t_k) := z_i(t_k) - \bar{z}_i$.

This approach results in a collection of time series that represent equally sized grid cells . However, the In this way we extract around 40 000 grid cells that contain complete elevation time series for the entire month. The point density per grid cell varies depending on distance to the laser scanner. For example, a grid cell on the paved path (at about 80 m range) contains about 40 points (i.e. time series at (x_0, y_0) (x_t, y_t) in Figure 1), whereas a grid cell located close to the water line, at about 300 m distance from the scanner, may contain around three values. This implies that the median per grid cell is based on more points the closer a grid cell is to the scanner.

3.1.1 Spherical Coordinates

Using spherical coordinates allows to generate a grid with constant angle increment in horizontal and vertical direction with roughly constant point density per grid cell. For each grid cell j, we derive a time series R_j consisting of the median range $r_j(t_k)$ per grid cell per time stamp t_k:

$$R_j = (r_j(t_1), \ldots, r_j(t_T)),$$

where $T = 30$, as above. The range r_j is defined as the line of sight distance from the laser scanner to the respective point. We choose grid cells of $0.1° \times 0.375°$ to ensure that grid cells on the beach (close to the dune foot) cover roughly 1 m^2, the same as in Cartesian coordinates, in order to make both methods comparable. However, transformed back into Cartesian coordinates, the size of the grid cells (in square-meters) varies with distance from the scanner (see Figure **??**).

The point density in a point cloud is generally lower, the farther away a point is from the scanner. This property of our data set is preserved using spherical coordinates and represented in the size of the grid cell, or distance between grid cell centres.

3.2 Distance Metrics

We consider two different distance metrics for our analysis: the Euclidean distance as the default for the k-means algorithm and agglomerative clustering, and correlation distance for the DBSCAN algorithm.

3.2.1 Euclidean Distance

The most common and obvious choice is the Euclidean distance metric defined as:

$$d_E(Z_0, Z_1) = ||Z_0 - Z_1|| = \sqrt{\sum_{i=1}^{n} |Z_{0i} - Z_{1i}|^2}, \tag{3}$$

for two time series Z_0 and Z_1 of length n.

3.2.2 Correlation Distance

Another well known distance measure is correlation distance, defined as one minus the Pearson correlation coefficient (see for example Deza and Deza (2009)). It is a suitable measure of similarity between two time series, when correlation in the data is expected (see Iglesias and Kastner (2013)). Correlation between two time series Z_0 and Z_1 is defined as:

$$\text{Cor}(Z_0, Z_1) = 1 - \frac{(Z_0 - \bar{Z}_0) \cdot (Z_1 - \bar{Z}_1)}{||Z_0 - \bar{Z}_0|| \cdot ||Z_1 - \bar{Z}_1||}, \tag{4}$$

with $\bar{Z}$ being the mean value of time series Z and $||\ \cdot\ ||$ the Euclidean 2-norm as in Equation (3). We have to note here, that correlation cannot compare simple constant time series (leads to division by zeros) and is therefore not a distance metric in the sense of the definition Deza and Deza (2009).

3.2.3 Comparison

For a comparison of the two distances for some example time series~~see Figure~~, see Figure 4. The example shows that the distance between two time series is not intuitively clear. The use of different distance metrics results in different sorting of distances between the shown pairs of time series. ~~However, when~~ When normalizing all time series (subtracting the mean and scaling by the standard deviation) correlation distance and Euclidean distance are equivalent (as shown for example by Deza and Deza (2009)). However, this leads to issues, when comparing to a constant time series (with zero standard deviation).

Both Euclidean distance and correlation are not taking into account the order of the values within each time series. For example, two identical time series that are shifted in time are seen as 'similar' with the correlation distance, but not as similar with the Euclidean distance and would not be considered as identical by either of them (see Figure 4). Additionally neither of the two distance metrics can deal with time series of different lengths or containing gaps.

3.3 Clustering Methods

Clustering methods for ~~Time Series~~ time series can be divided into two categories: feature based and raw data based (see for example Liao (2005)). Feature based methods first extract relevant features to reduce dimensionality (for example using Fourier- or wavelet-transforms) and then form clusters based on these features. They could also be used to deal with gaps in time series. We focus on the raw data based approach to not define features in advance and to make sure that no information within the data set is lost. We use three different methods: k-means clustering, agglomerative clustering and Density-Based Spatial Clustering of Applications with Noise (DBSCAN). In Figure 5 an illustration of a partitioning of a simple 2D data set is shown for each of the three algorithms. The two clusters that can be distinguished in this example have different variances and are grouped differently by each of the algorithms.

For the implementation of all three algorithms, we make use of the Scikit-learn package in Python (see Pedregosa et al. (2011)).

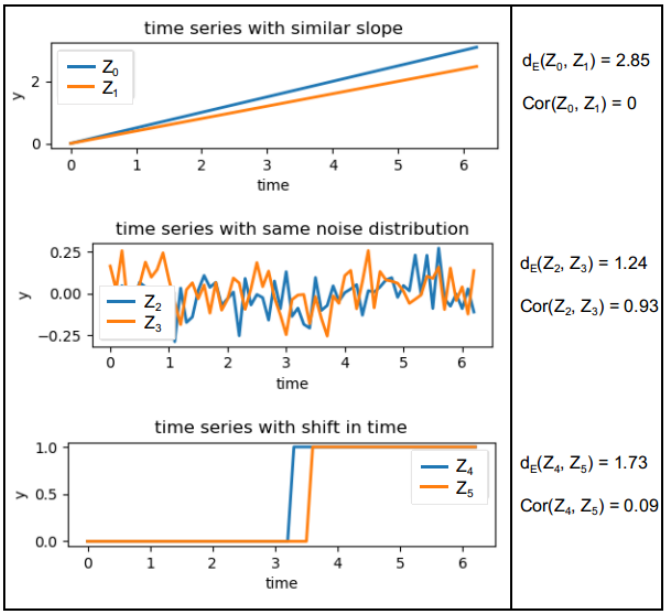

Figure 4. Example of three pairs of time series that are 'similar' to each other in different ways. The Euclidean distance would sort the differences as follows $d_E(Z_2, Z_3) < d_E(Z_4, Z_5) < d_E(Z_0, Z_1)$, whereas according to the correlation distance the order would be $\text{Cor}(Z_0, Z_1) < \text{Cor}(Z_2, Z_3) < \text{Cor}(Z_4, Z_5)$.

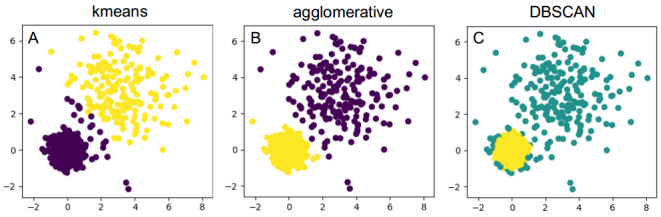

Figure 5. Example of clustering of data with two clusters with different variance: The k-means algorithm separates them, but adds a few points in the middle to the purple cluster instead of the yellow one (A). Agglomerative clustering separates both clusters according to their variances (B) and DBSCAN detects the cluster with low variance and high point density (yellow) and discards all other points as outliers (turquoise) (C).

3.3.1 k-means Clustering

The k-means algorithm was first introduced in 1955 and is still one of the most widely used clustering methods (Jain (2010)). The algorithm is based on minimizing the sum of all distances between points and centroids over all possible choices of k

cluster centroids $V = \{v_1, \ldots, v_k\}$:

$$\text{Min}_V J(V) = \sum_{j=1}^{k} \sum_{x_i \in v_j} ||x_i - v_j||^2, \tag{5}$$

with Euclidean distance metric $|| \cdot ||$. After the initial choice of k centroids among all points the following steps are repeated iteratively, until the above sum does not change significantly:

1. Assign each point to the cluster with closest centroid
2. Move centroid to mean of each cluster
3. Calculate sum of distances over all clusters (Equation (5))

Note that minimizing the squared sum of distances over all clusters, coincides with minimizing the squared sum of all within cluster variances. The convergence to a local minimum can be shown for the use of Euclidean distance (see for example Jain (2010)). The convergence is sped up using so-called k-means++ initialization: After the random selection of the first centroid, all following centroids are chosen based on a probability distribution proportional to their squared distance to the already defined centroids. In this way the initial centroids are spread out throughout the data set and the dependence on the random initialization of the cluster centroids is reduced.

There are variations of k-means using alternative distance metrics such as the L^1-norm (k-medoids, Park and Jun (2009)), however the convergence is not always ensured in these cases. Another issue to take into account when considering alternative distance metrics, is the definition of the cluster centroids as mean of time series, which is not automatically defined for any distance metric. For more information on k-means see Jain (2010), Liao (2005) and the documentation of the Scikit-learn package (Pedregosa et al. (2011)).

3.3.2 Agglomerative Clustering

Agglomerative clustering is one form of hierarchical clustering: It starts with each point in a separate cluster and iteratively merges clusters together until a certain stopping criterion is met. There are different variations of agglomerative clustering using different input parameter and stopping criteria (see for example Liao (2005) or the documentation of the scikit-learn package (Pedregosa et al. (2011))). We choose the minimization of the sum of the within cluster variances using the Euclidean distance metric (Equation (5), where the centroids v_j are the mean values of the clusters) for a pre-defined number of clusters k. The algorithm starts with each point in a separate cluster and iteratively repeats the following steps until k clusters are found:

1. Loop through all combinations of clusters:
   - Form new clusters by merging two neighbouring clusters into one
   - Calculate squared sum of distances (Equation (5)) for each combination
2. Keep clusters with minimal squared sum of distances

In this way we use agglomerative clustering with a similar approach to the k-means algorithm, the same optimization criterion with the same input parameter and Euclidean distance measure. We therefore expect similar results. However, this agglomerative clustering can easily be adapted to alternative distance measures and could therefore potentially deal with time series of different lengths or containing gaps.

3.3.3 DBSCAN Algorithm

Density-Based Spatial Clustering of Applications with Noise, DBSCAN, is a classical example of clustering based on the maximal allowed distance to neighbouring points that automatically derives the numbers of clusters from the data. It was introduced in 1996 by Ester et al. (1996) and recently revisited by Schubert et al. (2017). The algorithm is based on dividing all points into *core points* or *non-core points* that are close to core points but not themselves surrounded by enough points to be counted as core points. The algorithm needs the maximum allowed distance between points within a cluster (ε) and the minimum number of points per cluster (N_{min}) as input parameters. These two parameters define a core point: If a point has a neighbourhood of N_{min} points at ε distance, it is considered a core point. The algorithm consists of the following steps (Schubert et al. (2017)):

1. Determine neighbourhood of each point and identify core points
2. Form clusters out of all neighbouring core points
3. Loop through all non-core points and add to cluster of neighbouring core point if within maximal distance, otherwise classify as noise

In this way clusters are formed that truly represent a dense collection of 'similar' points. Since we choose to use correlation as distance metric, each cluster will contain correlated time series in our case. All points that can not be assigned to a close surrounding of a core point, are classified as noise or outliers.

3.4 Evaluation Criteria

To determine if an algorithm is suitable, we expect that it fulfils the previously defined criteria:

- A majority of the observation area is separated into distinct regions,
- each cluster shows a change pattern that can be associated with a geomorphic deformation process, and
- time series contained in each cluster roughly follow the mean change pattern.

In order to establish these criteria, we compare the three clustering algorithms, as well as ~~the two different ways to derive time series~~two choices for the number of clusters k, using the following evaluation methods.

3.4.1 Visual Evaluation

The clustered data ~~in Cartesian coordinates~~ are visualized in a top view of the observation area, where each point represents the location of a grid cell~~and its colour the corresponding cluster, which contains the time series in that location. The centre of each grid cell in spherical coordinates is mapped back to the mean Cartesian xy-coordinates, to visualize the clusters of the range time series in a comparable way~~. Each cluster is associated with its cluster centroid, the mean ~~time series in elevation~~ elevation time series of all time series in the respective cluster. ~~In this way, cluster centroids are visualized as elevation time series independent of the coordinate system that was used to generate them. This allows for a direct comparison of~~ For visualization purposes we have added the median elevation back to the cluster centroids~~between both time series extraction methods~~, even though it is not taken into account during the clustering. We subsequently derive change processes visually from the entire clustered area. We establish which kind of deformation patterns can be distinguished and ~~if they match, with what we expect in the respective areas (for example gradual erosion in the intertidal area)~~estimate rates of change in elevation and link them to the underlying process.

3.4.2 Quantitative Evaluation

We use the following criteria to compare the respective clustering and grid generation methods quantitatively:

- percentage of entire area clustered
- minimum and maximum within cluster variation
- percentage of correctly identified change in test area with bulldozer work

The percentage of the area that is clustered differs depending on the algorithm. Especially DBSCAN sorts out points that are too far away (i.e. too dissimilar) from others as noise. This will be measured over the entire observation area. The number of all complete time series counts as 100%.

Each cluster has a mean centroid time series and all other time series deviate from that to a certain degree. We calculate the average standard deviation over the entire month per cluster and report on the minimum and maximum value out of all realized clusters.

3.4.3 Test Area

To allow for a comparison of the clusters with a sort of ground truth, we selected a test area at the bottom of the footpath. In this area a pile of sand was accumulated by a bulldozer, after the entrance to the path was covered with lots of sand during ~~the storm, as found~~ a period of rough weather conditions (8 to 16 January, corresponding to day 7 to 14 in our time series), as reported by Anders et al. (2019). We chose two time stamps for illustration, and show the elevation before the bulldozer activity ~~on 12~~ at the end of the stormy period on 16 January, after the bulldozer activity on ~~16~~ 18 January and the difference between the elevations on these two days in Figure~~6~~ 6 (first row, A,B,C). The area does not change significantly after this event. Within

this test area we classify (manually) each point as 'stable' or 'with significant change' depending on a change in elevation of more than 5 cm (positive or negative). Then we evaluate for each clustering method if the points that are classified as 'with significant change' are in a separate cluster ~~from~~ than the 'stable' points.

The stable cluster consists of cluster 0, the largest cluster when using $k = 6$ for k-means and agglomerative clustering and cluster 0 and 1 combined in the case of $k = 10$ clusters. For evaluating the results of the DBSCAN algorithm we consider all locations that are not clustered (noise) and points in cluster 1 as the 'stable' areas, because the average erosion in cluster 1 is less than 0.15 cm per day. We do not distinguish if there are different clusters within the category of 'with significant change'. However, in Figure 6, the different clusters can be distinguished by their colours, corresponding to the colours of the clusters shown in subsequent figures (Figures~~7, 8 and~~ 7, 8 and 9). We then compare the percentage of correctly classified grid points for the test area, for each of the grid generation and clustering methods.

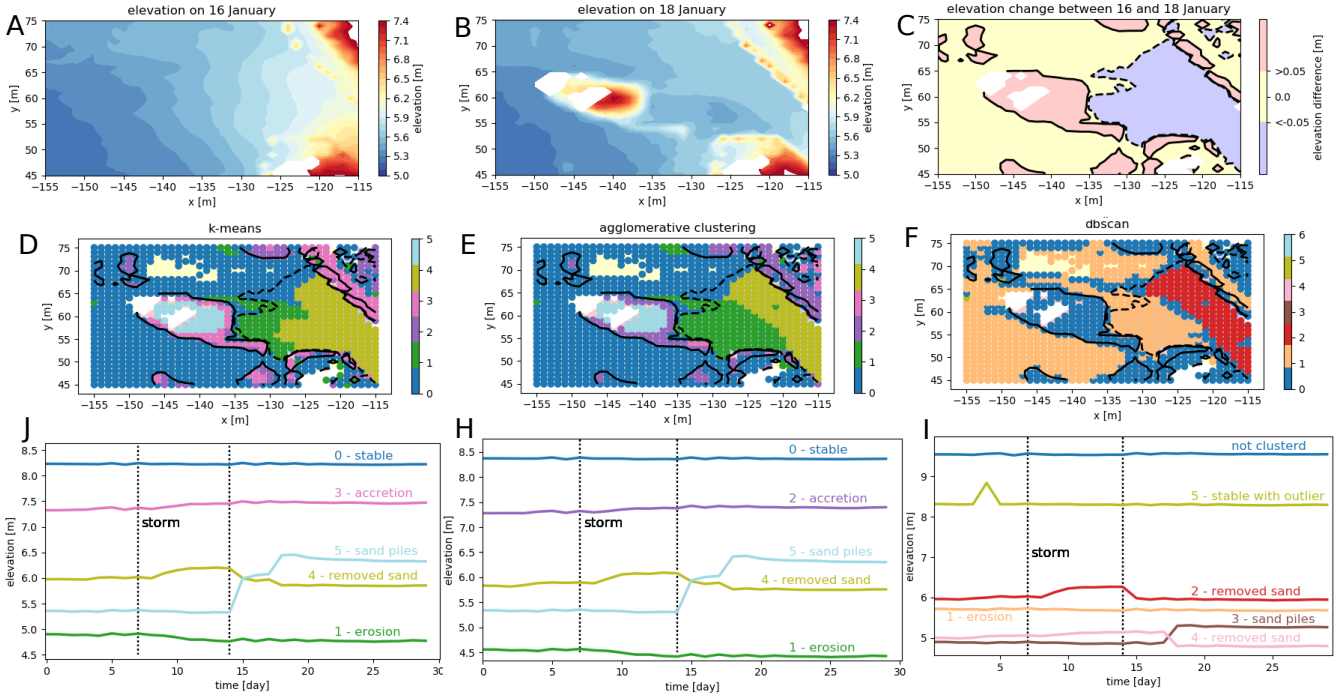

Figure 6. Test area for the comparison of clusters generated with three different algorithms. The test area is located where the northern access path meets the beach (~~as shown in~~ see Figure 1). 1st row: The elevation in the test area is shown on the day before the bulldozer accumulated a sand pile, when the entrance of the path was covered in sand (A) and after the bulldozer did its job (B). Behind the sand pile appears a gap in the data (in white), as the sand pile is obstructing the view for the laser scanner. To the right we show the difference in elevation between ~~the two days~~ January 16 and 18 from a significant level upwards (red) and downwards (blue) (C). 2nd row: Test area with significant changes in elevation (contour lines) and ~~clustered~~ points clustered using the k-means algorithm (D), agglomerative clustering (E) and the DBSCAN algorithm (F). The colours of the clustered dots represent the clusters as shown in Figures 7, 8 and 9, respectively. ~~Points in~~ 3rd row: The corresponding mean time series for each of the ~~'stable' cluster~~ relevant clusters are displayed below each of the plots (~~cluster 0 for k-means and agglomerative clustering~~G,H,I). The dotted lines mark the beginning and ~~outliers (DBSCAN) are not shown~~end of a stormy period.

4 Results

The results are presented in two parts. First, we compare ~~the different time series extraction methods~~two different choices of the parameter k for the k-means algorithm and for agglomerative clustering. Then, we ~~further analyse the clustering algorithms on the time series in elevation. We compare the~~ compare all three clustering methods and evaluate results on the test area, where a bulldozer created a pile of sand (as indicated in Figure 1) and in terms of percentage of data clustered, ~~average standard deviation~~ standard error within each cluster and physical interpretation of clusters.

~~Clustered observation area using the k-means algorithm for time series in elevation (A) and time series in range (B). Different areas are covered and the resulting clusters of range time series appear less suited for identification of change patterns, because the size of the grid cells is changing depending on the distance to the laser scanner, and deformations in elevation appear less pronounced.~~

4.1 ~~Elevation vs. Range Time Series~~Clustering

~~With the method to extract time series based on Cartesian coordinates we extract around 40000 grid cells that contain complete time series in elevation for the entire month. Using the method to extract time series based on spherical coordinates we obtain around 47000 complete time series in range. In Figure ?? it can clearly be seen that the area covered by both time series extraction methods is different . The data in spherical coordinates covers a slightly longer part of the beach and more points at larger distance to the scanner. However, the clusters do not yield a clear partitioning of the beach according to change patterns. Since the spherical coordinate representation is more native to the scanner and the grid cells generally grow in size with increasing distance, changes per grid cell over time are less pronounced. Most deformations on the beach are dominated by variations in elevation, which is not captured well in spherical coordinates~~For the k-means algorithm and agglomerative clustering, we consider two different values $k = 6$ and $k = 10$, exemplary for a smaller number of clusters and a higher number of clusters.

4.2 ~~Clustering methods on Elevation Time Series~~

4.1.1 K-means

~~For~~ With the k-means algorithm, ~~we choose to use $k = 6$ clusters . From our visual inspection this leads~~ the entire observation area is partitioned. The resulting partition depends on the random initialization. The standard error within each cluster is relatively high, compared to the stable area (see Table 2) and generally increases with the size of the cluster. Even the cluster with the smallest standard error (averaged standard deviation per time series over the clustered area), still shows a standard error of 0.77 m (cluster 5 for $k = 6$). We show the resulting clusters obtained using the k-means algorithm with number of clusters $k = 6$ and $k = 10$. Visual inspection shows that both values lead to good, usable results by partitioning the set of time series into clusters that are small enough to capture geomorphic changes but not too large, which would make them less informative. ~~With the k-means algorithm,~~ As displayed in Figure 7, a large part of the beach is contained in a 'stable'

Table 3. Summary of comparison of k-means algorithm, agglomerative clustering (AGG) and DBSCAN algorithm.

	k-means		**AGG**		**DBSCAN**
entire observation area					
number of clusters	6	10	6	10	6
min no. points/cluster	108	34	108	39	45
area clustered	100%	100%	100%	100%	44%
max std ~~within~~ error/cluster	~~3.23 m~~3.22 m	~~3.19~~ 3.1 m	3.18 m	2.86m	4.0 m
min std ~~within~~ error/cluster	~~0.7 m~~0.77 m	~~0.72~~ 0.68 m	~~0.3~~ 0.79 m	0.71 m	0.33 m
test area: correctly identified					
stable points	81%	82%	86%	~~41~~86%	99%
positive changes	97%	97%	86 %	~~1.5~~86%	0%
negative changes	93%	93%	98 %	~~87~~98%	54%
total	85%	86%	88 %	~~45~~88%	79%

cluster when using $k = 6$ (cluster 0, blue). This cluster, as well as some of the others, are split up into several smaller clusters when using $k = 10$. For example, the ~~entire observation area is divided into partitions, which change slightly depending on the random initialization.The standard deviation within each cluster is relatively high , and generally increases with the size of the cluster. Even the cluster with the smallest standard deviation~~ intertidal zone (i.e. the area that is under water during high tide and exposed during low tide) is eroding mostly during stormy days in the first half of the month. This zone is contained entirely in cluster 1 (green) when using $k = 6$. In the case of $k = 10$, this part is split up into three clusters, one with a similar mean time series (cluster 2, green), one eroding with a pattern similar to cluster 2, but mostly representing sand banks (cluster 3, brown) and one gradually eroding at a low rate over the entire month ~~, shows a standard deviation of 0.7 m (~~(cluster 1, orange). It also becomes clear, that the sand piles that were generated by bulldozer works at different locations ($k = 6$ cluster 5~~).~~, light blue) were created on different days ($k = 10$, clusters 8 and 9, yellow and light blue). Some features, like the cleared part of the paths, the sand piles and the intertidal zone can be distinguished in both cases.

On the test area the k-means algorithm correctly classifies about 85% of points into 'stable', 'significant negative change', or 'significant positive change' in the case of $k = 6$. However, as can be seen in Figure 6, a part of the points with negative change are not identified. These clusters are split up further in the case of $k = 10$, which does not influence the results in the test area a lot. A summary of these results is provided in Table 3.

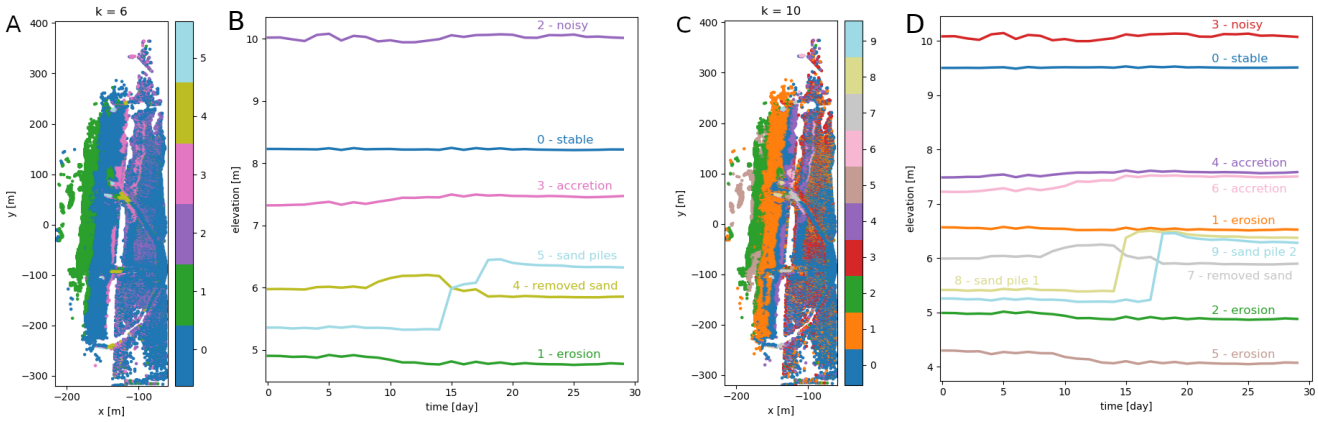

Figure 7. A/C: Overview of the entire observation area divided into ~~six~~ clusters using k-means ~~depending on elevation changes~~with $k = 6$ (A) and $k = 10$ (C). B/D: Corresponding cluster centroids for each of the clusters shown in A and C, respectively. ~~The bulldozer activity can be seen between 12~~ By using a larger number of clusters k, more processes become visible, for example two sand piles (A/B: cluster 5) created on two different days (C/D: cluster 8 and ~~16 January in~~ 9). Also the large stable areas (A/B: cluster ~~5.~~ 0) and slowly accreting areas (A/B: cluster 3) are split up into several clusters: A slightly eroding area (C/D: cluster 3) is split up from the stable part and the accreting area is split into two (C/D: cluster 4 and cluster 6).

4.1.2 Agglomerative Clustering

The agglomerative clustering algorithm is set up, as the k-means algorithm, to find six and ten clusters. It produces results very similar to the clusters found with the k-means algorithm, as can be seen comparing Figures 7 and 8 and Figures 6 D and E. Clusters 2 and 3 from agglomerative clustering correspond roughly to the clusters 3 and 2 from k-means clustering. The ordering of clusters is according to size, so more time series are considered 'noisy' according to k-means, whereas agglomerative clustering assigns more of these time series to the gradually accreting cluster. All other clusters appear to be nearly identical and show similar spatial distributions as well as centroid shapes. The differences between the two choices of the number of clusters k are also very similar.

On the test area, the detection of negative and positive changes is more balanced and leads to an overall score of 88 % correctly identified points. Agglomerative clustering clearly separates the path that was cleared by the bulldozer and identifies it as eroding.

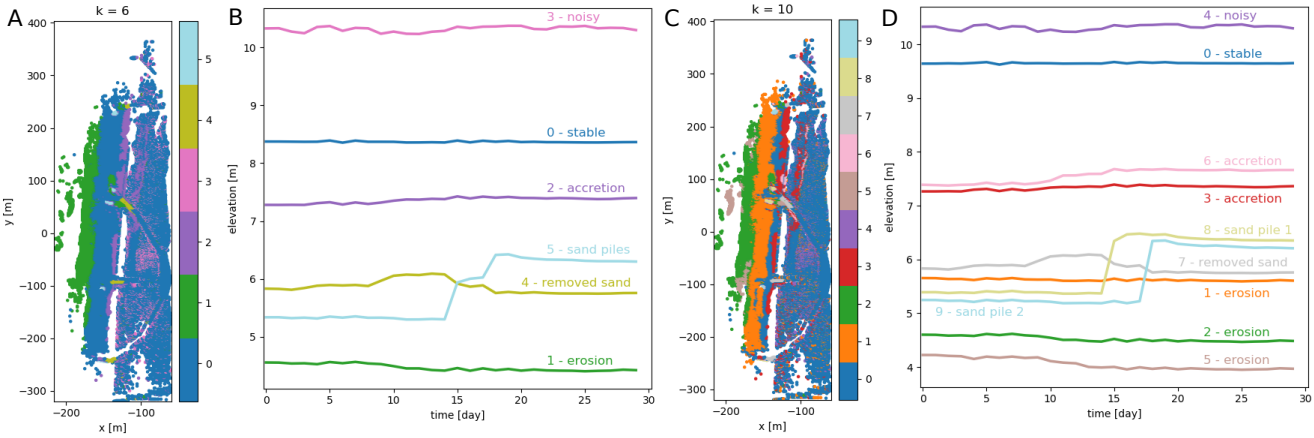

Figure 8. A/C: Overview of the entire observation area divided into ~~six~~ clusters ~~depending on elevation changes~~ using agglomerative clustering with $k = 6$ (A) and $k = 10$ (B). B/D: ~~Corresponding~~ corresponding cluster centroids for each of the clusters shown in A and C respectively. The clusters are ~~very~~ similar to the ones found with k-means.

4.1.3 DBSCAN

When we use the DBSCAN algorithm on the same data set, with minimum number of points $N_{min} = 30$ and maximum distance $\varepsilon = 0.05$, a large part of the time series (55 %) is classified as noise, meaning that they are not very similar (i.e. not correlated, since we use correlation as distance measure) to any of the other time series. However they roughly match the combined areas that are identified as stable and noisy by k-means (clusters 0 and 2 for $k = 6$). The remaining time series are clustered into six clusters. The standard ~~deviation~~ error within each cluster is generally lower than in the clusters generated with k-means (minimum standard ~~deviations is 0.4 m~~error is 0.33 m) without considering the time series that are classified as noise.

The intertidal zone cannot be separated clearly from the 'noise' part of the observation area, nor can we distinguish the stable path area or the upper part of the beach. In the test area, the sand pile is not represented by a separate cluster and positive changes in elevation are not found, which results in an overall worse percentage of correctly identified points. But, two clusters represent areas, which are relatively stable throughout the month, except for a sudden peak in elevation on one day. These peaks are dominated by a van parking on the path on top of the dunes and people passing by~~and~~ , and are not caused by actual deformation~~in the observed area, as shown in~~ , compare Figure 9.

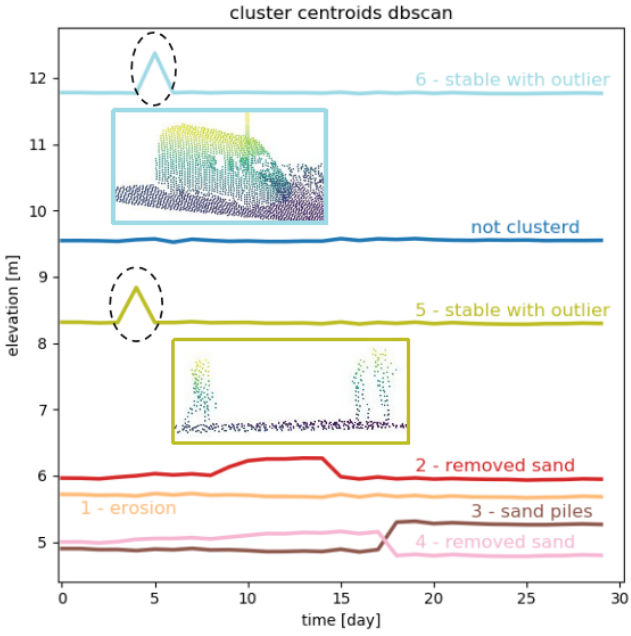

Figure 9. Mean time series per cluster found with the DBSCAN algorithm. Outliers or not clustered points are represented by the blue mean time series. The two most prominent time series (cluster 5 and 6, light green and light blue) are located on the path on top of the dunes. The peaks are caused by a group of people ~~passing by~~ and a van, on the 5th and 6th of January respectively, ~~as shown in~~ illustrated by the point clouds in the middle of the plot.

On the test area the DBSCAN algorithm performs worse than both other algorithms. In total ~~only 45~~79% of points are correctly classified into 'stable' ~~, 'significant negative change',~~ or 'significant ~~positive~~ negative change'. As stable points we count in this case all points that are classified ~~as noise, because only~~ either as noise or belong to cluster 1 (orange). The reason for this is that the mean of all time series that ~~show coherent change patterns are clustered by the DBSCAN algorithm~~are not clustered appears relatively stable, while cluster 1 describes very slow erosion of less than 0.15 cm per day. This matches with ~~about 41~~99% of points classified as stable in the ground truth data. ~~However only 1.5% of the points with positive significant changes are correctly identified, they~~ But, no single cluster is formed containing only the points where sand is accumulating, even though these clusters are distinguished by the other two algorithms. These points are mixed up with ~~a large class of only~~

~~slightly varying points .~~ the large cluster of slightly eroding points in cluster 1. We can see in Figure 6, that ~~a significant part of the stable area is also included in the same cluster~~the only significant process found in the test area is the cleared path (cluster 2, red).

4.2 Identification of Change Processes

390 ~~From the clustering using range time series, no clear change processes can be distinguished and the beach cannot be partitioned according to deformation patterns.~~

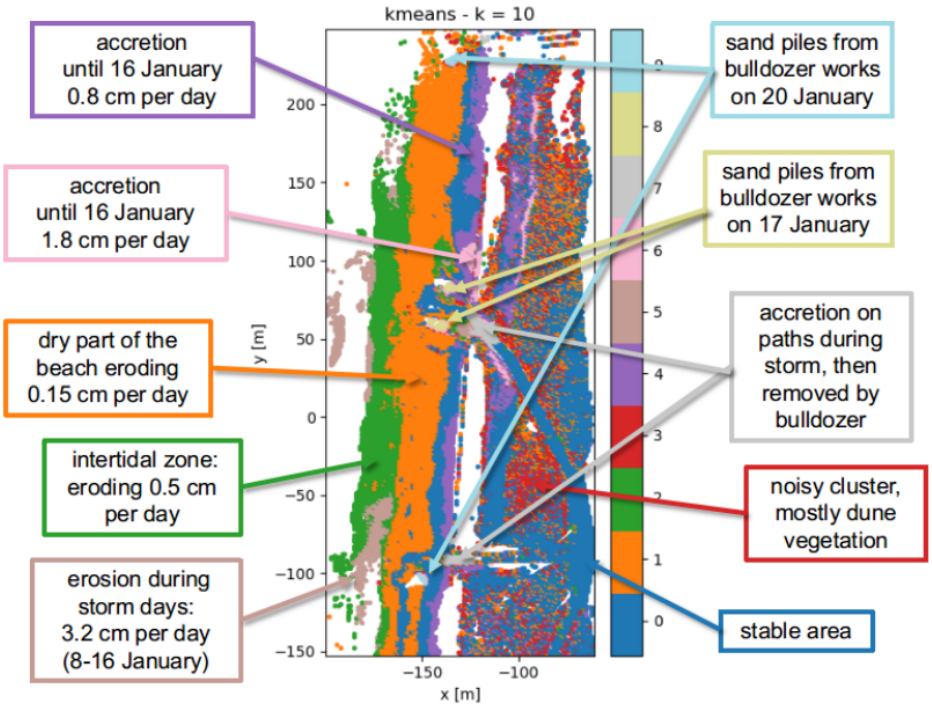

Figure 10. Observation area partitioned into clusters ~~with~~ by the k-means algorithm ~~used on elevation time series~~with $k = 10$. The ~~blue area represents the part where time series are mostly stable and show very few change. For the remaining clusters the main cause of the change process is indicated in the figure. Areas with erosion or accretion~~ associated processes are ~~marked~~ annotated with ~~a '-' or '+' symbol. The '++' symbol indicates a steep accretion, and '+/-' indicates an area where~~ the ~~accretion was followed by erosion (or removal) of sand~~corresponding colours.

Considering the clusters found by the k-means algorithm and agglomerative clustering~~on elevation time series~~, we can clearly distinguish between time series that represent erosion and ~~gradual accretion~~ accretion with different magnitudes and at different times of the month, as well as a sudden jump in elevation, caused by bulldozer work. In Figure 10 we show the clusters and associated main ~~cause for deformations. The cluster~~ process. To give an idea of the magnitude of the most prominent change patterns, we fit straight lines through the mean time series or parts of it (where the slope is steepest) and derived average rates of change in elevation from the estimated slopes. The clusters dominated by erosion~~is~~ , close to the water line ~~and roughly~~ (clusters 2 and 5) represent the inter-tidal zone of the beach~~and is~~ . The elevation changes in this area are likely caused by the effects of tides and waves. The ~~slowly accreting area is~~ change rates were partly accelerated during the stormy period in the first half of the month. Accreting areas are mostly at the upper beach, close to the dune foot and on the paths in the dunes (clusters 4, 6 and 7). These areas as well as a large cluster on the upper beach (cluster 1, orange), which undergoes a slight and gradual erosion over the entire month, are likely dominated by aeolian sand transport. The most obvious change process, is the sand removed from the entrances of the paths leading to the beach by bulldozer works (cluster~~4~~ 7) and accumulated in ~~a pile of sand (cluster 5). The noisy cluster is~~ piles of sand at four different locations on two days (clusters 8 and 9). Points contained in the noisy cluster (cluster 3) are spread out through the dune area and noise is probably caused by moving vegetation.

5 Discussion

We successfully applied the presented methods on a data set from permanent laser scanning and demonstrated the identification of deformation processes from the resulting clusters. Here we discuss our results on ~~time series extraction,~~ distance measures, clustering methods and the choice of their respective input parameters and derivation of change processes.

5.1 ~~Time Series Extraction~~

~~We compared two different methods to extract time series from the PLS data set either in elevation or in range. The time series extraction in range, which is a more native way of using the data, is very sensitive to vertical structures in the data set, or points in the air in-between the observed surface and the scanner. After removing those points and using the median range per grid cell in spherical coordinates, the time series appear to be dominated by noisy fluctuations, which do not vary a lot depending on location. Clear change patterns can therefore not be distinguished with any of our algorithms and the distinction of areas that follow a certain change pattern is not possible. A likely cause of this issue, is that most changes are observed in elevation (z-direction) and not in the direction of the range, which makes them less pronounced in spherical coordinates. An alternative approach could be the use of spherical coordinates for the generation of the grid cell, but extraction of time series in elevation instead of range.~~

~~In contrast, the time series extraction in Cartesian coordinates provides promising results. Some data is lost, due to low point density in grid cells that are at large distances of the laser scanner. Besides that, the resulting clusters clearly follow recognizable deformation patterns and the clustering allows to separate regions according to these patterns.~~

5.1 Distance Measures

Possible distance measures for the use in time series clustering are analysed among others by Iglesias and Kastner (2013) and Liao (2005). We use Euclidean distance in combination with the k-means algorithm and agglomerative clustering for our analysis. It has been shown by Keogh and Kasetty (2003) that especially for time series with high dimensions, alternative distance measures rarely outperform Euclidean distance. However, we have to note here, that Euclidean distance is affected by the so called 'curse of dimensionality', which causes a space of ~~time series with many epochs (~~long time series (with many dimensions) to be difficult to cluster. ~~For more details on this issue see Assent (2012) and Verleysen and François (2005)~~ The problem with clustering time series in high dimensional spaces with k-means, is that Euclidean distance is based on the sum of all pointwise differences. This leads to a space, where the variance of the distances decreases with increasing time series length. Therefore it will be harder to categorize time series as similar, and fewer meaningful clusters will emerge, the more observations we use. This could possibly lead to difficulties, when extending these methods to the use of longer time series, but does not appear to degrade results on our current data set. For more details on this issue see Assent (2012), Verleysen and François (2005) and Zimek et al. (2012).

We chose for the use of correlation distance with the DBSCAN algorithm, because correlation in principle represents a more intuitive way of comparing time series (see Figure 4). DBSCAN is based on ~~identifying the~~ identification of clusters of high density, which in our case works better using correlation distance instead of Euclidean distance. Using Euclidean distance, there are very few clusters of 'very similar' time series and an even larger part of the beach is classified as noise. Only in combination with correlation distance, we could derive a set of input parameters for the DBSCAN algorithm to produce relevant results.

Scaling the time series with their respective standard deviations for the use of Euclidean distance would make these two distance measures equivalent. However, this did not improve our results using k-means or agglomerative clustering. Subtle differences within the stable cluster would become prominent in that case, but the larger differences between clusters as we find them without the scaling, would be reduced.

Neither of the the two distance measures analysed here can deal with gaps in the time series~~. They also~~ , which would be of great interest to further analyse especially the intertidal area and sand banks. Additionally, both distance measures do not allow to identify identical elevation patterns that are shifted in time as similar. An alternative distance measure suitable to deal with these issues would be Dynamic Time Warping (DTW), which accounts for similarity in patterns even though they might be shifted in time (Keogh and Ratanamahatana (2005)). An interpolation method to fill gaps in elevation over short time spans based on surrounding data or a feature based clustering method could be other alternatives.

5.2 Clustering Methods

The use of k-means clustering on elevation time series from the same data set was demonstrated by Lindenbergh et al. (2019) and has shown promising first result. We follow the same approach and, as a comparison, use agglomerative clustering, with the same optimization criterion, distance metric and input parameter. As expected the results are similar, although agglomerative

clustering does not depend on random initialization. It therefore delivers the same result for every run, which is an advantage. Considering our previously defined criteria:

- a majority of the observation area is separated into distinct regions,
- each cluster shows a change pattern that can be associated with a geomorphic deformation process, and
- time series contained in each cluster roughly follow the mean change pattern,

both algorithms are suitable and the differences in the resulting clusters are negligible for our specific data set.

However, the computational effort needed to loop through all possible combinations of merging clusters for agglomerative clustering is considerably higher. Of the three algorithms that were used in this study, agglomerative clustering is the only one that regularly ran into memory errors. This is a disadvantage considering the possible extension of our method to a data set with longer time series.

One of the disadvantages of the k-means algorithm and our configuration of agglomerative clustering, is that the number of clusters has to be defined in advance. Our ~~choice~~ choices of $k = 6$ ~~clusters yields~~ and $k = 10$ clusters both yield promising results, but ~~remains~~ remain somewhat arbitrary, especially without prior knowledge of the data set. A lower number of clusters k (for example $k = 6$) yields a division of the beach into sections (inter-tidal zone, dry part of the beach) and highlights the most prominently occurring changes (bulldozer works). When using a larger number of clusters k, several of the previously mentioned clusters are split up again and more detailed processes become visible. The erosion and accretion patterns on the beach appear at different degrees in distinct regions, which is valuable information. Also the sand piles, which appeared in one cluster for $k = 6$ are now split up according to the different days, on which they were generated. We consider this possibility to identify and specify anthropogenic induced change an illustrative example of the influence of the choice of the number of clusters k. We have considered two ~~different~~ data independent methods to determine a suitable value for k: analysis of the overall sum of variances for different values of k and so-called *cluster balance* following the approach of Jung et al. (2003). Neither of them resolved the problem satisfactorily and we cannot make a generalized recommendation, independent of the application, for the choice of k at this point.

To avoid this issue we also compare both approaches with the use of the DBSCAN algorithm. It is especially suitable to distinguish anomalies and unexpected patterns in data as demonstrated by Çelik et al. (2011) using temperature time series. To decide, which values are most suitable for the two input parameters of the DBSCAN algorithms we plot the percentage of clustered points and the number of clusters depending on both parameters (see Figure 11). However, this did not lead to a clear indication of an 'optimal' set of parameters. After the trade-off analysis between the number of points in clusters and the number of clusters (not too high, so that the clusters become very small and not too low so that we generate only one big cluster) we chose $\varepsilon = 0.05$ and $N_{min} = 30$ by visually inspecting the resulting clusters.

An alternative clustering approach for time series based on fuzzy C-means is proposed by Coppi et al. (2010). They develop a method to balance the clustering based on the pattern of time series while keeping an approximate spatial homogeneity of the clusters. This approach was successfully applied to time series from socio-economic indicators and could be adapted for

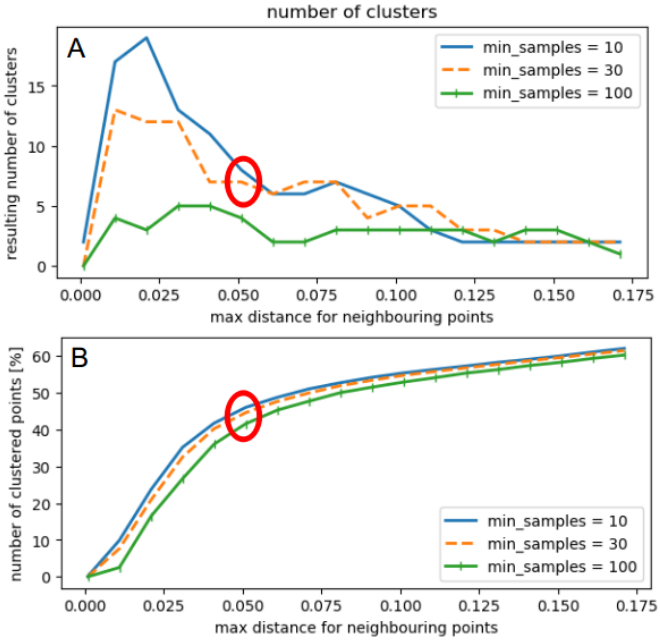

Figure 11. DBSCAN selection of input parameters: Number of clusters versus input parameter maximum distance within clusters and minimum number of points and percentage of total points in clusters (not classified as noise/outliers). The choice of an 'optimal' set of parameters is not obvious. We indicate our selection with a red circle in both plots.

our purpose. It could potentially improve detection of features like sand bars, or bulldozer work, but not distinguish moving vegetation in the dunes as our current approach does.

A similar approach would be to use our clustering results and identified change patterns as input to the region-growing approach of Anders et al. (2020). In this way we could combine advantages of both ~~approaches~~ methods by making the identification of the corresponding regions for each distinct deformation pattern more exact, without having to define possible deformation patterns in advance.

5.3 Derivation of Change Processes

As shown in Figure 10, we identified change processes from the clusters generated by k-means~~and agglomerative clustering~~. Agglomerative clustering shows similar clusters and therefore yields similar results. Each centroid representing the mean time series of ~~its cluster~~ the k-means clusters shows a distinct change pattern (see Figures~~7 and~~ 7 and 8), which allows to conclude on a predominant deformation ~~. By associating~~ process. By fitting a straight line through the mean time series, or part of it, we estimated the slope corresponding to the average rate of change in elevation. Associating the centroids with the location and spatial spread of the clusters, ~~we can~~ allows to derive the main cause for ~~this deformation~~the respective deformations. In some cases extra information, or an external source of validation data would be useful to verify the origin of the process. This will

be taken into account for future studies. ~~However,~~ The location of the clusters and for example the steep rise of the ~~centroid from cluster 5 allows to conclude~~ mean time series representing the sand piles allows for the conclusion that the cause of this sudden accretion is ~~not natural~~anthropogenic. The information found by Anders et al. (2019) for the research on their study, confirms the coinciding bulldozer works. The derived average rates of change in elevation allow for the possibility to derive mass budgets to quantify volume changes over specific amounts of time from our data, showing a possible application of our method, that is of large scientific interest (see for example de Schipper et al. (2016)).

The DBSCAN algorithm successfully identifies parts of the beach that are dominated by a prominent peak in the time series (caused by a van and a small group of people). Out of the three algorithms that we compare, it is most sensitive to these outliers in the form of people or temporary objects in the data. It was not our goal for this study, to detect people or objects on the beach, but this ability could be a useful application of the DBSCAN algorithm to filter the data for outliers in a pre-processing step.

6 Conclusions

We compared three different clustering algorithms (k-means, agglomerative clustering and DBSCAN) on a subset of a large time series data set from permanent laser scanning on a sandy urban beach. We successfully separated the observed beach and dune area according to their deformation patterns. Each cluster, described by the mean time series, is associated with a specific process (such as bulldozer work, tidal erosion) or surface property (for example moving vegetation cover).

The most promising results are found using k-means and agglomerative clustering, which both correctly classify between 85 and 88 % of time series in our test area. However, they both need the input of the number of clusters we are looking for and agglomerative clustering is computationally expensive. DBSCAN turned out to be more suitable for the identification of outliers or unnatural occurring changes in elevation due to temporary objects or people in the observed area.

Our key findings are summarized as follows:

1. Both k-means and agglomerative clustering fulfil our criteria for a suitable method to cluster time series from permanent laser scanning.

2. Predominant deformation patterns of sandy beaches are detected automatically and without prior knowledge using these methods.

3. Change processes on sandy beaches, which are associated with a specific region and time span, are detected in a spatio-temporal data set from permanent laser scanning with the presented methods.

Our results demonstrate a successful method to mine a spatio-temporal data set from permanent laser scanning for predominant change patterns. The method is suitable for the application in an automated processing chain to derive deformation patterns and regions of interest from a large spatio-temporal data set. It allows such a data set to be partitioned in space and time according to specific research questions into phenomena, such as for example the interaction of human activities and natural sand transport during storms, recovery periods after a storm event or the formation of sand banks. The presented methods enable the use of an extensive time series data set from permanent laser scanning to support the research on long-term and small scale processes on sandy beaches and improve analysis and modelling of these processes. In this way we expect to contribute to an improved understanding and managing of these vulnerable coastal areas.

Data availability. The data set used for this study is available via 4TU Centre for Research Data: https://doi.org/10.4121/uuid:409d3634-0f52-49ea-8047-aeb0fefe78af (Vos et al. (2020)).

Author contributions. M. Kuschnerus has carried out the investigation, developed the methodology and software and has realized all visualizations and written the original draft. R.C. Lindenbergh supervised the work and contributed to the conceptualization and to the writing by reviewing and editing. S. Vos developed the instrumental set-up of the laser scanner and collected the raw data set that was used for this research.

Competing interests. The authors declare that they have no conflict of interest.

Acknowledgements. This work is part of the Open Technology Programme with project number 16352, which is (partly) funded by the Dutch Research Council (NWO). The authors would also like to thank M.C. Mulder for the work on his Bachelor thesis with the title 'Identifying Deformation Regimes at Kijkduin Beach Using DBSCAN'.

References

Aarninkhof, S., De Schipper, M., Luijendijk, A., Ruessink, G., Bierkens, M., Wijnberg, K., Roelvink, D., Limpens, J., Baptist, M., Riksen, M., Bouma, T., de Vries, S., Reniers, A., Hulscher, S., Wijdeveld, A., van Dongeren, A., van Gelder-Maas, C., Lodder, Q., and van der Spek, A.: ICON.NL: coastline observatory to examine coastal dynamics in response to natural forcing and human interventions, international Conference on Coastal Sediments ; Conference date: 27-05-2019 Through 31-05-2019, 2019.

Anders, K., Lindenbergh, R. C., Vos, S. E., Mara, H., de Vries, S., and Höfle, B.: High-frequency 3D geomorphic observation using hourly terrestrial laser scanning data of a sandy beach, ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences, IV-2-W5, 317–324, 2019.

Anders, K., Winiwarter, L., Lindenbergh, R., Williams, J. G., Vos, S. E., and Höfle, B.: 4D objects-by-change: Spatiotemporal segmentation of geomorphic surface change from LiDAR time series, ISPRS Journal of Photogrammetry and Remote Sensing, 159, 352–363, 2020.

Assent, I.: Clustering high dimensional data, WIREs Data Mining and Knowledge Discovery, 2, 340–350, 2012.

Belgiu, M. and Csillik, O.: Sentinel-2 cropland mapping using pixel-based and object-based time-weighted dynamic time warping analysis, Remote Sensing of Environment, 204, 509–523, 2018.

Çelik, M., Dadaşer-Çelik, F., and Ş. Dokuz, A.: Anomaly detection in temperature data using DBSCAN algorithm, in: 2011 International Symposium on Innovations in Intelligent Systems and Applications, pp. 91–95, 2011.

Coppi, R., D'Urso, P., and Giordani, P.: A fuzzy Clustering model for multivariate spatial time series, Journal of Classification, 27, 54–88, 2010.

de Schipper, M. A., de Vries, S., Ruessink, G., de Zeeuw, R. C., Rutten, J., van Gelder-Maas, C., and Stive, M. J.: Initial spreading of a mega feeder nourishment: Observations of the Sand Engine pilot project, Coastal Engineering, 111, 23 – 38, 2016.

Deza, M. and Deza, E.: Encyclopedia of distances, Springer Verlag, Dordrecht : New York, 2009.

Ertöz, L., Steinbach, M., and Kumar, V.: Finding clusters of different sizes, shapes, and densities in noisy, high dimensional data, in: Proceedings of the 2003 SIAM International Conference on Data Mining, vol. 0 of *Proceedings*, pp. 47–58, Society for Industrial and Applied Mathematics, 2003.

Ester, M., Kriegel, H.-P., Sander, J., and Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise, Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, p. 6, 1996.

Harmening, C. and Neuner, H.: A spatio-temporal deformation model for laser scanning point clouds, Journal of Geodesy, 94, 26, 2020.

Iglesias, F. and Kastner, W.: Analysis of similarity measures in times series clustering for the discovery of building energy patterns, Energies, 6, 579–597, 2013.

Jain, A. K.: Data clustering: 50 years beyond K-means, Pattern Recognition Letters, 31, 651–666, 2010.

Jung, Y., Park, H., Du, D.-Z., and Drake, B. L.: A decision criterion for the optimal number of clusters in hierarchical clustering, Journal of Global Optimization, 25, 91–111, 2003.

Keogh, E. and Kasetty, S.: On the need for time series data mining benchmarks: A survey and empirical demonstration, Data Mining and Knowledge Discovery, 7, 349–371, 2003.

Keogh, E. and Ratanamahatana, C. A.: Exact indexing of dynamic time warping, Knowledge and Information Systems, 7, 358–386, 2005.

Lazarus, E. D. and Goldstein, E. B.: Is there a bulldozer in your model?, Journal of Geophysical Research: Earth Surface, 124, 696–699, 2019.

Liao, T. W.: Clustering of time series data - a survey, Pattern Recognition, 38, 1857–1874, 2005.

Lindenbergh, R., van der Kleij, S., Kuschnerus, M., Vos, S., and de Vries, S.: Clustering time series of repeated scan data of sandy beaches, ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, XLII-2/W13, 1039–1046, 2019.

Masselink, G. and Lazarus, E. D.: Defining coastal resilience, Water, 11, 2587, 2019.

Neill, D. B.: Expectation-based scan statistics for monitoring spatial time series data, International Journal of Forecasting, 25, 498–517, 2009.

O'Dea, A., Brodie, K. L., and Hartzell, P.: Continuous coastal monitoring with an automated terrestrial Lidar scanner, Journal of Marine Science and Engineering, 7, 37, 2019.

Park, H.-S. and Jun, C.-H.: A simple and fast algorithm for K-medoids clustering, Expert Systems with Applications, 36, 3336–3341, 2009.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E.: Scikit-learn: Machine learning in Python, Journal of Machine Learning Research, 12, 2825–2830, 2011.

Recuero, L., Wiese, K., Huesca, M., Cicuéndez, V., Litago, J., Tarquis, A. M., and Palacios-Orueta, A.: Fallowing temporal patterns assessment in rainfed agricultural areas based on NDVI time series autocorrelation values, International Journal of Applied Earth Observation and Geoinformation, 82, 101 890, 2019.

Schubert, E., Sander, J., Ester, M., Kriegel, H.-P., and Xu, X.: DBSCAN revisited, revisited: Why and how you should (still) use DBSCAN, 2017.

Steinbach, M., Klooster, S., Tan, P., Potter, C., Kumar, V., and Torregrosa, A.: Clustering earth science data: Goals, issues and results, Proceedings SIGMOD KDD Workshop on Temporal Data Mining, p. 8, 2001.

Tan, P., Potter, C., Steinbach, M., Klooster, S., Kumar, V., and Torregrosa, A.: Finding spatio-temporal patterns in earth science data, Proceedings SIGMOD KDD Workshop on Temporal Data Mining, p. 12, 2001.

Verleysen, M. and François, D.: The curse of dimensionality in data mining and time series prediction, in: Computational Intelligence and Bioinspired Systems, edited by Cabestany, J., Prieto, A., and Sandoval, F., Lecture Notes in Computer Science, pp. 758–770, Springer, Berlin, Heidelberg, 2005.

Vos, S., Lindenbergh, R., and de Vries, S.: CoastScan: Continuous monitoring of coastal change using terrestrial laser scanning, Proceedings of Coastal Dynamics 2017, 2017.

Vos, S., Kuschnerus, M., de Vries, S., and Lindenbergh, R.: CoastScan: Data of daily scans at low tide Kijkduin January 2017, https://doi.org/10.4121/uuid:409d3634-0f52-49ea-8047-aeb0fefe78af, dataset, 2020.

Zimek, A., Schubert, E., and Kriegel, H.-P.: A survey on unsupervised outlier detection in high-dimensional numerical data, Statistical Analysis and Data Mining: The ASA Data Science Journal, 5, 363–387, 2012.

Answer to reviewer comments on Interactive comment on "Coastal Change Patterns from Time Series Clustering of Permanent Laser Scan Data" by Mieke Kuschnerus et al.

Anonymous Referee #1

Figure 1 could use a pair of axis to help the reader locate him/herself when referring to figures 8,9,10,12.

– Figure 1 will be updated to match the orientation and axis of the following figures for easier comparison. (Figure 1, page 5)

Also, in the text you mention that the origin of the local coordinate system is at the laser scanner location, but in the figure there is an (xo,yo) just next to the wooden stairs. Which is the correct origin.

– The location of the stable time series will be marked with (x_t, y_t) instead of (x_0, y_0). With the added axes it will be easy to understand that the origin of the coordinate system is at the location of the laser scanner. (Figure 1, page 5)

Also note that in the figure caption you mention 2019 as the scan date, but in the text (line 98) it is 2017.

– This is a typo in the caption and will be fixed. (Figure 1, page 5)

Figure 2 should be plotted with the same aspect as the test area shown in fig.1 (abaout 40x55m?)

– Does this refer to Figure 7? Figure 2 is a photo of the laser scanner. I assume this is fixed with the update of Figure 1 and Figure 7. (Figure 6, page 16)

Figures 8,9,10,12 - I found a bit hard at first to relate this with figure 1 (due the lack of axes), so in fig 8 maybe include also the view of fig.1? Perhaps indicate the same fea- tures of fig.1 (stairs, road,etc)?

– see above. (Figures 7, 8, 10)

Also, did you checked if the colors are safe for colorblind people? If not I suggest a small application called ColorOracle (free, multiplatform) that allows you to simulate the three main colorblindness.

– Thank you for the suggestion. We will look into it.

In fig.9, try to move the legend so it won't cover any lines, and consider using no only colors, but also different line widths and symbols (dash, dot-dash, etc) so it will be easier to differentiate the lines visually

– This will be fixed by annotating each curve individually instead of showing a legend, as suggested by Referee #2. (Figure 9, page 21)

Anonymous Referee #2

[. . .] I would also urge the authors to better highlight (if possible) what exactly the advantages are of applying a clustering technique. I actually found the results in the end rather underwhelming, because if you remove the two clusters associated with the bulldozer works the results really just yield four different trends of geomorphic change: stable surfaces, steadily eroding and steadily accreting surfaces, and fluctuating surfaces (noise). These four trends are hardly surprising and can just as easily be detected from simple erosion/deposition mapping.

– The advantages of our method have to be highlighted and explained in more detail. We will address this in the result section 'Identification of Change Processes'. Shortly summarized we would like to emphasize the following points:

- *Erosion/accretion is detected at different rates, without prior specification of any rate of change or threshold/ We will highlight the specific average rates for different clusters.*
- *The intertidal area can clearly be distinguished, as well as the location of an 'edge' along the beach between the dry part of the sand and the wet intertidal area.*
- *Identification of 'noisy' areas in the dunes, which are areas dominated by moving vegetation*
- *Possibility to detect the time and location of anthropogenic changes like the sand pile in the test area (section 4.2, page 22/23)*

The limitation of the application here may be related to the fact that the analysis was directed to 6 clusters. The manuscript does not give objective or quantitative justification for this decision, other than that the results with 6 clusters seemed good or reasonable to the authors (L304), but this then effectively preempts the possibility of finding something new or interesting. Maybe 10 clusters could have revealed some more interesting trends, for example.

– It is hard to give a general rule for the 'best' choice of the number of clusters k. We will address this in more detail and present different options for the choice of k and their advantages and disadvantages. It comes down to a trade-off between very detailed small clusters, which for example allow to derive the different days, on which the sand piles were erected, versus splitting up a generally stable cluster into small clusters, which does not give new insight into any processes. (section 4.1, page 17/18 and section 5.2, page 25)

L39: define what you mean with "epochs" L39: why "high-dimensional"? This is just a 4D dataset. At this point in the paper it is not clear to the reader yet that you are going to define your data in a multi-dimensional space with the dimensions defined by the snapshots (epochs?).

– The number of dimensions and epochs both refer to the number of time steps in each time series of the data set. I will reformulate to make this more clear. (page 2, line 40/41)

L40: citation here is 15 yrs old, can you refer to more recent literature on the challenges of data mining?

– *yes (page 2, line 43)*

L85: why the second representation in terms of range data? The fact that it's the native data format doesn't really give us an actual justification. Your second criterion is about geomorphic deformation, which presumably relates specifically to height changes, hence the cartesian grid seems most suitable to that. Your results later on essentially show that the range format is simply not useful, so you could achieve a great simplification and a more focused message here if you strip out all this stuff about the range format and just report the results related to the cartesian data.

– *This is a valid point. We will remove the part on spherical representation and clustering of range time series in order to give the paper more focus.*

Fig.1 says data from 2019, but text says data was from Dec-2016 to May-2017?

– *This is a typo and will be adapted. (Figure 1, page 5)*

L122 and earlier: not suitable to use x_0 and y_0 for identifying a test location as it has nothing to do with zero. Suggest subscript 't' or even 'test'.

– *We will adapt this according to your suggestion. (Figure 1, page 5 and Figure 3, page 7)*

Table 1 and associated text: we really need more info on these test areas: are they single points? areas? If latter, what size. Then, the test statistics is not informative. Stdev is not sufficient, you should be able to calculate the standard error and the associated 95% confidence intervals around the mean height. Also, why the difference in N vs S? This requires discussion. In the manuscript you present these test results here, but in the results and discussion there should be further reflection on the potential impact of the error on the cluster classifications.

– *This is a good point. The stable reference surface can be represented by all paved paths combined. We will provide more information on the statistical properties to emphasize the stability and the order of magnitude of errors. The effect of errors in the instrument on the clustering results is assumed to be negligible, but with a more detailed specification of the rate of change in elevation in the eroding and accreting areas, this will be more straight forward to show. We will add this to the discussion section. (Table 1, page 8)*

L145: you remove the mean in the cartesian format, but not in the range format, why do this in the cartesian grid? The same logic you use there should somehow apply to the range time-series? More crucially however, in the later results it seems as if the mean was in fact NOT removed, for example in figure 9b, the centroids all have distinct absolute elevations, surely this can only be possible if the mean was not removed? Otherwise the centroids should all be fluctuating around zero?

– *The first point is obsolete, since we no longer report on the range time series. (Just for information, removing the mean did not change the results significantly in this case.) We did remove the median elevation for all time series to perform the clustering. In the later figures the time series that represent the cluster centroids are shown with the median added for visualization*

purposes. This was not very clear and will be explained. (page 14, line 275)

Eq 2 and L145: notation is not suitable; delta usually refers to a real discreet difference; suggest using prime ‘ as the fluctuating component.

– Ok, will be adapted according to your suggestion. (page 9, equation 2)

Section 3.2: from later on in the results I get the impression that euclideanw as used for k-means and aggregation, while correlation distance was only used for DBSCAN, si this correct? If so, this needs to be stated here.

– Yes, we will add some explanation. (page 9, line 174/175, see also Discussion, section 5.1, page 24)

L179-180: then why don’t you standardize your data?

– Standardizing would make the two distances more comparable, but it does not improve our results. Running the k-means algorithm on standardized time series leads to a separtion into very similar relatively stable clusters and does not detect any of the processes (sand pile, ersoion, inter-tidal zone) that we find without standardization. (section 5.1, page 24, from line 441)

L245: but you also evaluate Euclidean distance in DBSCAN? (or not, see above?) so what are the clusters in that case?

– A short explanation of this case (and why the results are not as good) will be added. (page 24, line 438)

Figure 7: something is seriously going wrong at the white polygons left of centre. The height changes don’t match at all. In A the original elevation in this area is around 5.6 m, in B those two white polygons appear to be at 6.4, so this should yield a height change of 0.9 (red) in C. Or are these polygons areas with No Data? If so the colour scales need to be completely different so as to avoid white being part of the scale (so that it can then indicate no-data). Fig7c: show contour lines of height changes beyond significant (as basis for additional clusters?) Stable points not shown? But not clear then how much of area has been allocated properly?

– Indead the choice of color scale was not very suitable here. White represents no data, as well as stable areas, as well as the ‘stable cluster’. This will be fixed in the future version. The polygons appear as a shadow of the sand pile, where the laser scanner cannot record data. (now Figure 6, page 16)

L304: why six? Not enough justification. What is ‘good’? Why not 8 or 10? Isn’t there a statistic to tell you when to stop clustering?

– See my second comment in the beginning. We will adress this in the new version. (k = 10 gives indeed some more details that are not detected with k = 6.) (section 4.1, page 17/18 and section 5.2, page 25)

Figure 9b: needs horizontal gridlines; labels should be added to lines, rather than a legend (because the sequence in the legend doesn't match the sequence from top to bottom). Vertical axis labels don't make sense: I don't understand how you can have real values asl here for the centroids when the original time-series was mean-subtracted?

– *Yes, labels will be added instead of the legend. See previous comments. (Figures 6,7,8,9)*

L341: this is the only place where you really say that there are no results like 4.2 for the range data; this needs to be made more explicit in Line 300

– *obsolete with removal of processing of range time series.*

L372-374: please elaborate a bit on this here, please summarize or give us a taste of the cause for this 'curse'.

– *more explanation will be added. (page 24, line 430)*

5.2: it only gets clear to me here that you use Euclidean for k-means and aggregation, and corr for dbscan! Conclusion: not really clear what all this work benefits; if you remove the clusters associated with the bulldozer work you basically end up with 4 obvious trends: stable, erosion, accretion, noise. This is a bit underwhelming. . .

– *see previous comments. (section 4.2, page 22)*