TALLINN UNIVERSITY OF TECHNOLOGY
School of Information Technologies

Kristiina Oks

R LIBRARY FOR POST-PROCESSING OF MULTI-TEMPORAL INSAR RESULTS USING MULTIVARIATE OUTLIER DETECTION

Master's thesis

Supervisor: Juhan-Peep Ernits
PhD
Andreas Kiik (AS Datel)

Tallinn 2019

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Kristiina Oks 107665IVSM

R-TEEK AJALISTE INSAR TULEMUSTE JÄRELTÖÖTLUSEKS TUVASTAMAKS MITME MUUTUJAGA VÕÕRVÄÄRTUSI

Magistritöö

Juhendaja: Juhan-Peep Ernits
PhD
Andreas Kiik (AS Datel)

Tallinn 2019

Author’s declaration of originality

I hereby certify that I am the sole author of this thesis. All the used materials, references to the literature and the work of others have been referred to. This thesis has not been presented for examination anywhere else.

Author: Kristiina Oks

07.01.2019

Abstract

Results of multi-temporal InSAR processing of satellite data often contain noise. Common practice for separating reliable results from unreliable is application of threshold on temporal coherence. Excluding results solely based on their temporal coherence may cause areas, which are undergoing complex deformation scenarios, to be left unnoticed. Decrease in temporal coherence may happen for example in case of landslides, earthquakes or subsidence caused by underground mining activities.

The goal of this thesis is to create a free and open source solution for post processing of multi-temporal InSAR results by identifying multivariate outliers in order to improve identification of surface deformation. The created solution is based on approach proposed in [1] . The purpose of created method is to separate multivariate outliers from processed satellite data in order to provide a reliable alternative to practice of applying threshold on temporal coherence.

The expected outcome of the implemented method is to supplement high coherence points with low coherence points which behave in a similar manner and to identify new areas of interest which might be deforming but for some reason experience decrease in temporal coherence.

This thesis is written in English and is 40 pages long, including 9 chapters, 20 figures and 7 tables.

Annotatsioon

R-teek ajaliste InSAR tulemuste järeltöötluseks tuvastamaks mitme muutujaga võõrväärtusi

Satelliitidelt kogutud ja ajalise InSAR töötluse läbinud andmed on sageli mürarikkad. Levinud viis huvipakkuvate objektide tuvastamiseks on ajalisele koherentsusele alumise piirmäära kehtestamine. Tulemuste välistamine pelgalt ajalise koherentsuse põhjal võib viia olukorrani, kus reaalselt vajunud alad jäävad edasise uurimise valimist välja, kuna mingil põhjusel on nende koherentsus piirist madalam. Koherentsuse langus on sagedane juhtudel, kui maapinna deformatsioon toimub prognoositavast mudelist kiiremini – näiteks maanihete ja maavärinate korral, aga sageli ka kaevandamisest tingitud maapinna vajumise korral

Käesoleva magistritöö sisuks on luua tasuta ja vabalt kättesaadav tarkvaraline lahendus, mis võimaldaks eeltöödeldud satelliidiandmete põhjal tuvastada mitme muutujaga võõrväärtusi selleks, et maapinna deformatsioone saaks paremini tuvastada. Töö põhineb metodoloogiale, mis on välja pakutud [1] poolt. Loodud tarkvara ülesandeks on analüüsida mitme muutujaga andmeid, millest tuvastada ja eraldada võõrväärtused, pakkumaks koherentsuse piirmäära seadmisele alternatiivset meetodit võõrväärtuste väljaselgitamiseks. Loodud meetodi käivitamise oodatav tulemus on kõrge koherentsusega punktidele täiendavate huvipakkuvate punktide leidmine, seejuures kinnitades kõrge koherentsusega punktide usaldusväärsust. Lisaks on loodud meetodi eesmärgiks tuvastada täiendavaid piirkondi, kus maapinna nihkumine aset leiab, ent mis jääksid märkamatuks rakendades tavalist koherentsuse alampiiri tehnikat.

Lõputöö on kirjutatud inglise keeles ning sisaldab teksti 40 leheküljel, 9 peatükki, 20 joonist, 7 tabelit.

List of abbreviations and terms

CSV	Comma-separated values
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
DEM	Digital elevation model
ESA	European Space Agency
InSAR	Interferometric synthetic aperture radar
LOS	Line of sight
MAD	Median absolute deviation
MTI	Multi-temporal InSAR
OC	Outlier candidate
PCA	Principal component analysis
PS	Persistent scatterer
ROBPCA	Robust Principal Component Analysis
SAR	Synthetic aperture radar
SNAP	Sentinel Application Platform
UTM	Universal Transverse Mercator

Table of Contents

List of Figures

List of Tables

1 Introduction

European Space Agency has launched family of satellites called the Sentinels, each carrying advanced radar systems and taking images of the Earth. Although satellite information has been available for a while already, the Sentinels provide it much more often and the collected data is available for free.

One of the many applications of using satellite data is measuring deformations of surface. Information about surface deformation can be used for example to monitor subsidence of different areas (e.g. active landslides, volcanic areas, mines etc.) but also could be used to make predictive models on how certain areas might behave or deform. The biggest benefit of using data from the Sentinels satellites for deformation measurements is their free and frequent data delivery. If processed in effective and correct way, they provide ultimate value in gathering information from areas which are difficult to access.

One of the challenges of processing satellite data is how to separate noisy points from points that contain useful information. A common strategy is applying a threshold on temporal coherence. But sometimes important information can be found in low coherent points as well, for example in case of earthquakes or landslides. Time series data does not follow linear model when unexpected changes happen and therefore points in these deformed areas experience decrease in temporal coherence resulting in exclusion from the final results.

Overall goal of this thesis is to build a filter for the points with low temporal coherence with the purpose of keeping the points that contain useful information in the final data set. For this purpose an outlier detection approach is implemented on software level. The techniques used for post processing of data is based on [1] . In the scope of this thesis an R package is created for outlier detection.

The expected value of the implemented method is to provide points of interest in addition to selection made by applying threshold on coherence value. The method aims to supplement high coherence points with low coherence points behaving in a similar manner. As well as to potentially detect new areas which experience decrease in coherence value due to more complex deformation scenarios.

For validation, the results of executing implemented methods are tested on datasets of three different areas where is known that deformations take place. Test data was provided by AS Datel. Results are validated by visualisation of identified points and are compared to the specifics known about areas under investigation as well as using interferograms for two of the test regions.

1.1 Organization of the thesis

The thesis includes following chapters:

- **Background**: an overview of specifics of satellite data processing is given.
- **Problem statement**: motivation, problem statement and alternative implementations are described.
- **Methodology**: a short overview of different algorithms and techniques which are used in the method implementation is given.
- **Method implementation in R**: reasoning for choice of technology, running manual, dependencies on other packages and requirements for input and output data are specified.
- **Step by step implementation**: each step of the implemented method and the most important functions are described in detail.
- **Validation of results**: experimental results and validations of outputs of created method are visualised and explained. Also performance measurements are displayed.
- **Conclusions and further development options**: results of the thesis with suggestions for future work are presented.

2 Background

2.1 Copernicus programme

Copernicus is a joint programme of European Space Agency (ESA) and European Commission [2] . The purpose of the initiative is to provide easily accessible satellite information in order to benefit to the environment, civil security and detection of climate change. Central part of the Copernicus programme is a family of satellites called the Sentinels. Each of the 6 missions are a constellation of two satellites which carry advanced radar instruments to be able to provide images of Earth's surface during day and night, in all weather conditions.

Relevant to this thesis are Sentinel-1A and -1B which were launched respectively in April 2014 and April 2016. They carry a C-band Synthetic Aperture Radar (SAR) which is an enhancement of ESA's and Canada's heritage SAR systems ERS-1, ERS-2, Envisat and Radarsat [3] . The centre frequency of C-band SAR is 5.405 GHz which corresponds to a wavelength of about 5.5 cm.

Sentinel-1A and Sentinel-1B orbit the entire Earth 180° apart each other in every six days, covering large areas on land and sea [3] . The data collected by Sentinel-1 missions is meant for example for monitoring of land deformations, sea-ice mapping and forest and soil mapping.

2.2 Synthetic aperture radar

“Synthetic aperture radar (SAR) is an active, coherent, microwave imaging remote sensing system that can be mounted on an airborne or a spaceborne platform” [4] . SARs are able to deliver images in all weather conditions, during day and night and can be used to create two-dimensional images of three-dimensional reconstructions [5] .

Two different orbital geometries can be distinguished in acquiring of SAR images [6] , enabling areas to be observed from two angles which are almost symmetrical.

Descending geometry means that the satellite is passing from north to south and observing from east. Ascending orbital geometry means that satellite moves from south to north and observes from west. It is possible to measure displacements from SAR images, but only along the satellite's Line of Sight (LOS). So for example when there are ground movements to west, the satellite on ascending orbit seems it as moving near to the satellite, while descending satellite sees it as movement away from the satellite. When both geometries recognize the movement as distancing, it can be said, that the land is subsiding vertically.

2.3 Interferometric synthetic aperture radar

Interferometric synthetic aperture radar (InSAR) is a radar technique which involves interferometric phase comparison of SAR images which are gathered at different times and with different baselines. By using the radar phase information, InSAR measures distance between the satellite and a target point on Earth's surface. InSAR can provide digital elevation models (DEM) with meter accuracy and terrain deformations with millimetre accuracy [7] .

The process of comparing phase information of two (or more) SAR images is called interferometry and as a result, an interferogram is produced. In order to measure deformation of the ground, the images which are used to form an interferogram have to be acquired from different times and from different positions [8] , [9] .

InSAR is used to detect displacements of Earth's surface. For example it can be used to measure and observe changes from earthquakes, volcanic eruptions and subsidence occurring from mining activities. Although there are alternative ways for measuring surface movements (for example GPS), InSAR provides clear advantages by not needing any special equipment on ground in areas under investigation and by covering even the areas which are difficult to access by other means than satellite data.

2.4 Multi-temporal InSAR

Application of InSAR is limited for example due to changes in Earth's surface with time and seasonal changes and variations in atmospheric properties and therefore the displacement measure may be overprinted by noise. In order to overcome this obstacle, Multi-temporal InSAR (MTI) techniques [10] – processing of multiple images in time – can be applied.

MTI techniques are used for measuring deformations and for extracting time-series of LOS surface displacements with millimetre accuracy [7] , [11] . The key idea is to identify persistent scatterers (or permanent scatterers [7]) by using all archived but suitable data of a certain area. Differential interferograms are co-registered to a common master which is chosen from the middle (or near middle) of the time series. Phase of isolated coherent points is analysed as a function of time and space [4] . Persistent scatterers are identified from the stack of co-registered interferograms as points with high coherence.

Interferometric coherence [4] is a correlation measure of the phase noise and phase precision of two SAR images. Coherence value ranges from 0 to 1 – higher value means less noise and more reliable phase measurement. Coherence is affected for example by geometric decorrelation, system noise, temporal decorrelation (due to physical changes in the target), decorrelation due to processing.

2.4.1 Temporal Coherence

Desired estimates of MTI results are for example velocity, height, cumulative displacement which often are considered reliable only when their temporal coherence exceeds some desired limit. Parameter of temporal coherence expresses the quality of fit between deformation model and measurements of phase [12] . Decrease in temporal coherence expresses lack of effectiveness in removing noise by linear deformation model.

Some of the causes for decrease in temporal coherence are: temporal and geometrical decorrelation, noise from signal delays, orbit errors. Temporal coherence decreases also in cases when non-linear movements occur, for example landslide activations and earthquakes.

A common practice in post-processing of MTI results is to apply threshold of 0.7 [1] on temporal coherence in order to distinguish reliable results from noise. The biggest issue with this kind of approach is that non-linear movements may be left unnoticed due to their low or medium temporal coherence.

3 Problem statement

For the last few years, since the Sentinels were launched, it has been possible to get satellite data for free in every six days for the whole Earth. The data is available, but the question of how to process it most efficiently still remains

By using multi-temporal InSAR methodology it is possible to identify offsets of persistent scatterers by magnitude of 1mm. This presents us an opportunity to measure deformation of different areas and buildings anywhere in the world over long periods of time without having to do it manually or without being present physically. In general MTI techniques are successfully applied for measuring subtle deformations of land-surface. But the results often include inaccuracies which cause problems with identifying unexpected deformations. For example causes of inaccuracies might be orbit errors, sub-pixel positions, noise from signal delays, seasonal changes (snow, temporal expansion).

After initial data has been processed, the results, including time series of measurements, might still contain a lot of noise. It is somewhat possible to manually go over measurements to identify which of them are reliable and which are outliers. But this approach requires involvement of an expert and is not cost nor time efficient, especially if there are thousands or even tens of thousands of points under investigation. So the key issue is, how to identify and eliminate outlying observations. The idea of this thesis is to contribute to finding deformations by creating an automated procedure for post-processing of multi-temporal satellite data.

Common practice for dividing observations into reliable and non-reliable is application of lower limit on temporal coherence. Eliminating targets based solely on their low temporal coherence can give incomplete estimates and unreliable results as well as decrease the number of interesting points too dramatically. Especially in cases of rapid, non-linear movements which would not be expected from the linear model behaviour, changes may be left unnoticed. In those cases it might happen that more than half of identified persistent scatterers are with low-medium temporal coherence, although many of them still carry important information.

The scope of this thesis is to take into use methodology proposed in [1] and to create a software component for post-processing of multi-temporal InSAR results by identifying outlying observations. The overall purpose is to be able to identify subsidence of areas by interferometric measurements in an unattended way, while not excluding persistent scatterers solely on their low temporal coherence value.

Expected outcome of the created method is to supply additional areas of interest as well as to confirm the results obtained by coherence threshold (for examples on high coherent areas). Another important outcome is to complement high coherence points with low coherence points which behave in a similar manner.

The required input is a dataset of persistent scatterers which have undergone standard MTI processing. Important variables would be e.g. velocity, height, residual height, standard deviations, temporal coherence. The key idea of the methodology is to separate non-outlying observations from outlying.

Results of the created method are validated by executing outlier detection on data of areas where it is already known that deformation takes place. It is then also possible to compare the remaining relevant scatterers to the result of simply removing low coherence points. This can give an overview of whether the used techniques provided any value.

3.1 Alternative implementations

The author is not aware of any implementation of the approach described in [1] that would be open source and could be used in a processing pipeline. Different steps implemented in current work could manually be carried out in several applications, for example Matlab [13] , but this would require extensive amount of manual work and therefore the approach could not be generalized.

4 Methodology

Functions implemented in scope of this thesis are greatly based on an approach and techniques proposed in [1] . The overall purpose of applying these methods is to distinguish observations which are statistically significant from those which seem to be outlying by identifying location-, data- and application driven outliers and therefore increase the number of points of interest. This section briefly explains theoretical background of algorithms and techniques which have been proposed by [1] and which are the basis for the implementation in R.

4.1 Used techniques

4.1.1 Clustering analysis

Location-driven outliers can be separated by applying clustering analysis on data under investigation. Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm [14] is capable of finding clusters with arbitrary sizes and shapes and additionally to identify noise. The algorithm requires two parameters – *minPts* which is minimum number of points needed to form a cluster. And *eps* which is the maximum distance between two connected points in a cluster.

In order to form a cluster, DBSCAN starts with an arbitrary point p and retrieves all points which are density-reachable from p within distance *eps* with regards to *minPts* [14] . Each point is either assigned to a cluster or marked as a noise point.

Selection of *minPts* is dependent on specific data. *Eps* can be determined by calculating pairwise distances of points in the dataset. Then when sorting the distances and plotting them in ascending order, it is possible to find a knee point of the plotted graph which would represent the optimal value for e*ps* [15] .

DBSCAN is widely used and effective algorithm for detecting clusters of varying shapes but it might fall short of identifying clusters of varying densities. There are several alternatives and additions to DBSCAN which could be used for clustering as well, for example HDBSCAN [16] , OPTICS [17] , DMDBSCAN [15] etc. Concept of *eps* would need to be further analysed when different algorithm is chosen to be used instead of DBSCAN as value of *eps* has a key role in the final step of processing outlier candidates (OC). If a chosen algorithm does not involve a value for maximum distance of a neighbouring point (*eps* or similar), then this distance would have to be calculated or set based on some other calculation. In scope of this thesis, the DBSCAN seemed to provide sufficient value and good results in data clustering.

4.1.2 Principal component analysis

Principal component analysis (PCA) [18] is a widely used technique which is capable of discovering multivariate outliers. Overall purpose of PCA is finding a small number of linear combinations of correlated parameters in order to convert them to a set of linearly uncorrelated variables (principal components) which would describe most of the variation in the dataset [19] .

PCA is very sensitive to outliers, but there are some modifications of the technique which are more robust, outlier-resistant and can be used for outlier detection purpose. In this thesis Robust PCA approach (ROBPCA) [20] is used. The purpose of this technique is to find linear combinations of original variables which contain most of the information despite presence of outliers and hence to identify outlying observations.

In general, for each observation, their score and orthogonal distances are calculated. By applying a cut-off value, it is possible to identify outliers – observations which do not behave in a similar manner as majority of observations are marked as outliers.

4.1.3 Analysis of detected outliers

The proposed procedure for final processing of outlier candidates found by ROBPCA consists of several steps. The goal here is to investigate further the behaviour of all detected outliers in context of their allocated cluster.

First, by calculating Delaunay triangulation and Voronoi diagram [21] , [22] , it is possible to separate grouped and isolated outliers. Grouped outliers are points that share Voronoi adjacency cells and are within Eps radius (the same as used in DBSCAN).

Then calculation of Median Absolute Deviation (MAD) [23] is used to estimate each variable in each point for both isolated and grouped outliers. MAD allows to compare a certain point to the behaviour of non-outlying part of the cluster.

The final step is calculating pairwise Jaccard index [24] for outliers in every group in order to compare their similarity and diversity. The aim of this comparison is to discover set of points in near distance which behave in a similar (outlying) manner. When such groups are found, they become relevant despite their previous classification as outliers.

4.2 Alternative approaches

There are several techniques which try to automate extraction of meaningful data of MTI results. Most common is thresholding on coherence. Another way is focusing solely on velocity. But these are not multivariate analysis and therefore their efficiency can be rather limited. There are alternative approaches for exploiting low coherent areas e.g. [25] ,[26] .

One additional approach for effectively detecting multivariate outlier could be [27] which uses information of time series for continuous monitoring of ground deformations.

5 Method implementation in R

Most suitable programming language for implementing multivariate outlier detection was chosen by three main criteria

- support for statistical computing
- free-ware
- high integration options with other systems and potential work flows

Considering all prerequisites, choice of technology was made between Python and R. After some experiments, the final decision was made in favour of R, as it seemed to have even better and faster support for some of the statistical methods needed in the implementation (for example DBSCAN, ROBPCA).

5.1 Running manual

Package *outdetect* was created in scope of this thesis. In order to run the outlier detection process, the package needs to be cloned from git repository *https://github.com/kristiin/outlier-detection* and then locally installed as any other R package.

The main method is executed by running following command:

```
outdetect::extractOutliers(data)
```

5.2 Dependencies

Most important dependencies of the created R package are on libraries*, deldir [28] , dbscan [29] ,rrcov [30] , igraph [31] , sp[32]* . All dependencies are listed in project directory in file *DESCRIPTION* under section *Imports*. When installing package *outdetect*, all required packages listed in *Imports* get installed as well.

Library *testthat* [33] is used for unit testing. *Testthat* is listed as *Suggests* in DESCRIPTION file. This means that the package is not needed to execute the main code, but it is necessary when wanting to run the tests and therefore has to be installed manually.

5.3 Requirements for input data

The purpose of created package and its functions is to further analyse satellite data which has already gone through standard processing by MTI techniques. In a work-flow context, MTI results are retrieved either by direct select from some database or by e.g. a CSV file. The aim of the created code is to work regardless of the specific data source, therefore at this stage the functions have been created with an assumption that input parameters are R data-types. No support for querying database or reading files from disk has been provided as this does not seem relevant at this point and can easily be added once the need for it arises.

5.3.1 Arguments of ***extractOutliers()***

All input arguments, their type and default value for optional arguments are given in Table 1.

Table 1. List of arguments for *extractOutliers.*

Name	Type	Description	Default value
data	data frame	MTI data to be analysed for outliers	
minPts	integer	Minimum number of points to form a cluster for DBSCAN. This is highly dependent on specific data to be analysed.	3
eps	numeric	Neighbourhood radius for DBSCAN. If no value is provided, then the optimal value is calculated. Considered as euclidean distance.	NULL
utm	integer	Universal Transverse Mercator (UTM) zone	NULL
minCoher	numeric	Coherence threshold	0.7
k	integer	Number of principal components to be retained ROBPCA	2
cl	numeric	Confidence level for ROBPCA	0.9
rejCrit	numeric	Conservancy index for data when comparing to median absolute deviation	3
minJacc	numeric	Threshold for pairwise Jaccard index used in application driven outliers	0.6

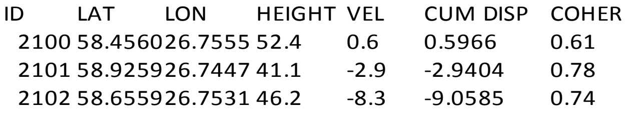
5.3.2 Required columns of *data* and their order

All required columns which have to be present in input data frame with their sequence number are given in Table 2.

Table 2. Required columns of data.

Name	Description	Sequence no
ID	ID of observation	1
LAT	Latitude value in coordinate system WGS84	2
LON	Longitude, coordinate system WGS84	3
COHER	Coherence with value between 0 and 1	Not relevant

All other columns present in the input data frame are considered as variables and are included as basis for detection of multivariate outliers. Examples of usual variables are velocity and its standard deviation, height at sea level and its standard deviation, height with regards to digital elevation model, cumulative displacement. Example of a structure of *data* argument is given in Figure 1.



```
ID      LAT      LON      HEIGHT  VEL     CUM DISP   COHER
  2100 58.456026.7555 52.4    0.6     0.5966     0.61
  2101 58.925926.7447 41.1    -2.9    -2.9404    0.78
  2102 58.655926.7531 46.2    -8.3    -9.0585    0.74
```

Figure 1. Example of input data structure.

5.4 Output requirements

Output value of function *extractOutliers* is a list providing following elements:

nonoutliers – dataframe of all points which were found as non outlying

outliers – dataframe of all points which were found to be outliers

params – list of parameters used in data processing

6 Step by step implementation

This section describes main functions which were created in package *outdetect* as well as their required input parameters and dependencies on other R packages.

The entry point of package *outdetect* is function *extractOutliers*. Its implementation consists of three logical steps

1) Input data validation and transformation

2) Detection of location, data and application driven outliers.

3) Extracting and binding outliers and outlier free data

Example call of *extractOutliers*:

```
outdetect::extractOutliers(data)
```

6.1 Input data validation

When calling the method *extractOutliers*, input data is validated for structure correctness by function *validate*.

Validations:

- First three columns must be in order *ID, LAT, LON*
- Presence of column *COHER* with values between 0 and 1.
- All fields in *data* must be in numeric form.
- All input parameters are numeric.
- *minPts* greater than 0.
- *utm* in range 1 – 60 if present.

If parameter *utm* is present, then coordinates are transformed into applicable coordinate system with corresponding UTM zone by function *prepareData*. If *utm* is not present, then coordinate system EPSG3359 [34] is used and added to the dataframe as LAT.1 and LON.1. Function *spTransform* in package *sp [32]* is used for transformation of coordinates. Reasoning for the transformation is explained in 6.2. Value of *utm* (or its presence) does not have much impact on the end result (difference might be only a few points), but it benefits mostly to debugging and to investigation of clusters and groups. The more accurate the provided UTM zone is for an area under investigation, the closer the value of calculated *eps* is to actual distance in meters.

6.2 Location driven outliers

The purpose of this step is to cluster data based on geographic coordinates and extract noise points.

Function *getDbscan(data, params)* is used for extraction of location based outliers. DBSCAN clustering is available by function *dbscan* of package *dbscan [29]* . The *dbscan* implementation is very fast but its calculations are based on Euclidean distances of coordinates and it does not support clustering of geographic coordinates with great-circle/spherical distance.

While it would be possible to pre-calculate distance matrix based on geographic coordinates and pass it to *dbscan,* it is much faster to simply transform coordinates into suitable system (done by *prepareData()*) and perform clustering based on the transformed values. Due to the nature of the problem at hand, the pairwise distances of points are rather small and therefore the Euclidean versus great-circle/spherical distance do not differ much from each other.

In case value for *eps* has not been given, it is found by finding a knee point in a curve of calculated pairwise distances of points. Although there is a way for calculating the optimal value of *eps,* one might want to set it manually. Especially when its clear that there are clusters with different densities. Therefore it is best to be able to set the value manually as well.

As a result of clustering, each point gets assigned to a cluster. All noise points are assigned to cluster 0. Cluster with the most points belonging to it, is set as GROUP cluster – to be used in the next steps.

6.3 Data driven outliers

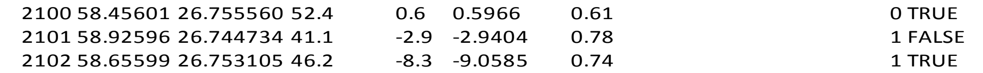
Central function for extracting data driven outliers is *applyPCA(data, params).* This function is responsible for applying robust principal component analysis (ROBPCA) on initial data. For execution of ROBPCA function *PcaHubert* in package *rrcov [30]* is used. Parameters to follow in this step are *cl,* which is used for computing the cut-off values for the orthogonal and score distances, and *k* which is value for number of principal components.

Lowering value of *cl* results in larger number of outlier candidates identified by ROBPCA, hence more points get passed to next steps of processing and therefore the execution time may be longer. Increasing value of *k* has the same impact. The default values are *cl=0.9* and *k=2* as used in [1] .

It is important to select relevant variables for comparison inside the method. As prerequisite of the input stated, all fields except for ID, LAT, LON are considered as variables describing the behaviour of the point. Therefore all other initial fields are subject to principal component analysis.

Return value of *applyPCA* is the input data with additional column *ISCORE* for each column. *ISCORE* is a representation of whether a certain point was identified as a core point (*true*) or as an outlier (*false*) by ROBPCA.

From here on, all next steps and calculations require data with outlier and noise indications. Example set of data is shown in Figure 2.



```
ID     LAT       LON        HEIGHT   VEL   CUM DISP   COHEFLAT.1   LON.1   CLUSTER   ISCORE
  2100 58.45601 26.755560 52.4      0.6   0.5966     0.61                          0 TRUE
  2101 58.92596 26.744734 41.1     -2.9   -2.9404    0.78                          1 FALSE
  2102 58.65599 26.753105 46.2     -8.3   -9.0585    0.74                          1 TRUE
```

Figure 2. Data with outlier candidacy information.

All points with *CLUSTER!=0* and *ISCORE==TRUE* are kept in the final dataset as non-outliers. These are points which were not classified as noise by DBSCAN, and at the same time were also found as non-outlying by PCA.

6.4 Application driven outliers

6.4.1 Overview

All outlier candidates which were identified by ROBPCA and were not marked as noise by DBSCAN are processed in this step. The aim is to analyse these outlier candidates even further to identify if they act as outliers on their own or perhaps they form a group of outliers – in case they are located in "near enough" distance (*eps* radius) and behave in a similar manner, they could be included into final non-outlying dataset.

Top level function which is responsible for detecting application driven outliers is *processOCs(pointsWithOCFlag, madOfVariablePerClusterDF, params)*.

Arguments:

- *pointsWithOCFlag – o*riginal data with additional columns of cluster and outlier flag.
- *madOfVariablesPerClusterDF* – result of *calculateMadOfVariablePerCluster()* (described in 6.4.2).
- *params* – list of additional parameters. Relevant in this step are: *eps, minCoher, cl, k, rejCrit, minJacc.*

6.4.2 Calculation of MAD

Function *calculateMadOfVariablePerCluster()* is responsible for one of the key steps in processing outlier candidates.

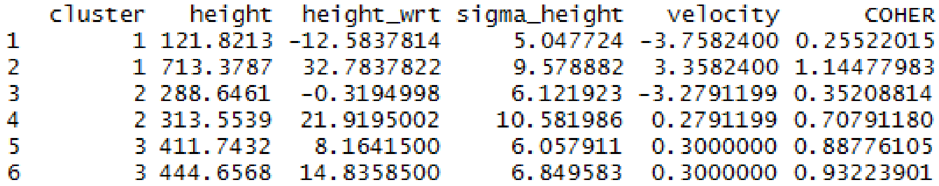
Prior to any further processing of outlier candidates found in previous step, it is needed to calculate upper and lower borders for each variable in each cluster. Calculations are based on median absolute deviation [23] :

$$\frac{xi-M}{MAD}>|\pm 3|$$

Specified rejection criterion with values 3 as very conservative, 2.5 moderately conservative and 2 as poorly conservative. The index is specified by parameter *rejCrit*.

In case there are clusters which have more outlying points than non-outlying points, borderline data of core group clusters are used.

Example of a return value of *calculateMadOfVariablePerCluster()* is shown in Figure 3. The data frame contains minimum and maximum values of each variable in each cluster based on non-outlying observations. Cluster based MAD information is used in final decision making steps described in 6.4.4 and 6.4.5.



```
  cluster   height  height_wrt sigma_height   velocity      COHER
1       1 121.8213 -12.5837814     5.047724 -3.7582400 0.25522015
2       1 713.3787  32.7837822     9.578882  3.3582400 1.14477983
3       2 288.6461  -0.3194998     6.121923 -3.2791199 0.35208814
4       2 313.5539  21.9195002    10.581986  0.2791199 0.70791180
5       3 411.7432   8.1641500     6.057911  0.3000000 0.88776105
6       3 444.6568  14.8358500     6.849583  0.3000000 0.93223901
```

Figure 3. Example of MAD per cluster

6.4.3 Division of outliers

All points detected as OCs by ROBPCA and not classified as noise by DBSCAN are processed. First, Delaunay triangulation and Voronoi tesselation are calculated by *deldir* function in package *deldir* in order to identify neighbouring points which share Voronoi adjacency cells.

Then euclidean distance between each point pair in triangulation is calculated. Outlier candidates are separated into two – outlier candidates which have any other outlier within distance *eps* (the same as used in clustering) are marked as grouped outliers. All others are considered isolated.

6.4.4 Grouped outliers

We now have set of point pairs with pairwise distance less than or equal to *eps.* The structure allows to apply graph theory grouping to separate outliers into groups. Function *graph_from_data_frame* of package *igraph [31]* is used to make an undirected graph of non-isolated outlier candidates. It is then possible to extract groups of connected outlier candidates from the graph.

Then, for each outlier in a given group, a rejection table (described in 6.4.6) is calculated to indicate if a specific variable exceeds rejection criteria (1) or not (0). Based on rejection table, Jaccard difference and similarity are calculated.

If there are no point pairs with smaller index than selected similarity threshold *minJacc*, then all points are kept in the outlier-free dataset. Otherwise only the points with coherence more than initially selected minimum are kept. Smaller value of *minJacc* means more conservative approach in deciding whether outlier candidates behave in a similar manner and therefore might cause smaller number of non-outlying points to be present in final results. Default value is 0.6 as was used in [1] .

6.4.5 Isolated outliers

All outlier candidates which were not classified as grouped, are considered as isolated. Each isolated outlier with coherence greater than selected minimum *minCoher*, is then processed. For each variable rejection table is calculated. If there are no dissimilarities found, then the outlier candidate is kept in the final outlier free dataset. Otherwise the point will be marked as an outlier.

6.4.6 Calculation of rejection table

Function *calculateMatrixPerOc(noiseFreeOutlierCandidates, ocInds, madOfVariablePerClusterDF)* is responsible for calculating a rejection matrix for every outlier.

NoiseFreeOutlierCandidates is a set of points which were used for calculation of Voronoi tessellation.

Argument *ocInds* is either a list of indexes or a single index (in case of an isolated outlier) of a record in *noiseFreeOutlierCandidate* dataframe.

Each variable of every outlier candidate is compared to corresponding range per cluster in *madOfVariablesPerClusterDF.* Each variable is then flagged as 0 (within the limits) or 1 (outside the limits) as illustrated in Table 3.

Table 3. Example of a structure of rejection table for group of outlier candidates.

ID	HEIGHT	VEL	CUM DISP	COHER
2100	1	1	1	1
2101	1	1	1	0
2102	0	1	1	1

6.5 Returning results

Return value of *outdetect::extractOutliers* is an object consisting of fields *outliers, nonOutliers*, *params.*

6.5.1 NonOutliers

Points identified as non-outlying by ROBPCA.

Points identified as outlying by ROBPCA and as noise (cluster 0) by DBSCAN and with coherence value more than *minCoher*.

Isolated outlier candidates with coherence more than *minCoher* and none of the variables exceeding rejection criteria.

Grouped outlier candidates with similar behaviour or grouped outlier candidates with coherence value more than *minCoher.*

6.5.2 Outliers

Points identified as outliers by both ROBPCA and DBSCAN and coherence less than or equal to *minCoher.*

Isolated outlier candidates with coherence less than or equal to *minCoher* or at least one value exceeding rejection criteria.

Grouped outlier candidates from groups which do not behave in a similar manner and with coherence less than or equal to *minCoher*.

6.5.3 Params

For information purpose. Includes all values of input parameters, calculated value of *eps,* names of all the fields which were used in ROBPCA.

7 Validation of results

Validation of results of the implemented algorithm is not very straightforward as the desired outcome is not a specific value which could easily be defined. The expected output of the method is a set of additional points, which indicate that they are suitable for further analysis. One of the greatest challenges of satellite data processing in general is the problem of identifying relevant information from noise and acquiring meaningful information. In smaller scale, the challenge of MTI results is to increase number of relevant points, since threshold on coherence often leaves out areas of significant importance.

The effectiveness of the method can be tested by executing the method on regions where it is known for a fact that deformations take place. Then the results obtained by multivariate outlier detection can be visualised in order to compare them to what would be the expected behaviour.

Tests were performed on three different data sets and results are presented in this section. The data was provided by AS Datel in a CSV format. Data originates from Sentinel-1 satellites and has undergone MTI processing. While the original unprocessed satellite data is freely available [35] , the processed results in a CSV format were provided as confidential information by AS Datel.

Tools which have been used for visualisation purposes for this section are QGIS [36] , RStudio [37] and Google Earth [38] .

7.1 Rattlesnake Hills Landslide

Rattlesnake Hills is a mountain ridge south of Yakima, WA, USA. In late 2017 a large landslide was reported to be occurring in the ridge with blocks of basalt sliding on a weaker sedimentary layer. Moving rate was reported to be around half a meter per week in a southward direction. It has been estimated by observing geologists and engineers that most probably the landslide will keep moving south until accumulating into the quarry [39] .

Approximate extent of the landslide and its movement direction are visible in Figure 4 and Figure 5.

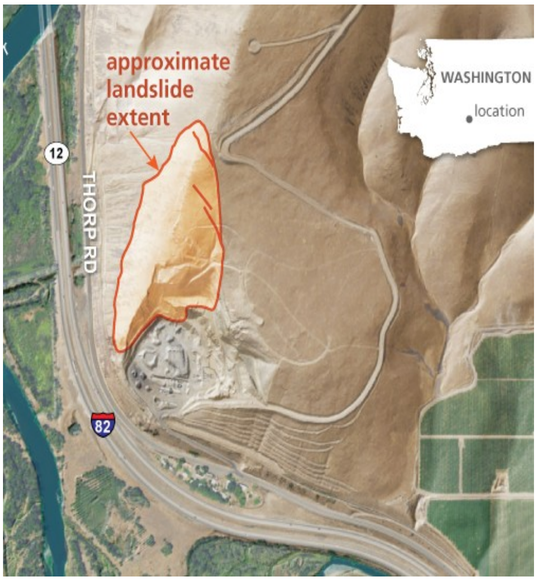


Figure 5. Approximate landslide extent [39] .

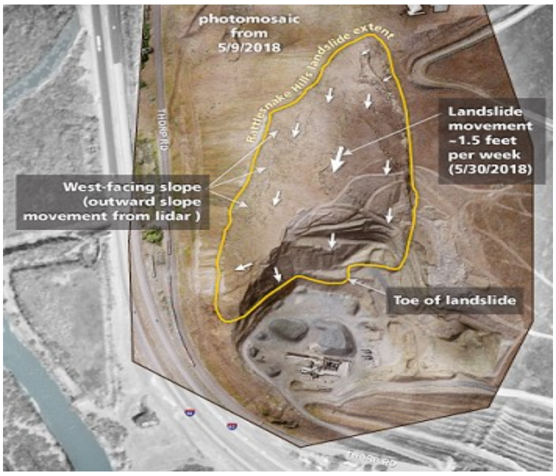


Figure 4. Movement direction [39] .

7.1.1 Test data of Rattlesnake Hills

The dataset consists of 4710 points, with temporal coherence from 0.38 to 0.96 and was acquired from Sentinel-1 relative orbit number 115 between 06th November 2014 and 7th January 2018. Variables presented in MTI results are height, height with regards to DEM, standard deviation of height, velocity, standard deviation of velocity, cumulative displacement and temporal coherence. When applying the standard threshold of 0.7 on coherence, only 1902 points remain.

Propagation of temporal coherence and velocities of high coherent points are shown in Figure 6. As can be seen from images, majority of high coherent points do not express any significant movement and only few points with increased movement rate have been identified in the landslide area.

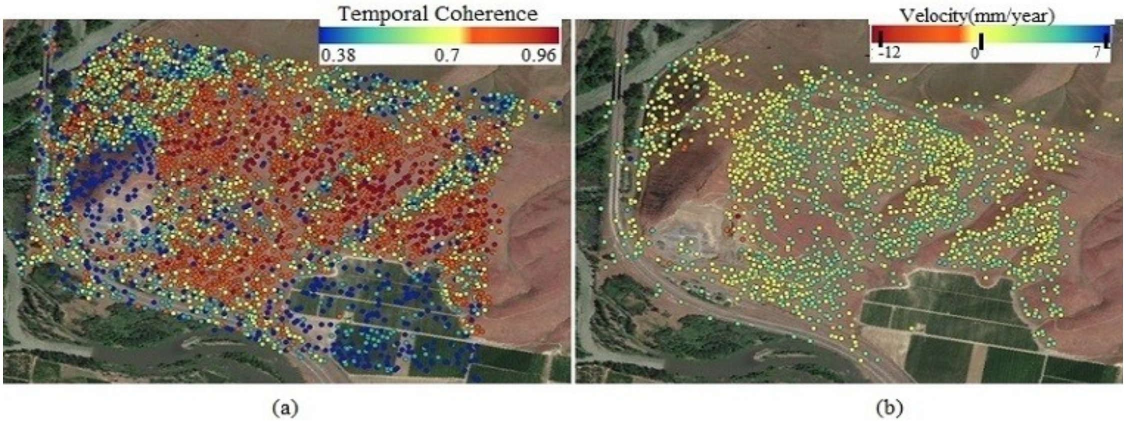


Figure 6. Temporal coherence (a) and velocity (b) of Rattlesnake area (image from QGIS).

7.1.2 Results of executing multivariate outlier detection

Method was executed with command: *outdetect::extractOutliers(data, utm=10)*

Location based clustering identified 35 clusters and 102 noise points. 4325 points were assigned to core group cluster. Application of robust principal component analysis identified 3342 points as non-outlying and 1368 as outliers. Histogram of the overall coherence versus coherence of points identified as outliers by principal component analysis is given in Figure 7. 1305 outlier candidates remain when excluding points identified as noise by DBSCAN.

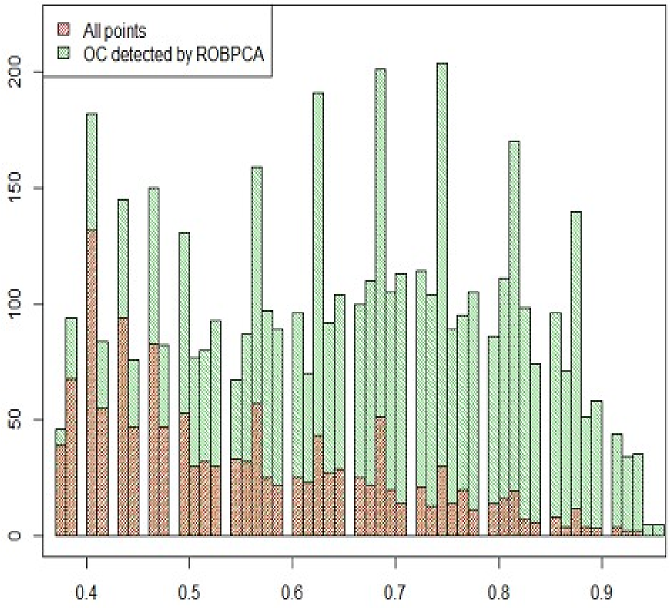


Figure 7. Histogram of temporal coherence of Rattlesnake (image from RStudio).

Next steps identified additionally 55 low coherent points as non outlying. Several of additional points are located in the most active landslide area (illustrated in Figure 8).

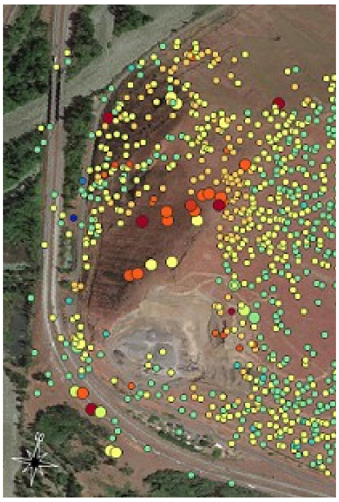


Figure 8. Velocities of grouped outlier candidates to be included as non-outliers (large dots) in context of high coherent points (small dots), Rattlesnake (Image from QGIS).

As final result, 3608 points were identified as non-outlying and 1102 as outliers. Most of additional non-outlying points support the model of high coherence points. Additional deforming area has been identified on the landslide area which would have been left unnoticed by the standard coherence threshold procedure. Velocities of all identified points are visible in Figure 9.

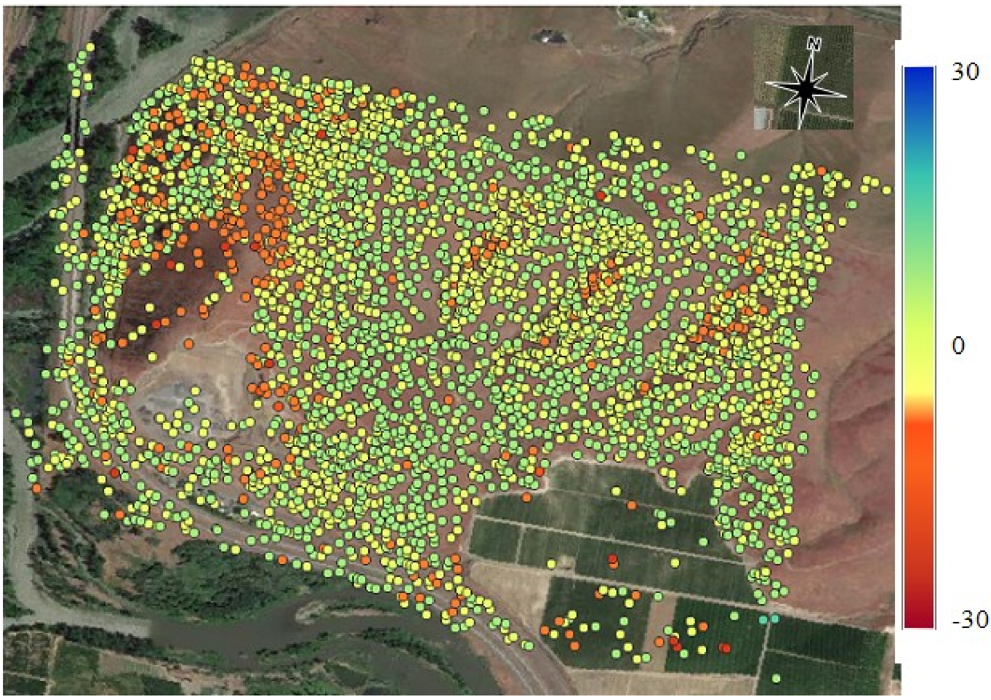


Figure 9. Velocities of all identified non-outliers (image from QGIS)

When comparing the results of Figure 9 to Figure 6 (b) and to illustrated movement directions presented in Figure 4, then it can be seen, that new areas of interest have been identified by the applied outlier detection method.

7.1.3 Comparison to an interferogram

Another possibility for validation could be comparison of identified points with an interferogram of an area under investigation. The Sentinels data is available for download on Copernicus Open Access Hub [35] . Collection of Sentinel Toolboxes called Sentinel Application Platform (SNAP) [40] can be used for generating SAR interferograms. Manual for creating SAR interferograms is available by ESA on [41] . A plugin named SNAPHU [42] can be used for phase unwrapping in order to eventually visualise displacements.

Successful generation of a SAR interferogram is somewhat tricky, because the data files are large (around 5-10GB), SNAP requires extensive amount of RAM and tends to occasionally fail in the middle processing. Also, the resulting interferogram is very dependent on selected time periods.

Interferogram of data from 20th November 2017 and 24th February 2018 is visible in Figure 10 (applied as KMZ on Google Earth) as a result of executing Goldstein phase filtering and terrain correction [41] as final steps of processing. The legend in Figure 10 expresses value of phase which ranges from -π to π. The full colour cycle is called a fringe (e.g. from blue to blue as seen on the image) and the "pixelated" area indicates noise. The fringe expresses change of elevation in half the wavelength of sensor (around 5.4cm for C type used in Sentinel 1) along the line of sight. The actual displacement cannot be estimated directly from examining the fringes and need to be unwrapped instead.

Displacement map created by phase unwrapping, phase to displacement and terrain correction steps is shown in Figure 11. Additional points (without high coherence points) coloured by cumulative displacement (movement of the point with regards to the first time series) are represented in Figure 12.

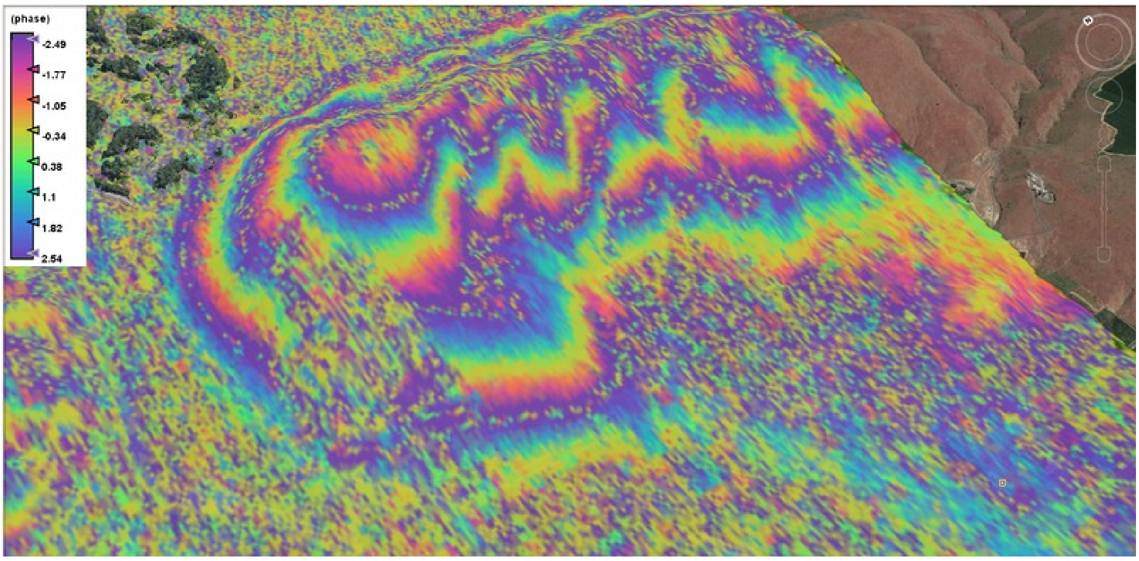


Figure 10. Interferogram 20.11.2017/24.02.2018 (result after Goldstein Phase Filtering) of Rattlesnake applied on Google Earth.

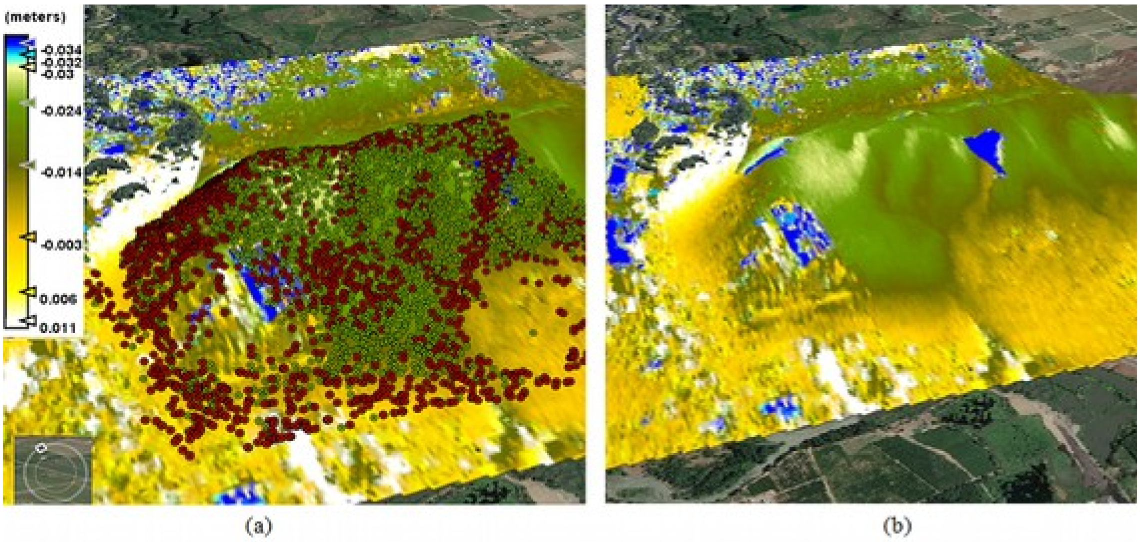


Figure 11. Interferogram 26.12.2017/24.02.2018 (result after Phase To Displacement) of Rattlesnake applied on Google Earth

(a) Interferogram with points - high coherence points are in green and additional identified points are in red (b) Interferogram without points

The method implementation appears to fulfil its purpose on Rattlesnake ridge dataset by identifying new areas of deformation which experience more rapid movement than was expected from linear model. Additional low coherent points were identified supporting high coherent points behaving in a similar manner. When taking into account that landslide was reported only late 2017 but time frame of data was between 06th

November 2014 and 7th January 2018, then it can be concluded that the method implementation proved to be effective on detecting unexpected movements caused by landslide.

The only concern with the results seems to be south-east part of the selected area where some low coherent points with significantly low velocities were identified. This could be something to investigate further in context of whether these points are just noise – and if they are, then how to further improve the outlier detection process.

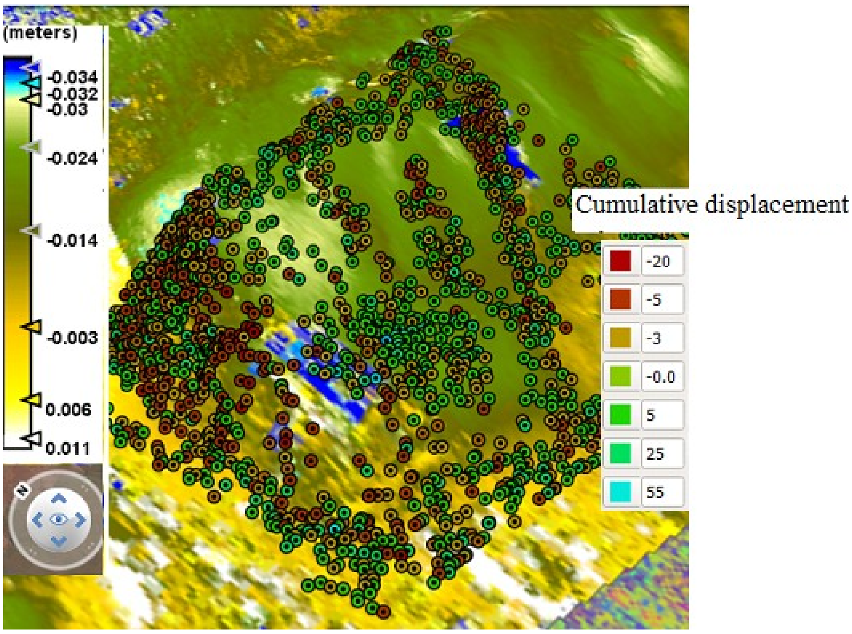


Figure 12. Additional points identified by outlier detection, Rattlesnake (image from Google Earth).

7.2 Kiruna mine

Kiruna mine is a large scale mine in Sweden, near city of Kiruna, owned by Luossavaara-Kiirunavaara AB and producing 28 Mt of iron ore per year [43] , [44] . It was transformed from an open pit to an underground mine in 1960s. The mine is oriented from north to south, with foot-wall on its west and hanging-wall on east. Ore body positions at around 60-degree angle in the bedrock underneath the city.

Deformations have been identified on both hanging-wall and foot-wall [44] , [45] . Surface deformations in hanging-wall are caused by collapsing cavities which are left in the bedrock by mining activities. These deformations are moving towards the city centre of Kiruna, situated on the hanging-wall side of the mine, forcing the entire city to relocate around 3.2 km east by 2033.

7.2.1 Test data of Kiruna mine area

Test data was retrieved from both ascending and descending orbits and have gone through MTI processing. Summary of points of both orbits is given in Table 4.

Table 4. Overall parameters of test data of Kiruna mine.

	Initial data				
Orbit	**Relative orbit number**	**From**	**To**	**Total number of points**	**Temporal coherence > 0.7**
Ascending	58	14.05.17	05.10.17	29381	21060
Descending	95	11.05.17	14.10.17	28184	18838

Propagation of temporal coherence and velocities of high coherent points for each orbit geometry is visible in Figure 13 and Figure 14.

Interferogram of ascending orbit between dates 14th May 2017 and 5th October 2017 is visible in Figure 15. While phase interferogram does not give information on exact measures of displacement, it can still be used to indicate whether the movement is present.

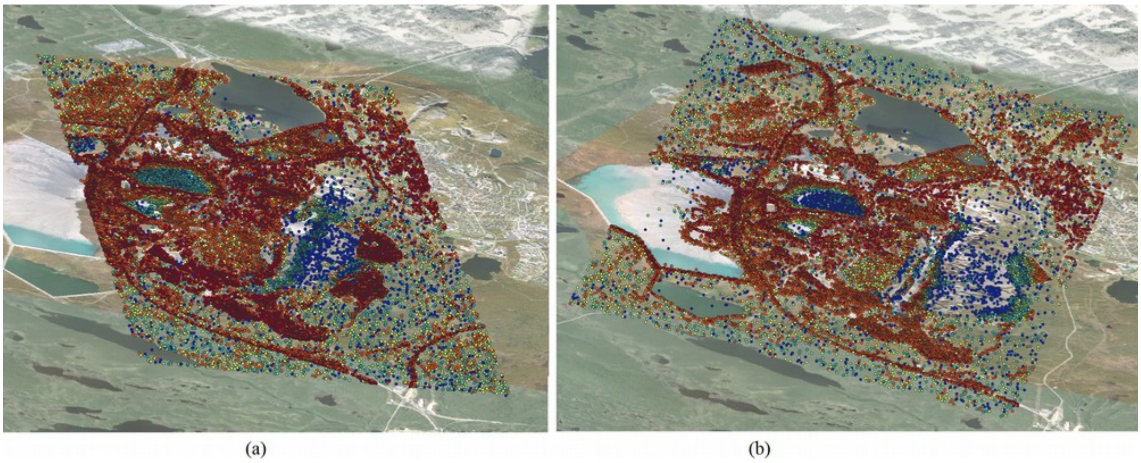


Figure 13. Coherence of Kiruna mine area (a) ascending orbit (b) descending orbit (images from QGIS).

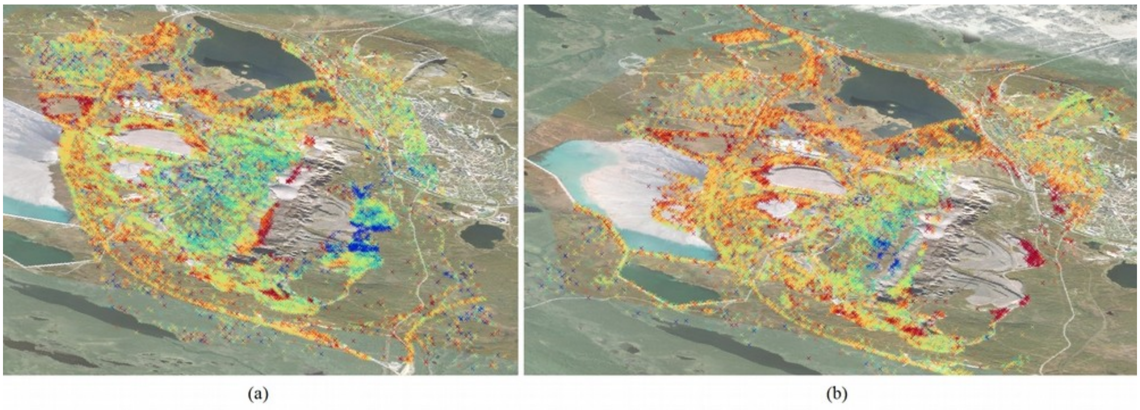


Figure 14. Velocities of points with high coherence in Kiruna (a) ascending orbit (b) descending orbit (images from QGIS).

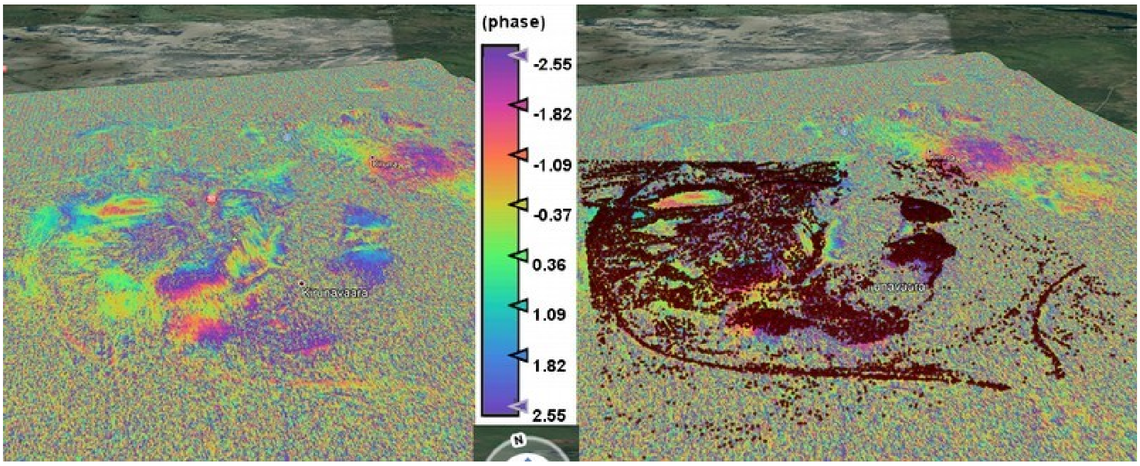


Figure 15. Phase interferogram. Descending orbit, 14.05.2017 vs 05.10.2017. (a) – interferogram (b) – propagation of high coherence points.

7.2.2 Results of executing outlier-detection

Results of executing outlier detection with parameters *minPts = 5* and *utm=34* is shown in Table 5. Velocities of additional points identified as non-outlying in context of high coherent points are visualised in Figure 16.

The method identified an area on the foot-wall side for both orbits (Figure 16, white rectangle) which shows movement also on the interferogram. This is an indication that one area of interest has been recognized and could be included into further investigation.

Table 5. Results of applying multivariate outlier detection on Kiruna mine dataset

			After outlier detection	
Orbit	**Total number of points**	**Temporal coherence > 0.7**	**Outliers**	**Non-outlying points**
Ascending	29381	21060	3803	24418
Descending	28184	18838	4902	23282

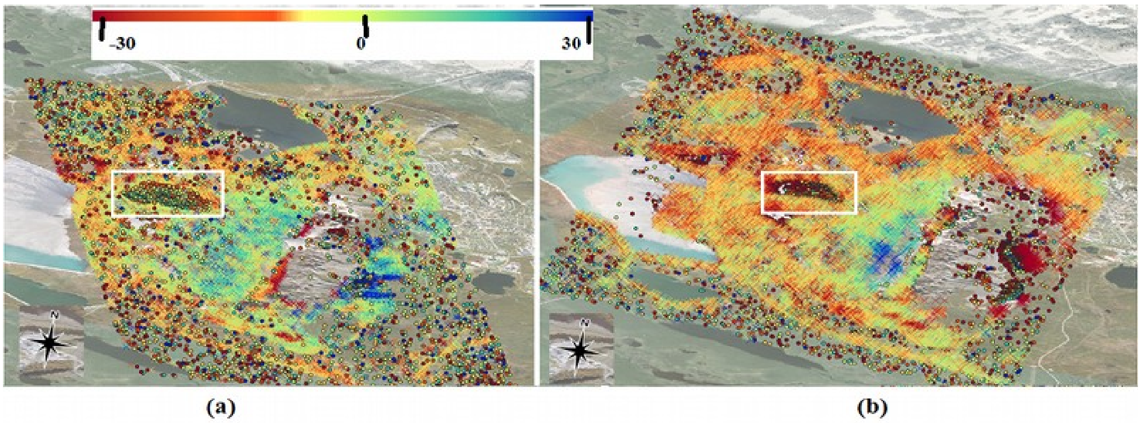


Figure 16: Kiruna results of executing outlier detection. (a) - ascending orbit (b) - descending orbit. Dots express additional points identified by outlier detection method (images from QGis).

From Figure 13 and Figure 14 it is clearly visible that a large area in the east is uncovered by application of coherence threshold – the area where the mine ridge is. For experimental purposes this smaller area can be chosen to be observed more closely.

7.2.3 Results of small area of Kiruna mine

Input data and results of outlier detection on smaller area of Kiruna mine are given in Table 6. Both tracks had rather low coherent results, especially for the descending orbit. After execution of outlier detection, number of points identified as non-outlying increased severely in both cases. Velocities of identified points are visible in Figure 17 and additional points identified on ascending geometry are illustrated in context of interferogram in Figure 18.

Table 6. Results of applying multivariate outlier detection on small region of Kiruna mine.

			After outlier detection	
Orbit	**Total**	**Temporal coherence > 0.7**	**Outliers**	**Non-outlying points**
Ascending	7825	4799	882	6943
Descending	5859	2633	853	5006

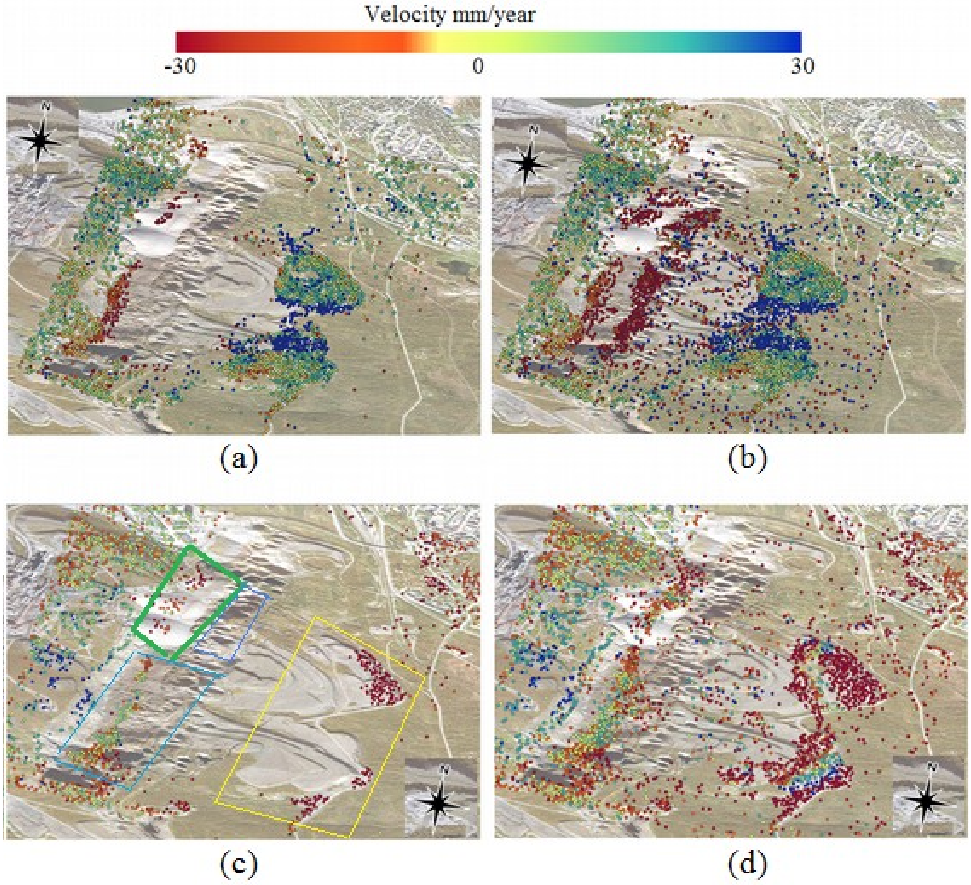


Figure 17. Kiruna high coherent (HC) vs all identified points (a) ascending orbit HC, (b) ascending orbit all points, (c) descending orbit HC, (d) descending orbit all points (images from QGIS).

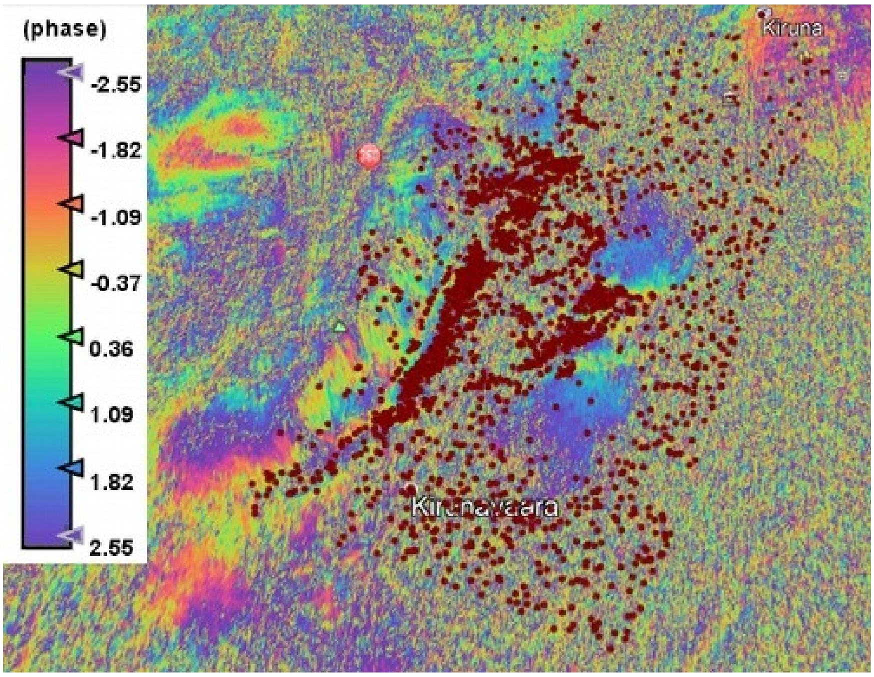


Figure 18. Kiruna (small area) identified low coherent points of ascending orbit with interferogram (image from Google Earth).

7.2.4 Interpretation of results

In case of ascending geometry, satellite moves from south to north, observing from west to east. Hence, all points visible with positive values of velocities in figures 13-17 for ascending geometry are movements towards the satellite, while negative values indicate movement away from the satellite. And it is the opposite situation with descending geometry. When value is in the same direction in case of both geometries, then this most probably implies vertical movement.

Several observations could be pointed out from results presented above:

1) Small deformation area (green tetragon in Figure 17 (c)) has been identified by high coherent points as subsiding by both ascending and descending orbits. Very few extra points were identified in addition to high coherent points by outlier detection performed on larger area. Processing of small area, on the contrary, provided many additional points of interest and expanded the small subsidence area. So it can be said that the method provided value in context of complementing high coherence areas with additional points.

2) Quite a small area of high temporal coherence has been identified by descending orbit – marked with yellow tetragon in Figure 17 (c) – while coverage from ascending orbit is quite extensive. It can be assumed that in this area some movement from east to west is taking place. For both orbits, complimentary motion areas were detected, assuring correctness of the initial model and expanding the area under investigation.

3) Rather few points with high temporal coherence (light blue) or no points at all (dark blue) were identified in areas marked by blue tetragons in Figure 17 (c) by either of satellites. After executing outlier detection, ascending orbit identified new areas of interest in areas marked by both tetragons, while descending mostly in area of light blue. When comparing results to an interferogram of phase change, it appears that deformations most probably take place and the area could be taken for further investigation.

4) Comparing the processing results of larger area (Figure 16) to smaller area (Figure 17), we can see that local results of area in yellow tetragon are quite similar while results of regions of blue and green tetragons differ a lot. This raises a topic which could be investigated in further work – how to assure an optimal region selection for multivariate outlier detection.

7.3 Bytom

Bytom is a town in Silesia, Poland located in Upper Silesian Coal Basin. Underground coal mining is a common reason for subsidence caused by human activities. Widespread subsidence has been registered in Bytom which has caused for example building and road breaks as well as railway track movements. The deformation from underground mining and damages to infrastructure have been widely investigated for several years [46] , [47] .

Two areas which are well known and investigated for deformation are Karb and Miechowice [48] , [49] and have therefore been selected as test areas for multivariate outlier detection in this section.

7.3.1 Test data of Bytom area

The test data originates from two different orbits of Sentinel-1 – relative orbit number 124 with descending pass direction (01.12.2014 – 12.08.2018) and relative orbit number 175 with ascending pass direction (26.06.2015 – 20.09.2018). In original data provided by AS Datel there were 41718 points identified from descending direction with only 377 low coherence points. And 64150 points from ascending orbit of which 848 were low coherence points.

For testing purposes a small selection of data has been used which covers Karb and Miechowice areas. Points with their temporal coherence are shown in Figure 19. 1183 points have been selected from descending orbit (a) of which 83 have low temporal coherence. And 1422 points from ascending orbit (b) with 73 low temporal coherence points. Overall it can be said that in this test case outlier detection is applied on very coherent data set and therefore it provides additional value to previous test sets which have medium and high amount of points with low coherence.

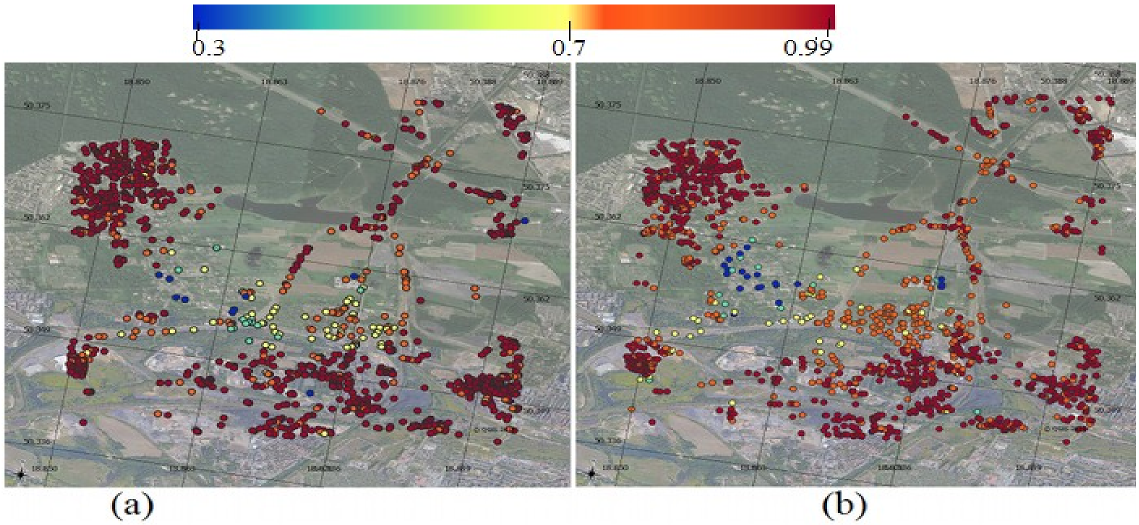


Figure 19. Bytom Temporal Coherence (a) – descending, (b) – ascending (image from QGis).

22 low coherent points were identified by outlier detection from descending track, and 35 from ascending track. Results are displayed in Figure 20. It's visible that for both tracks large proportion of identified low coherent points complement neighbouring high coherent points. In results of both tracks, the displacement is negative, which indicates subsidence of the area.

Although the amount of additionally identified points was not large in this test case, the experiment provides value in localizing and mapping the extent of damages caused by mining in current test region.

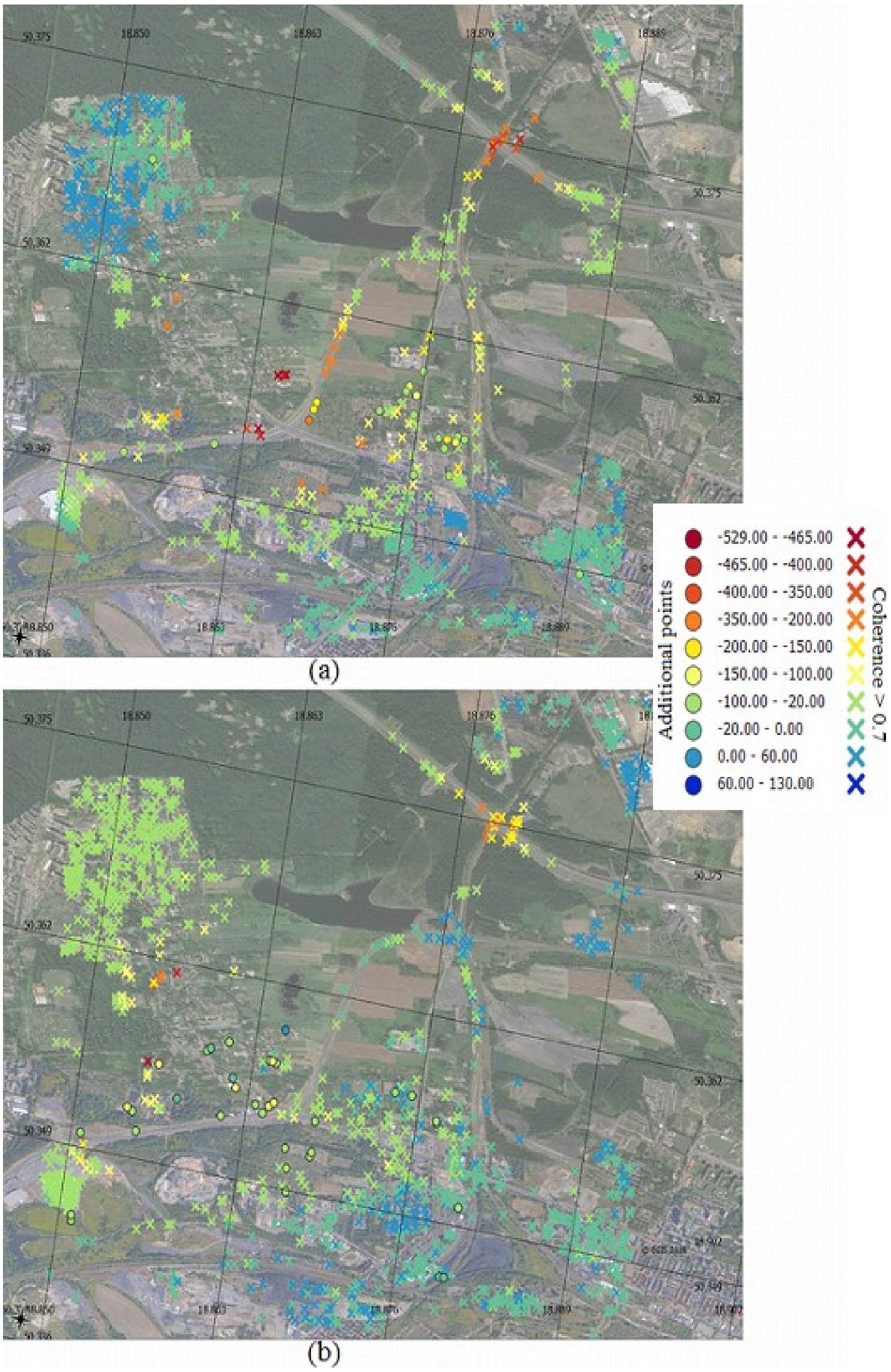


Figure 20. Bytom detected outliers compared to high coherence points. (a) descending (b) ascending (images from QGIS).

7.4 Performance

Performance of the implemented method was measured by executing it on different data sets. Results are displayed in Table 7. Execution time is primarily dependent on the number of processed points. Additionally, the number of clusters identified by clustering and the structure of outlier candidates identified by principal component analysis play important role in execution time of specific data.

Table 7. Comparison of execution time.

Number of points	minPts	cl	Processing time/seconds
56478	5	0.9	144.93
56478	3	0.9	162.77
56478	3	0.85	217.06
25202	5	0.9	29.18
25202	3	0.9	21.45
25202	3	0.85	68.26
4710	5	0.9	8.18
4710	3	0.9	7.35
4710	3	0.85	7.06

8 Conclusions and further development options

The implemented method for multivariate outlier detection was tested on three different regions and on seven different datasets. Results of the execution were validated in context of specifics of areas under investigation. Additionally for two areas, the identified points were compared to interferograms.

The implementation of outlier detection proved to be successful in both goals specified as expected outcome – identifying new areas of interest amongst low coherent points as well as finding additional points to support high coherent points. Application of the method was successful in datasets with both large (Kiruna, Rattlesnake Hills) and very small (Bytom) proportion of low coherent points.

Although the overall results seem to provide value, the author would like to raise one additional issue about how the optimal size of the area under investigation should be determined. As seen from experiments with data of Kiruna, size of selected area (and number of points) affects results quite a lot. Selection of specific area under investigation is definitely something that could be analysed in further work.

Current implementation of outlier detection could be enhanced in future to include analysis of time series. In the applied approach it was already assumed that low coherent points behaving in a similar manner might be of interest. The assumption could be developed even further and applied on time series analysis as well.

9 Summary

The scope of the thesis was to create an outlier detection method on software level for post processing of multi-temporal InSAR results in order to identify multivariate outliers. The implementation is coded in R and the solution greatly based on approach proposed in [1] .

The overall purpose of the implemented outlier detection method was to address the problem of extracting meaningful information from processing of satellite data. The implemented solution was created to benefit to monitoring and observing surface changes as well as estimating the extent of deformations especially in areas which experience non-linear movements, for example due to landslides, earthquakes or underground mining.

The implemented method was tested on three different areas and in total on seven data sets. Test data was provided by AS Datel in CSV format containing results of multi-temporal InSAR processing on data collected by Sentinel-1 satellites. Results of executing outlier detection method were validated based on different studies on each area as well as using interferograms for two areas. It can be concluded that the implementation of the method fulfilled its purpose of detecting additional and relevant points of interest.

Future work could benefit from including time series analysis as well in order to make the outlier detection even more efficient.

References

[1] M. Bakon, I. Oliveira, D. Perissin, J. J. Sousa and J. Papco, "A Data Mining Approach for Multivariate Outlier Detection in Post processing of Multitemporal InSAR Results," in IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 10, no. 6, pp. 2791-2798, June 2017.

[2] “Copernicus”, ESA. [Online]. Available: http://www.esa.int/Our_Activities/Observing_the_Earth/Copernicus/Overview3

[3] “Introducing Sentinel-1”, ESA. [Online]. Available: http://www.esa.int/Our_Activities/Observing_the_Earth/Copernicus/Sentinel-1/Introducing_Sentinel-1

[4] K. Zalite ja K. Voormansik, “Differential and Persistent Scatterer SAR Interferometry” Tartu Observatoorium, 2016

[5] Simons, M., & Rosen, P. A. (2007). Interferometric Synthetic Aperture Radar Geodesy. In Treatise on Geophysics (Vol. 3, pp. 391–446). https://doi.org/10.1016/B978-044452748- 6.00059-6

[6] Mazzanti, P., Perissin, D., & Rocca, A. (2015). “Structural health monitoring of dams by advanced satellite SAR interferometry: investigation of past processes and future monitoring perspectives”. In 7th Internation Conference on Structural Health Monitoring of Intelligent Infrastructure, Torino, Italy.

[7] A. Ferretti, C. Prati and F. Rocca, "Permanent scatterers in SAR interferometry," in IEEE Transactions on Geoscience and Remote Sensing, vol. 39, no. 1, pp. 8-20, Jan 2001.

[8] A. Ferretti, A. Monti-Guarnieri, C. Prati ja F. Rocca, „InSAR Principles: Guidelines for SAR Interferometry Processing and Interpretation“. ESA Publications, 2007.

[9] Hooper, A., Bekaert, D., Spaans, K., & Arikan, M. (2012). Recent advances in SAR interferometry time series analysis for measuring crustal deformation. Tectonophysics. https://doi.org/10.1016/j.tecto.2011.10.013

[10] Hooper, A. (2008). A multi-temporal InSAR method incorporating both persistent scatterer and small baseline approaches. Geophysical Research Letters, 35, 2008 ; doi:10.1029/2008GL034654. 35. 10.1029/2008GL034654.

[11] P. Berardino, G. Fornaro, R. Lanari and E. Sansosti, "A new algorithm for surface deformation monitoring based on small baseline differential SAR interferograms," in IEEE Transactions on Geoscience and Remote Sensing, vol. 40, no. 11, pp. 2375-2383, Nov. 2002. doi: 10.1109/TGRS.2002.803792

[12] Sousa, Joaquim & Lazecky, Milan & Hlavacova, Ivana & Bakon, Matus & Patrício, Glória & Perissin, D. (2015). “Satellite SAR Interferometry for Monitoring Dam Deformations in Portugal.”

[13] MathWorks, Matlab, Available: https://www.mathworks.com/products/matlab.html

[14] Ester, Martin, et al. “A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise.” KDD, 1996, pp. 226–231.

[15] Elbatta, Mohammed & Ashour, Wesam. (2013). A dynamic Method for Discovering Density Varied Clusters. International Journal of Signal Processing, Image Processing and Pattern Recognition. 6. 123-134.

[16] Campello, R. J. G. B.; Moulavi, D.; Sander, J. (2013). Density-Based Clustering Based on Hierarchical Density Estimates. Proceedings of the 17th Pacific-Asia Conference on Knowledge Discovery in Databases, PAKDD 2013, Lecture Notes in Computer Science 7819, p. 160.

[17] Mihael Ankerst, Markus M. Breunig, Hans-Peter Kriegel, Joerg Sander (1999). OPTICS: Ordering Points To Identify the Clustering Structure. ACM SIGMOD international conference on Management of data. ACM Press. pp. 49–60.

[18] I. T. Jolliffe, Principal Component Analysis. Springer New York, 1986.

[19] Einasto M, Liivamägi LJ, Saar E, Einasto J, Tempel E, Tago E, Martínez VJ. SDSS DR7 superclusters-Principal component analysis. Astronomy & Astrophysics. 2011 Nov;535:A36.

[20] Mia Hubert, Peter J Rousseeuw & Karlien Vanden Branden (2005) ROBPCA: “A New Approach to Robust Principal Component Analysis”, Technometrics, 47:1, 64-79, DOI: 10.1198/004017004000000563

[21] Fortune, S., 1995. Voronoi diagrams and Delaunay triangulations. In Computing in Euclidean geometry (pp. 225-265).

[22] Aurenhammer, F., 1991. Voronoi diagrams—a survey of a fundamental geometric data structure. ACM Computing Surveys (CSUR), 23(3), pp.345-405.

[23] Leys, Christophe & Ley, Christophe & Klein, Olivier & Bernard, Philippe & Licata, Laurent. (2013). Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median. Journal of Experimental Social Psychology. 49. 764–766. 10.1016/j.jesp.2013.03.013.

[24] Tan, P.N., Steinbach, M. and Kumar, V., 2005. Introduction to data mining.

[25] A. Ferretti, A. Fumagalli, F. Novali, C. Prati, F. Rocca and A. Rucci, "A New Algorithm for Processing Interferometric Data-Stacks: SqueeSAR," in IEEE Transactions on Geoscience and Remote Sensing, vol. 49, no. 9, pp. 3460-3470, Sept. 2011.

[26] D. Perissin and T. Wang, "Repeat-Pass SAR Interferometry With Partially Coherent Targets," in IEEE Transactions on Geoscience and Remote Sensing, vol. 50, no. 1, pp. 271-280, Jan. 2012.

[27] Raspini, Federico & Bianchini, Silvia & Ciampalini, Andrea & Del Soldato, Matteo & Solari, Lorenzo & Novali, Fabrizio & Del Conte, Sara & Rucci, Alessio & Ferretti, Alessandro & Casagli, Nicola. (2018). Continuous, semi-automatic monitoring of ground deformation using Sentinel-1 satellites. Scientific Reports. 8. 10.1038/s41598-018-25369-w.

[28] CRAN, "deldir", [Online], Available: https://CRAN.R-project.org/package=deldir

[29] CRAN, "dbscan", [Online], Available: https://CRAN.R-project.org/package=dbscan

[30] CRAN, "rrcov", [Online], Available: https://CRAN.R-project.org/package=rrcov

[31] CRAN, "igraph", [Online], Available: https://CRAN.R-project.org/package=igraph

[32] CRAN, "sp", [Online], Available: https://CRAN.R-project.org/package=sp

[33] CRAN, "testthat", [Online], Available: https://CRAN.R-project.org/package=testthat

[34] Spatial Reference, [Online], Available: http://spatialreference.org/ref/epsg/3359/

[35] Copernicus Open Access Hub, [Online], Available: https://scihub.copernicus.eu/dhus/#/home

[36] QGIS, [Online], Available: https://qgis.org/en/site/

[37] Rstudio, [Online], Available: https://www.rstudio.com/

[38] Google Earth, [Online], Available: https://www.google.com/earth/

[39] "Rattlesnake Hills Landslide", Washington State Department of Natural Resources, [Online], Available: https://www.dnr.wa.gov/rattlesnake-hills-landslide

[40] Sentinel Application Platform, [Online], Available: http://step.esa.int/main/toolboxes/snap/

[41] TOPS Interferometry Tutorial, [Online], Available: https://step.esa.int/docs/tutorials/S1TBX%20TOPSAR%20Interferometry%20with%20Sentinel-1%20Tutorial.pdf

[42] SNAPHU: Statistical-Cost, Network-Flow Algorithm for Phase Unwrapping, [Online], Available: https://web.stanford.edu/group/radar/softwareandlinks/sw/snaphu/

[43] Kiruna Kommun, [Online], Available: http://www.kiruna.se/contentassets/b26bffa78a124a8ca4fa6f3f694d0110/kiruna-stadsomvandl-folder-en-webb.pdf

[44] Svartsjaern, M., Saiang, D., Nordlund, E. “Conceptual Numerical Modeling of Large-Scale Footwall Behavior at the Kiirunavaara Mine, and Implications for Deformation Monitoring”, Rock Mechanics and Rock Engineering, (2016) 49: 943. https://doi.org/10.1007/s00603-015-0750-x

[45] Tomás Villegas, Erling Nordlund, Christina Dahnér-Lindqvist, Hangingwall surface subsidence at the Kiirunavaara Mine, Sweden, Engineering Geology, Volume 121, Issues 1–2, 2011, Pages 18-27, ISSN 0013-7952, https://doi.org/10.1016/j.enggeo.2011.04.010.

[46] R. Murdzek, H. Malik, A. Leśniak, “Ground subsidence information as a valuable layer in GIS analysis”, E3S Web of Conf. 36 02006 (2018), DOI: 10.1051/e3sconf/20183602006

[47] Sille Mining, [Online], Available: https://www.sille.space/en/news/analysis-use-cases/31-landing-website/faq/analysis-use-cases/90-mining

[48] Lipecki, Tomasz & Ligarska, Hanna & Zawadzka, Małgorzata. (2018). The influence of mining activities on the Church of St. Cross in Bytom-Miechowice. Reports on Geodesy and Geoinformatics. 105. 7-18. 10.2478/rgg-2018-0002.

[49] R. Murdzek, H. Malik, A. Leśniak, “The use of the DInSAR method in the monitoring of road damage caused by mining activities”, E3S Web of Conf. 36 02005 (2018), DOI: 10.1051/e3sconf/20183602005