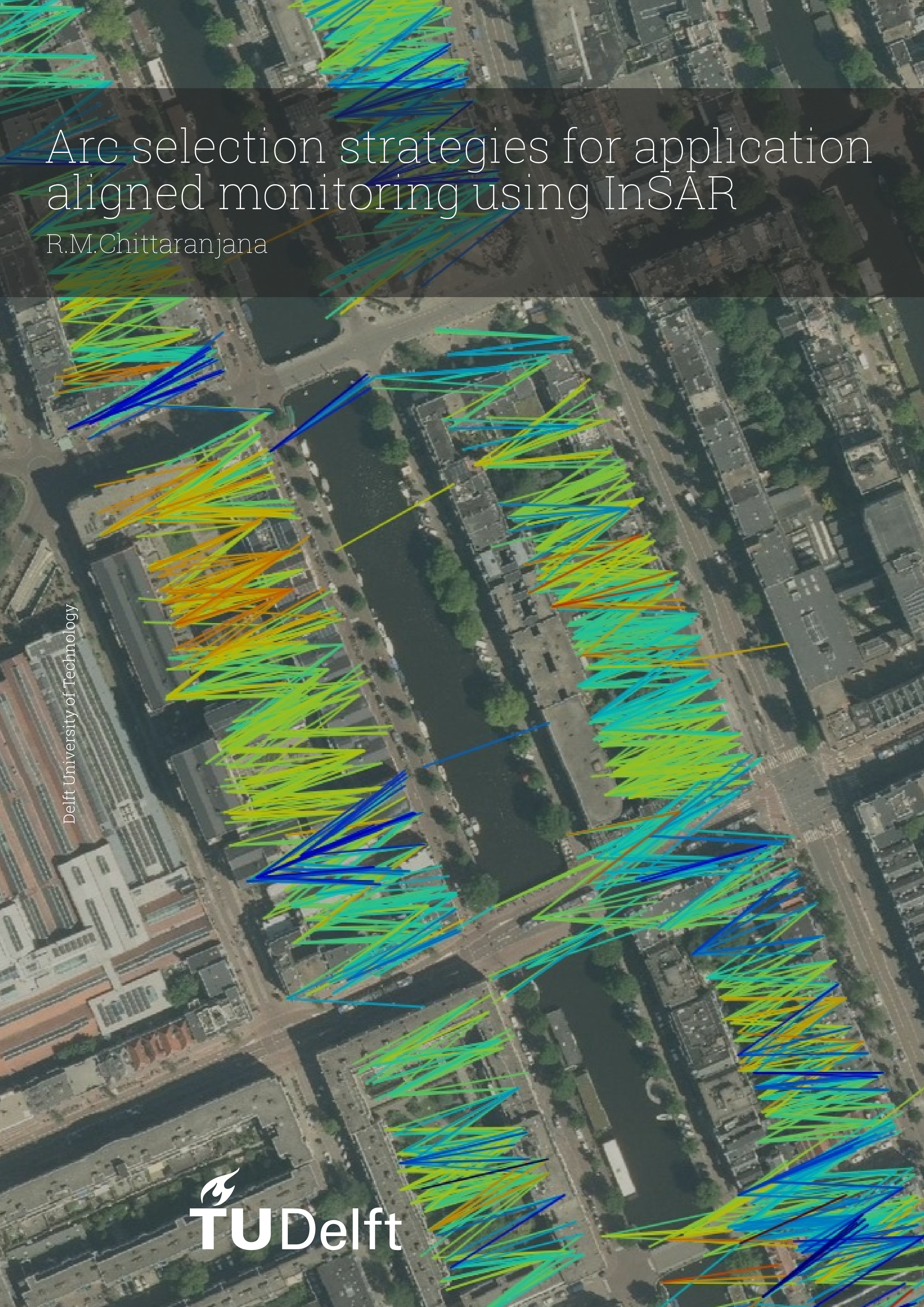
Arc selection strategies for application aligned monitoring using InSAR

R.M.Chittaranjana



Delft University of Technology

Arc selection strategies for application-aligned monitoring using InSAR

by

R.M. Chittaranjana

in partial fulfillment of the requirements for the degree of

Master of Science
in Civil Engineering

at the Delft University of Technology,

to be defended publicly on Wednesday January 31, 2024 at 10:00.

Thesis committee:	Prof. dr. ir. R.F. Hanssen,	TU Delft, Geoscience and Remote Sensing (CiTG)
	Y. Wang, MSc.	TU Delft, Geoscience and Remote Sensing (CiTG)
	Dr. S.R. de Roode.	TU Delft, Geoscience and Remote Sensing (CiTG)

TU Delft

Preface

This thesis is a product of my research over the past year and concludes my journey as a master's student at the TU Delft.

I want to express my gratitude to my supervisors, Ramon Hanssen and Yuqing Wang, who have guided me through this project. Ramon, I appreciate your line of questioning and evaluation of my work during our meetings, which pushed me to be more critical of my work. Yuqing, thank you for always being available; your expertise and constant support were extremely important, without which this project would have been impossible.

I want to thank my parents, brother, family, and friends. Their unwavering support and companionship throughout my master's were instrumental in my success.

R.M. Chittaranjana

Abstract

Interferometric Synthetic Aperture Radar (InSAR) is a geodetic technique that is capable of monitoring surface displacements up to millimeter-level of precision. The end products from conventional InSAR processing are application-agnostic, which means that they are not optimized for any particular application. InSAR products could be more beneficial if tailored for a relevant application, particularly if expert users can tune the products according to their monitoring requirement. Here, we develop tools for application-aligned monitoring by means of the selection of application-relevant arcs between scatterers in InSAR. We are interested in the use of local (short) arcs between point scatterers, as these arcs are more likely to be better suited for monitoring localized differential deformation, and may provide observations of better quality due to the fact that they are less prone to atmospheric noise.

We first compare the time series of local arcs and conventional time series w.r.t. a common reference point based on their deformation behavior. The comparison reveals that the time series of local arcs are capable of providing additional information on deformation behavior over the conventional method. However, the quality of observations in local arcs in general is found to be more variable, and often even worse than those from the conventional method. Most likely, the reason for this is the absence of noise reduction in local arcs in comparison to the time series from the conventional method which optimizes the selection of the common reference point to reduce noise in the time series.

In addition, to optimize the arc selection for a given application, we propose an arc tuning strategy, where criteria can be set based on arc parameters, i.e., the length, the elevation difference (between point scatterers) and the azimuth of the arc. We also introduce the arc clustering method as an exploratory data analysis algorithm for general-purpose monitoring using local arcs. Both of these methods are demonstrated on test scenarios over the quay walls along the canal network of Amsterdam. The demonstration on arc tuning shows that arc setting criteria on arc geometry parameters are adequate to select arcs with certain orientations, and the selection can be further aided by estimating displacement parameters with multiple hypothesis testing. The results from the arc clustering show the potential of detect instability over a certain area using arcs without knowing the motion of the specific object.

This study contributes to monitoring deformation where the InSAR data can be optimally attuned based on a particular application. In order to convey information on selected arcs effectively, a visualization tool based on an interactive map is created in a jupyter notebook environment.

List of Acronyms

2D Two Dimensional

3D Three Dimensional

AA Application-Agnostic

AAA Advanced Application-Aligned

HDBSCAN Hierarchical Density-Based Spatial Clustering of Applications with Noise

InSAR Interferometric Synthetic Aperture Radar

KDTree K-Dimensional Trees

LoS Line of Sight

MDD Minimum Detectable Deformation

MHT Multiple Hypothesis Testing

OMT Overall Model Test

PS Point Scatterer

PSI Persistent Scatterer Interferometry

SAR Synthetic Aperture Radar

SLC Single Look Complex

List of Figures

List of Tables

Contents

1

Introduction

Satellite radar interferometry (InSAR) has been proven to be a powerful technique for deformation monitoring with a broad range of applications. The traditional approach resolves to map wide-scale displacements that are application-agnostic, which means that one does not know a-priori what the InSAR results will be used for. While deformation monitoring using a application-agnostic is feasible, monitoring relative displacements tailored to a well-defined application and to a well-defined end-user will lead to better results. This approach is labeled advanced application-aligned. Here, we aim to study and develop methods for using satellite radar data for advanced application-aligned monitoring.

1.1. InSAR review

Deformation monitoring is a critical task, particularly in the built environment, as any instability or failure that may occur in infrastructure has the potential to pose a threat to human safety. Furthermore, the failure of critical transport infrastructure, such as bridges and tunnels, can have significant socioeconomic consequences.

The problem of accessing and replacing aging infrastructure is prevalent among Western countries and is an expensive task. In the Netherlands, the annual costs of replacing and renovating the existing civil infrastructure are expected to rise from around one billion euros in 2019 to two to three billion in 2030 and around four billion in 2050 [NWO, 2021; TNO, 2021]. A report by the American Society of Civil Engineers on the United States's infrastructure [ASCE, 2021], estimates that the lack of funding can lead to a loss $10 trillion in growth. The report also finds infrastructure maintenance backlogs to be a significant issue and suggests that asset management helps prioritize the limited funding.

Therefore, there is a need to regularly monitor such infrastructure at lower costs. Conventional monitoring techniques for infrastructure usually involve field surveys or the use of in-situ devices, such as accelerometers, laser interferometers, electronic distance measurement instruments and GNSS. Using such instruments is often costly and can only be applied in a limited scale, while their deployment can also be weather dependent. A cost-effective alternative to performing such a task is a specialized technique using satellite remote sensing called Interferometric Synthetic Aperture Radar (InSAR), which can offer a favorable solution by regularly monitoring areas with high precision and being able to look back at its archived observations over the past years.

Radar Remote Sensing

Radar remote sensing involves the use of instruments mounted on satellites or aircraft to gather information about the Earth's surface and atmosphere. Radar remote sensing works by emitting microwave pulses and observing the reflected signals to measure various properties of the Earth's surface. Since radars are active systems, which means they send out signals and do not rely on external sources [Hanssen, 2001], and the radio waves with longer wavelengths can also penetrate through clouds [Skolnik, 1980], they have the benefits of being able to operate regardless weather conditions or the availability of daylight.

Synthetic Aperture Radar and Interferometry

Synthetic Aperture Radar (SAR) is a specialized imaging technique used in radar remote sensing to generate high-resolution images of the Earth's surface. SAR systems work by simulating a large antenna aperture virtually by combining signals along the motion of the platform carrying the radar instrument. This process generates a two-dimensional complex image for each single radar pass over a specific area on the Earth's surface; the data generated of this form is known as a Single Look Complex (SLC) image. The SLC data contains both amplitude and phase information. A complex phasor P denotes the value of each pixel in a SLC image [Hanssen, 2001] and is represented as

$$P = A \exp i\psi, \tag{1.1}$$

where A represents the amplitude of the backscattered signal and ψ represents its fractional phase.

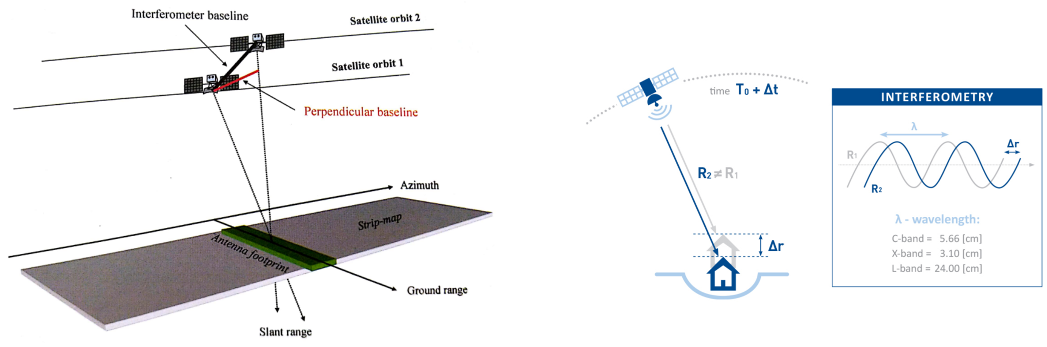
Interferometric Synthetic Aperture Radar

In order to obtain three-dimensional information, two SAR images are needed; these can be two images taken at the same time from different orbital tracks or two images taken at two different times. One of the SAR images is used as a reference and, therefore called the mother image, and the other is called a daughter image as it aligned to the grid of the mother and is subsequently resampled. Then, an interference image can be created by taking the two SAR images and calculating the phase difference; this can be implemented by taking the pixel-wise multiplication of the mother image with the complex conjugate of the daughter image. The resulting phase difference is called the interferometric phase, which can be utilized to measure the elevation or the displacements.

For a mother and a daughter image with complex values, the interferogram can be represented as

$$I_{md} = P_m P_d^* = A_m A_d \exp(i(\psi_m - \psi_d)), \tag{1.2}$$

here P_m and P_d represent the complex phasor values of mother and daughter images, A_m and A_d represent the amplitudes of the mother and daughter images, and ψ_m and ψ_d represent the phase values in the mother and daughter images [Van Leijen, 2014]. The asterisk (*) indicates the complex conjugate.



(a) (b)

Figure 1.1.: (a) Geometry of a Interferometric SAR system. Source: [Ferretti et al., 2007]. (b) Phase difference observed using InSAR from repeat observations before and after the motion. Source: [TRE, 2023].

The interferometric phase between an arbitrary pixel in two SAR images is given by

$$\phi_{md} = \psi_m - \psi_d. \tag{1.3}$$

These interferometric phases not only contain information resulting from surface displacement but are a combination of other factors that affect the phase of a signal, thus the interferometric phase can be represented as

$$\phi_{md} = -2\pi a + \phi_{\text{flat}} + \phi_{\text{topo}} + \phi_{\text{displ}} + \phi_{\text{atmo}} + \phi_{\text{scat}} + \phi_{\text{noise}}, \tag{1.4}$$

where a represents the integer number of cycles of the signal along the distance to and back from the surface. The following term ϕ_{flat} represents the flat earth phase and results from the assumption of the reference shape of the earth. The topographic phase ϕ_{topo} arises from the elevation of the terrain above the reference shape. The displacement phase ϕ_{defo} is a result of the contribution from surface displacements, and it is what we focus on in this study and is further elaborated in the next subsection. The atmospheric phase ϕ_{atmo} results from the different atmospheric states between the mother and daughter acquisitions. The terms ϕ_{scat} and ϕ_{noise} are a result of change in scatter characteristics of an observed area between acquisitions, together with thermal and processing noise [Van Leijen, 2014].

Displacement phase

The displacement phase ϕ_{displ}, is a result of the phase contribution from surface displacements. The unit of the phase contribution is in radians while displacements are measured in meters, the equation for unit conversion is given by Van Leijen [2014], as

$$\phi_{\text{displ}} = -\frac{4\pi}{\lambda} d_{\text{LoS}} \,. \tag{1.5}$$

The term d_{LoS} represents the displacement along the satellite's line of sight (LoS) direction, and λ is the signal's wavelength. Although the deformation is projected onto the LoS direction of the signal, the contributing phenomena are three-dimensional. In this study, we are interested in detecting the displacement component in the line of sight direction.

InSAR processing is not discussed in detail in this study, for detailed information about the processing the works done by Hanssen [2001]; Van Leijen [2014] can be refereed.

Persistent Scatterer Interferometry

In order to overcome the spatial and temporal decorrelation and atmospheric error signal present in conventional InSAR and effectively exploit multi-temporal InSAR, further processing techniques such as Persistent Scatterer Interferometry (PSI) were developed, first reported by Ferretti et al. [2000]. PSI exploits coherent point scatterers (PS), which are objects showing strong reflectance that can be regarded as a single point. Since most man-made structures, such as buildings, show such desired scattering characteristics, PSI is highly suited and well-used in urban areas. Although many PSI algorithms exist [Crosetto et al., 2016], here we focus on the method for PSI parameter estimation of Van Leijen [2014].

The processing starts with interferogram generation from a stack of images; for this, a mother image is selected based on the criteria of maximizing the stack coherence [Kampes, 2005]. The next step in the PSI process is the selection of point scatterer candidates. This is usually done based on metrics that can be calculated from the amplitude information of scatterers, which acts as a proxy for phase stability. Pixels that satisfy a particular criterion or threshold are selected as point scatterer candidates. Out of the selected PS candidates, a select set that are distributed homogeneously across the area, are designated as first-order PS candidates. These first-order PS candidates are then connected with each other to form a reference network, using algorithms like Deluaney triangulation [Van Leijen, 2014]. An 'arc' is defined as the spatial phase difference between two connecting PS. After the formation of the network, the relative phase observations for each arc (between 1^{st} order PS candidates) are calculated, and the phase ambiguities are resolved together with the estimation of the parameters of interest (relative deformation, relative height difference). The parameters are then integrated in space with a common reference point in order to obtain absolute values. Assuming correct estimation of the topography and deformation parameters, the atmospheric contribution, along with unmodeled deformation, orbit errors, and noise, will be included in the residuals between observation and estimated phase. Filtering is done in time and space to separate the atmospheric contribution from the unmodeled deformation. A second-order network is then used to densify the network of PS. The parameters for the second-order network are calculated relative to the closest first-order PS.

1.2. Problem statement

Conventional InSAR processing leads to end-products that are application-agnostic (AA), meaning that they are not optimized for a particular application and thus serve for general-purpose use, like technology demonstrators.

However, when a specific problem or application needs to be addressed, InSAR products tailored to that application may be more appropriate. These products are known as advanced application-aligned (AAA) products. For instance, if a particular building needs to be monitored, since InSAR measure relative displacements, using an application-agnostic product might result in displacement measurements relative to an arbitrary location, making it harder to interpret. However, if InSAR estimates can be optimized to obtain relative to a local or known location whose stability is known or can be presumed; it will lead to a better interpretation of the estimates. This interpretation and analysis of a numerical AAA product requires a set of tools, that can be used to focus on particular aspects of the data. Currently, such tools are largely non-existing.

A way to enable the creation of such advanced application-aligned products is based on the generation or selection of application-relevant arcs between PS points. Arc selection methods currently used as conventional InSAR processing, e.g., Delaunay triangulation, are more focused on generating arc networks for estimation and removal of error sources and, hence, less suited for selecting application-relevant arcs [Van Leijen, 2014]. Thus, we require methods of selecting arcs for application-aligned monitoring.

In regard to application-relevant arcs for monitoring. Local arcs, which are arcs of short length, may be better at monitoring localized displacements and may be averse to atmospheric noise as atmospheric delay for two points of the arc would be similar. Thus, we would expect the measurements from such arc to be of better quality over measurements from conventional methods. The validity of these expectations need to be investigated.

1.3. Research objective and questions

The main objective of this study is to develop and demonstrate methodology for application-aligned monitoring using InSAR . To this end, we formulate the main research question:

How can application-aligned monitoring and interpretation be enabled in the context of satellite radar interferometry?

We seek to enable such monitoring approach based on the selection of relevant arcs. To do this, first, we investigate the benefits of using local arcs over conventional methods. We then develop strategies and tools that can be used to optimize the arcs based on application requirements. These are translated into the following sub-questions:

- *How do localized arcs compare to conventional time series with respect to a common reference point for deformation monitoring?*

  In this first research question, we aim compare the displacement time series obtained from local arcs with conventional time series w.r.t. a common reference point for the purpose of displacement monitoring. The question is aimed at evaluating the benefits of using local arcs over conventional methods with single global reference point.

- *How can user-initiated arc selection be implemented for given case of deformation monitoring?*

  Currently used arc selection methods are both autonomous and application-agnostic. Thus there is a need for new methods where relevant arcs can be selected by an expert user pertaining to a monitoring case.

In addition to these research questions, we look at methods to communicate deformation information from these selected arcs effectively. While traditionally, static deformation maps are often used for this purpose, when considering arcs, the selected arcs may often cross and overlap each other, thus becoming more challenging to interpret. In such cases, an interactive visualization of arcs over a map would be more beneficial.

1.4. Limitations

In order to focus only the posed objective we provided ourselves with some limitations.

For the displacement time series of arcs in this study, we ignore any instances of phase unwrapping error that may be present. Regardless of this, for most plots containing displacement time series in this study, the ambiguity is plotted for visual analysis.

For the position and elevation of the PS we use the estimates provided along with the data. Although, accurate information on the 3D location of PS is critical for proper interpretation and literature exists on to do 3D geolocation [van Natijne, 2018], we do not apply this.

We also limit ourselves to the line-of-sight displacements of the arcs and do not go into resolving and analyzing the displacements in 3D.

1.5. Outline

Chapter 2 comprises the methodologies that were developed and used in this study. In Chapter 3, we first evaluate the time series of local arcs against the conventional time series with respect to a common reference point and then demonstrate the application of the two arc selection methods. Finally, in Chapter 4, we have conclusions of this study and recommendations for further research.

2
Methodology

This chapter covers the methodologies used and developed as part of this study. First, we define an arc and its parametrization in space in section 2.1. Subsequently we describe the local arc connection strategy in 2.2. In section 2.3, we describe the methodology used for deformation modeling and estimating displacement parameters. We then present two strategies for arc selection, the arc tuning strategy and the arc clustering strategy, in sections 2.4 and 2.5.

2.1. InSAR arcs

In InSAR, the connection between two vertices, which are coherent scatterers, is referred to as an arc. InSAR measurements are by nature are double-differenced, across space and time, the arc indicates the spatial difference between two scatterers.

An arc can be described in three-dimensional (3D) space using azimuth (θ), elevation difference (e), arc length (d), and position of the (x, y, z) of the points of the arc . We use a tuple to represent an arc, where a tuple is an ordered collection of objects; for a tuple (i, j) representing an arc, its objects i and j represent the two-point scatterers of an arc. The order of the two-point scatterers in a tuple is relevant because an arc represented by tuple (i, j) represents the dynamic behavior of point I relative to j while a tuple (j, i) represents the dynamic behavior of point j relative to i. Figure 2.1 shows the representation of the tuple (i, j) along with its geometry parameters in 3D space.

2.1.1. Arc characteristics

The coordinates of the two point scatterers, i and j, in Fig. 2.1 are denoted by (x_i, y_i, z_i) and (x_j, y_j, z_j), where x and y denote the east and north coordinates of the point and z represents the estimated height of the PS. For this project the Rijksdriehoek coordinate reference system (EPSG: 28992), which is the local Dutch reference system is used, thus all units in meters.

The arc length (d_{ij}) for an arc (i, j) is the distance between the two points forming an arc, it is given by

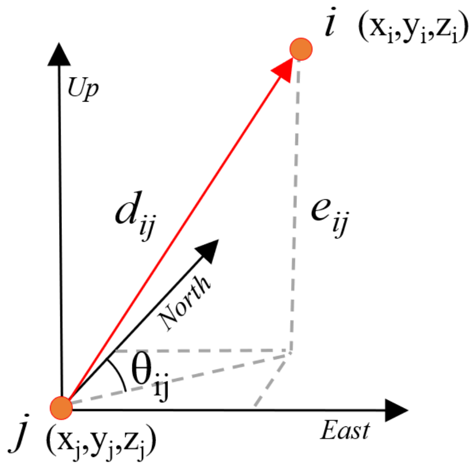


Figure 2.1.: An arc (i, j) between two point scatterers i and j, represented in 3D space by azimuth (θ_{ij}), elevation difference (e_{ij}) and arc length (d_{ij}).

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + (z_i - z_j)^2}. \tag{2.1}$$

The difference in elevation between the two points of an arc (i, j) is given by e_{ij}, calculated by

$$e_{ij} = z_i - z_j. \tag{2.2}$$

The azimuth (θ_{ij}) of an arc (i, j) is the horizontal angle made by the arc relative to the north direction and is calculated by

$$\theta_{ij} = \text{atan2}(\frac{(x_i - x_j)}{(y_i - y_j)}). \tag{2.3}$$

The arctan2 is an extension of the arctan function. It takes into account the signs of both the numerator and the denominator to determine the correct quadrant of the angle, giving the output in the interval $(-\pi, \pi]$, whereas the regular arctan function gives an output in the interval $(-\pi/2, \pi/2)$. We further convert the azimuth from radians to degrees with the interval now being $(-180°, 180°]$.

For an arc, if its azimuth is positive, the arc is taken to represent the motion of the eastern-most point w.r.t. the western-most point, similarly an arc with negative azimuth is taken to represent the motion of the western-most point w.r.t. the eastern-most point. Fig. 2.2 shows an example visualization of two such arcs (i, j) and (k, j). Here, arc (i, j) has a positive azimuth θ_{ij} thereby representing the motion of point i w.r.t. to point j, similarly, the arc (k, j) with negative azimuth θ_{kj} represents the motion of point k w.r.t. to point j. For simplicity, the visualization is done in the north-east plane.

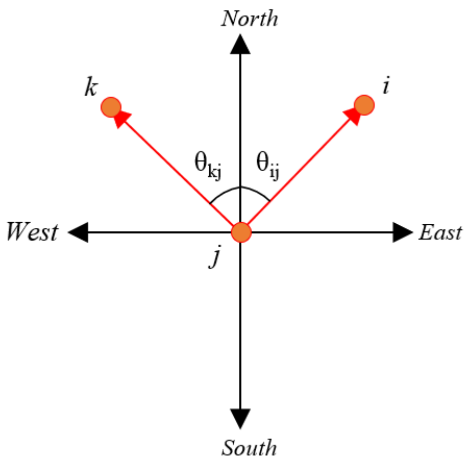


Figure 2.2.: Two arcs (i, j) and (k, j) with a positive and negative azimuth respectively. Arc (i, j) with positive azimuth θ_{ij} represents motion of point i w.r.t. to point j and arc (k, j) with negative azimuth θ_{kj} represents the motion of point k w.r.t. to point j. For brevity only the projection of arcs in the North-East plane is shown.

2.1.2. Displacement time series

The time series of an arc is obtained by subtracting displacement time series of the two points with respect to a common reference point.

$$y_{ij} = y_i - y_j, \tag{2.4}$$

where the vectors y_i and y_j represent the displacement time series w.r.t. common reference point of the points i and j respectively, and vector y_{ij} represents the displacement time series of the arc (i, j), i.e., the displacement of point i w.r.t. j.

For an arc (i, j), we decided to use the position of the point i to denote the position of the arc. This is done because when we calculate the displacement time series of an arc (i, j), based on Eq. (2.4), it is taken to represent the motion of point i with respect to point j, therefore we feel that in order to represent a position of an arc, it is better to represent it as the position of i, the point whose dynamic behavior we seek to describe.

2.2. Arc connection

A simple method for connecting all possible arcs in a dataset of point scatterers would be to loop over every point in a dataset and create all possible tuples with all other points. This method is inefficient when scaled to a large number of points. It is also unnecessary to generate all possible arcs between points in a dataset; rather, it might be more efficient to generate arcs that may be of relevance. For example, for monitoring localized deformation, we might only be interested in arcs of short length.

In order to efficiently find and connect relevant arcs for a large amount of PS, we use a KD-Tree data structure introduced by Maneewongvatana and Mount [1999]. A KD-tree, short for a k-dimensional tree, is a data structure used for organizing points in a k-dimensional space. In this project the data structure is used in 3D space, since PS points and arcs exist in 3D space. This type of data structure is particularly efficient at nearest neighbor searches. The KD-Tree data structure and its affinity for nearest neighbor searches is relevant to us because when we store the location of PS in the KD-Tree format, the optimal nearest neighbor search allows us efficiently find pairs of points that are within a specified length threshold. Thus, these pairs of points can be turned to tuples in order to represent the generated arcs.

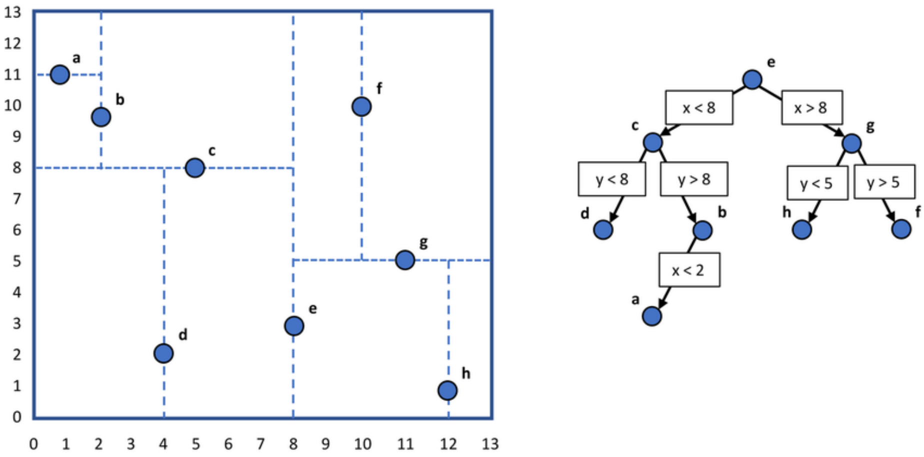


Figure 2.3.: An example of points in space (left) being converted into a KD-Tree structure (right), the graph on the right shows points in 2-Dimensions being partitioned and arranged on the right in a hierarchical tree like format. Figure obtained from [Ullrich and Zarzycki, 2017].

In the KD-Tree method, points in space are recursively partitioned into hyper-rectangles, and the points are arranged in a hierarchical tree-like structure. Fig. 2.3 shows an example of how kd-tree is built in 2D space; on the left, there are eight points labeled from a to h. The tree structure is built by recursively partitioning the space based on the median point on an axis and cycling through the axis. We start with the x-axis; here, point e is found to have the median value among all points in the x-axis; hence, the space is partitioned along point e. The newly partitioned spaces are further partitioned based on the exact procedure but now along the y-axis; this leads to partitioning along point c and point g, which are the median points in the y-axis in their respective spaces. This procedure is followed through until all points are partitioned. Thus, The partitioning results in a tree-like structure shown in Fig. 2.3 (right). The hierarchical tree-like structure makes operations like nearest neighbor searches much faster, as one can narrow down the scope of the search for each point to the closest branches of the tree.

The implementation of this was done using the *scipy.spatial* module in Python, here the *cKDTree* function is used to form the data structure, and the *query_pairs* function is used to obtain the (arc) tuples when an maximum (arc) length is specified. The algorithms used are based on work by Maneewongvatana and Mount [1999].

This procedure can be performed in all PS in a given dataset. However, suppose there is a specific area of interest for an application. In that case, only the PS within that area can be selected, and only arcs among those selected PS points inside the area of interest can be generated.

2.3. Displacement time series modelling

The temporal behavior of a displacement time series in InSAR is generally represented as a linear function of time with a certain velocity. Although this might be well suited in some instances, time series may show non-linear behavior. We use the probabilistic method of InSAR time-series modeling by Chang and Hanssen [2016] to estimate each time series's behavior and displacement parameters. We use some of the most common deformation models to represent the time series behavior. We build a library of deformation models, with which we can build a hypothesis that could better represent the time series behavior.

2.3.1. Library of canonical deformation models

The library of functions used in this project is limited to linear, constant, periodic, step model and break point model. These functions are given as

$$\begin{aligned} M_1(v) &= t \cdot v + a, \\ M_2(s,c) &= sin(2\pi t) \cdot s + (cos(2\pi t) - 1) \cdot c, \\ M_3(\Delta_i) &= \Delta_i \cdot \mathscr{H}(t - \tau_i), \quad i \in [1, m-1], \\ M_4(v_b) &= t^i \cdot v_b, \quad i \in [b+1, m-1]. \end{aligned} \tag{2.5}$$

here

- M_1 represents the linear model, with temporal baseline t. Here the parameters are linear velocity v and initial displacement a.
- M_2 represents a periodic model , where s and c represent coefficients of periodic displacement.
- M_3 is the step model where $\mathscr{H}(t - \tau_i)$ is the Heaviside step function centered at τ_i, with offset Δ_i.
- M_4 represents the change in velocity v_b after an epoch b, M_5 in combination with M_1 forms a breakpoint model with a breakpoint at epoch b.

2.3.2. Multiple hypothesis testing

The null hypotheses (H_0) and the alternative hypotheses (H_j,$\forall j$) can be expressed as a set of equations, given by

$$\begin{aligned}
H_0: \quad & E\{\underset{m\times 1}{\underline{y}}\} = \underset{m\times n}{A}\ \underset{n\times 1}{x}\ ; \\
& D\{\underline{y}\} = \underset{m\times m}{Q_{yy}} = \sigma^2 R_{yy}, \\
H_j: \quad & E\{\underset{m\times 1}{\underline{y}}\} = \underset{m\times n}{A}\ \underset{n\times 1}{x} + \underset{m\times 1}{C_j}\ \underset{q\times 1}{\nabla_j},\ \underset{q\times 1}{\nabla_j} \neq 0; \\
& D\{\underline{y}\} = \underset{m\times m}{Q_{yy}} = \sigma^2 R_{yy},
\end{aligned} \tag{2.6}$$

here $E\{.\}$ is the expectation operator where A is the design matrix and x is the vector of n unknown parameters. $D\{.\}$ is the dispersion operator where Q_{yy} is the covariance matrix of the m observations given by its unit variance σ^2 and co-factor matrix R_{yy}. For the alternate hypothesis H_j, C_j represents the design matrix for the vector of additional unknown parameters ∇_j of the dimension q, for $q \in [1, m-n]$.

The functional models for the hypothesis are built using the models specified in Eq. (2.5), the hypotheses used in this project are

$$\begin{aligned}
& H_0 : M_1(v, a), \\
& H_1 : M_1(v, a) + M_2(\eta), \\
& H_2 : M_1(v, a) + M_2(\eta) + M_4(\Delta_i), \\
& H_3 : M_1(v, a) + M_3(\Delta_i), \\
& H_4 : M_1(v, a) + M_4(v_b).
\end{aligned} \tag{2.7}$$

For the hypothesis that uses the either step or break-point models, prior knowledge on time of occurrence is necessary. In practice such information is not always available, as is the case in this project, hence we evaluate the hypothesis with these models for every time step of the observations. Thus, the total number of alternative models is far more than the five mentioned in Eq. (2.7).

For testing the alternate hypothesis, we use a test statistic $\underline{T}_q$, which follows a Chi-square distribution $\chi^2(q, \lambda)$, where λ is the level of non-centrality. The test statistic is defined as [Teunissen et al., 2004]

$$\begin{aligned}
\underline{T}_q^j &= \underline{\hat{e}}_0^T Q_{yy}^{-1} \underline{\hat{e}}_0 - \underline{\hat{e}}_j^T Q_{yy}^{-1} \underline{\hat{e}}_j \\
&= \underline{\hat{e}}_0^T Q_{yy}^{-1} C_j (C_j^T Q_{yy}^{-1} Q_{\hat{e}_0\hat{e}_0} Q_{yy}^{-1} C_j) C_j^T Q_{yy}^{-1} \underline{\hat{e}}_0,
\end{aligned} \tag{2.8}$$

where the vector $\underline{\hat{e}}_0$ is the residual between the functional model and the observations under the null hypothesis H_0, $\underline{\hat{e}}_j$ is the vector of residuals between the functional model and the observations under the alternate hypotheses H_j and $Q_{\hat{e}_0\hat{e}_0}$ is the covariance matrix of residuals, given by $Q_{\hat{e}_0\hat{e}_0} = Q_{yy} - A(A^T Q_{yy}^{-1} A)^{-1} A^T$. For an alternate hypothesis H_j, the null hypothesis is rejected whenever $\underline{T}_q > \chi^2_\alpha(q)$ where $\chi^2_\alpha(q)$ is the predefined critical value for a significance level of α.

A special case occurs when $q = m - n$, resulting in no redundancy in system, thus $\underline{\hat{e}}_j = 0$ and Eq. (2.8) becomes

$$\underline{T}_{q=m-n} = \underline{\hat{e}}_0 Q_{yy}^{-1} \underline{\hat{e}}_0, \tag{2.9}$$

which is used to access the general correctness of the null hypothesis, this is known as Overall Model Test (OMT) . Whenever $\underline{T}_{q=m-n}$ is less than critical value K_α, the null hypothesis is sustained and no further hypothesis testing is done.

In case the null hypothesis is rejected in the OMT, we need to test for alternate hypotheses and also find the optimal alternative hypothesis among them. Since the alternate hypotheses have different dimensions q, their test statistics $\underline{T}_q^j$ have different χ^2 distributions and can't be compared with each other. Thus, we normalize the test statistic by dividing it with its critical value $\chi^2_{\alpha_j}(q_j)$ for a given significance level of α_j. The test ratio is given as

$$\underline{\mathbf{T}}_{q_j}^j = \underline{T}_{q_j}^j / \chi^2_{\alpha_j}(q_j). \tag{2.10}$$

The power of a test γ gives the probability of correctly rejecting the null hypothesis. For multiple alternate hypothesis, we need to make sure that the power of a test is identical in for all alternate hypotheses [Chang and Hanssen, 2016]. To do this we use the B-method of testing [Baarda, 1968], where we can fix a reference power of test γ_0 and then calculate the reference non-centrality parameter λ_0 for a given level if significance α_j and dimension q_j, given as

$$\lambda_0 = \lambda(\gamma_0, \alpha_j, q_j). \tag{2.11}$$

Constant values of non-centrality parameter λ_0 and power γ_0 assure that each alternate hypothesis is selected with equal probability [Chang and Hanssen, 2016], regardless of dimension q_j. For application in this project, we first calculate λ_0 with $q = 1$, power of test $\gamma_0 = 50\%$ and a significance level of $\alpha_0 = 1/2m$ as used by Chang and Hanssen [2016]. The then obtained reference values of λ_0 and γ_0 are used to to calculate the significance level α_j for dimension q_j, which in turn is used to determine the critical value $\chi^2_{\alpha_j}(q_j)$, thereby ensuring equal probability of selection of all alternate hypothesis. The most suitable alternate hypothesis is considered to be the one that strongly rejects the null hypothesis, thus optimal alternate alternate hypothesis is given where $\underline{\mathbf{T}}_{q_B}^B > 1$ and

$$\underline{\mathbf{T}}_{q_B}^B = \max_j \{\underline{\mathbf{T}}_{q_j}^j\}. \tag{2.12}$$

In order to validate the results obtained from MHT metrics such as posterior variance factor F, precision of the parameters $Q_{\hat{x}\hat{x}}$ and eigenvalue ζ are used by Chang and Hanssen [2016]. Here we only use and describe the variance factor. The posterior variance factor is given by

$$F = \frac{\underline{\hat{e}} Q_{yy}^{-1} \underline{\hat{e}}}{m - (n + q_j)}, \tag{2.13}$$

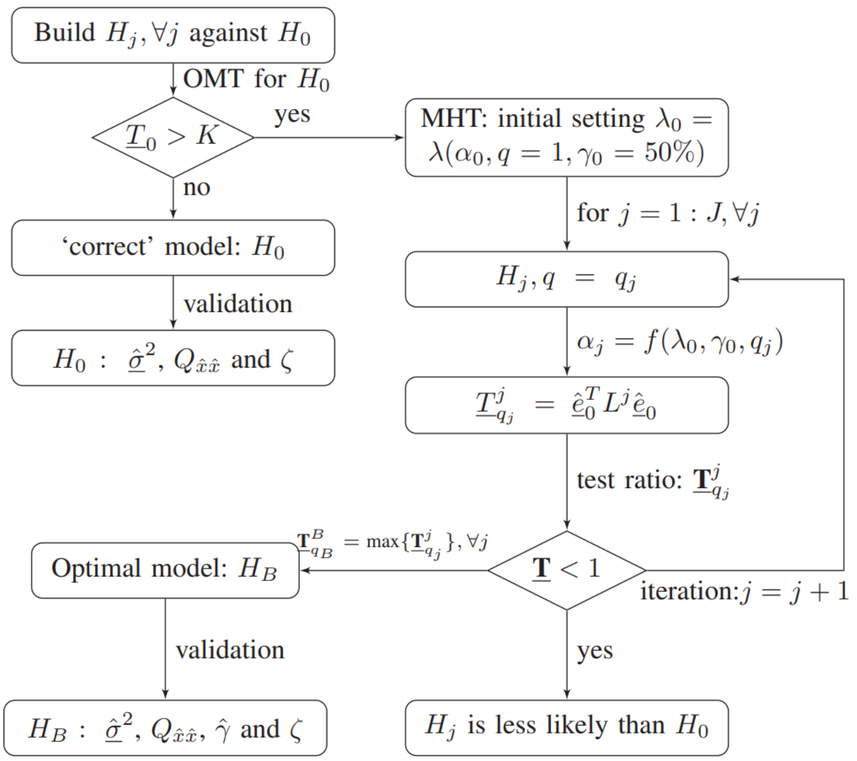

Build $H_j, \forall j$ against H_0
OMT for H_0
$\underline{T}_0 > K$
yes
MHT: initial setting $\lambda_0 = \lambda(\alpha_0, q = 1, \gamma_0 = 50\%)$
no
'correct' model: H_0
validation
$H_0 \ : \ \hat{\underline{\sigma}}^2$, $Q_{\hat{x}\hat{x}}$ and ζ
for $j = 1 : J, \forall j$
$H_j, q \ = \ q_j$
$\alpha_j = f(\lambda_0, \gamma_0, q_j)$
$\underline{T}^j_{q_j} \ = \ \hat{\underline{e}}_0^T L^j \hat{\underline{e}}_0$
test ratio: $\underline{\mathbf{T}}^j_{q_j}$
$\underline{\mathbf{T}}^B_{q_B} = \max\{\underline{\mathbf{T}}^j_{q_j}\}, \forall j$
Optimal model: H_B
$\underline{\mathbf{T}} < 1$
iteration:$j = j + 1$
validation
yes
$H_B \ : \ \hat{\underline{\sigma}}^2$, $Q_{\hat{x}\hat{x}}$, $\hat{\gamma}$ and ζ
H_j is less likely than H_0

Figure 2.4.: Flow chart for deformation modelling procedure. Chart obtained from Chang and Hanssen [2016].

here $\underline{\hat{e}}$ is the vector of residuals between the functional model and the observations under the optimal hypothesis. The denominator in Eq. (2.13) represents the redundancy in the system. When calculating the posterior variance factor for time series that sustained null hypothesis, $q_j = 0$.

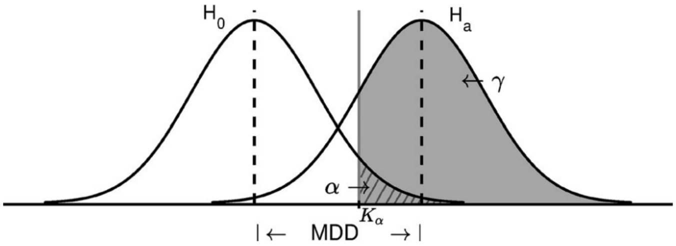


Figure 2.5.: Probability density functions under the null hypothesis H_0 and alternative H_a hypothesis. The MDD is the distance between the main modes of both PDFs, and follows from the desired detectability power γ and confidence level $1-\alpha$. The critical value is denoted by K_α [Chang et al., 2018] .

Minimum detectable deformation (MDD) is another metric that is important in statistical hypothesis testing, it gives the minimum value that an additional displacement parameter ∇_j should have, in the functional model of a specific alternate hypotheses, to be detectable with a power, γ_0 [Chang and Hanssen, 2016]. The MDD is important because, for a given significance level and a predefined functional and stochastic model, it is linked to the choice of power of test. For example, choosing large discriminatory power would imply that the additional parameter ∇_j would need to be larger to be detected.

The MDD has a proportional relationship with the dispersion (stochastic model). This means that for noisier measurements where the the defined dispersion would be larger, the MDD will also be larger compared to measurements with lower dispersion, for a given α_0 and γ_0.

2.4. Arc tuning

As discussed in Section 1.2, when using conventional InSAR processing methods like PSI, the resulting output is a set of displacement time series estimates for point scatterers w.r.t a common reference point. However, this output is application-agnostic, i.e., not optimized for any specific application or use case. Moreover, the estimates are relative to an arbitrary reference point that may be selected automatically during processing. The stability of this reference point is usually unknown, but it is ideally located somewhere unrelated to an application's area of interest. As a result, interpreting the measurements can be challenging.

Given a monitoring case, an InSAR strategy that is application-aligned (i.e., optimized of the use case) will lead to estimates that are better interpretable as they can be based on selection of application-relevant arcs between point scatterers. For example, given the problem of structural health monitoring of a specific building, we can select arcs formed between points over the building and points over an adjacent road whose stability might be known; moreover, if the arc selection method is readily adaptable, multiple results can be generated from the same case, like arcs between the building in question and multiple different surrounding buildings. Having different measurements for one application will also aid in the validation of observed behaviors.

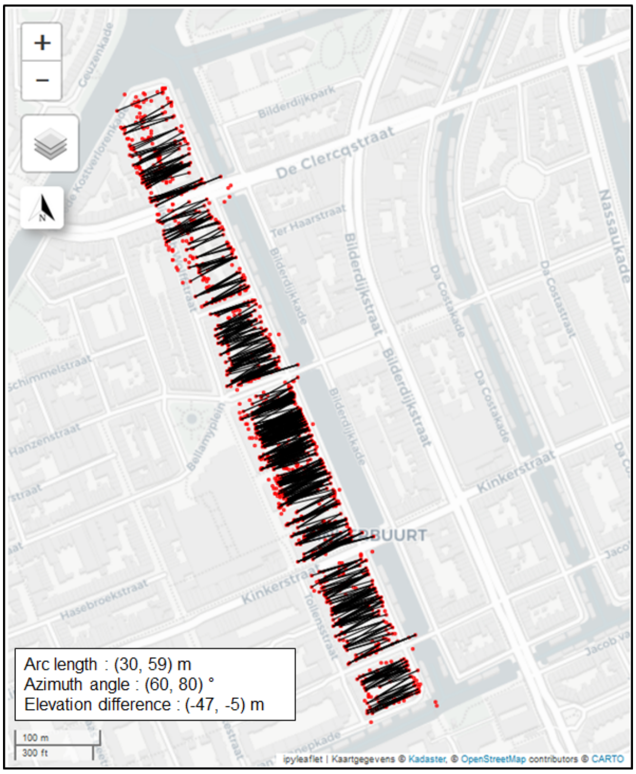


Figure 2.6.: An example of the arc-tuning approach, the red dots represent the point scatterers and the black lines represent the arcs. In this case, we have chosen the arcs based on criteria of arc length (30, 59)m, azimuth angle (60, 80)° and elevation difference (-47,-5)m to select arcs formed between buildings and canals that are orthogonal to the canal.

Such arc selection can be done by setting criteria based on application on parameters of arc geometry like arc length, elevation difference and azimuth and relevant arcs are selected based on the ones that match the set criteria. Since the criteria do not need to be definitive, they can be tuned until necessary or adjusted for different use-cases. This process is called arc-tuning.

Fig. 2.6 shows the arc selection done using arc tuning for an example problem. In this example problem, we have the task of monitoring the motion of the quay wall/road that is along the canal. Suppose the decision was taken to monitor the motion of the quay wall relative to the adjacent building; any relevant arc for this purpose can be selected by setting criteria on different arc geometry parameters. For elevation difference, here, the criteria were set that the difference has to be less than -5 meters, which ensures that there is a substantial elevation difference in the arc, which would be expected if one of the points of an arc was on a building and the other on a quay wall. Here, the elevation difference is negative due to how the arc was taken; in a scenario from an opposite viewpoint where we need to monitor the building relative to the quay wall or road, the criteria for elevation difference would have been positive, e.g., greater than 5 meters. The choice of the azimuth to be between 60° and

80° was done so the motion is obtained relative to the building to the left of the quay wall. The choice of criteria for arc length to be between 30 and 59 meters here was arbitrary, but it can be used to ensure sufficient distance between the points of an arc.

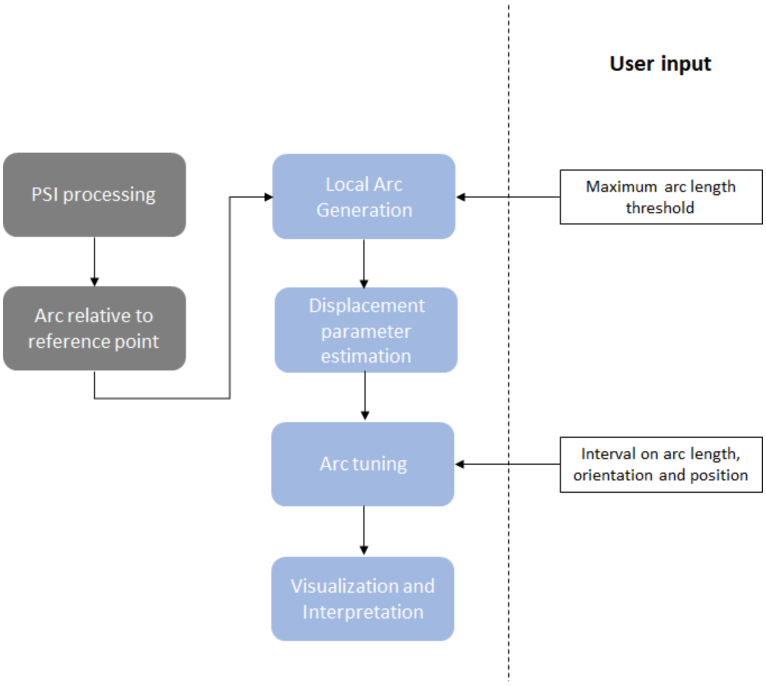

User input
PSI processing
Arc relative to reference point
Local Arc Generation
Displacement parameter estimation
Arc tuning
Visualization and Interpretation
Maximum arc length threshold
Interval on arc length, orientation and position

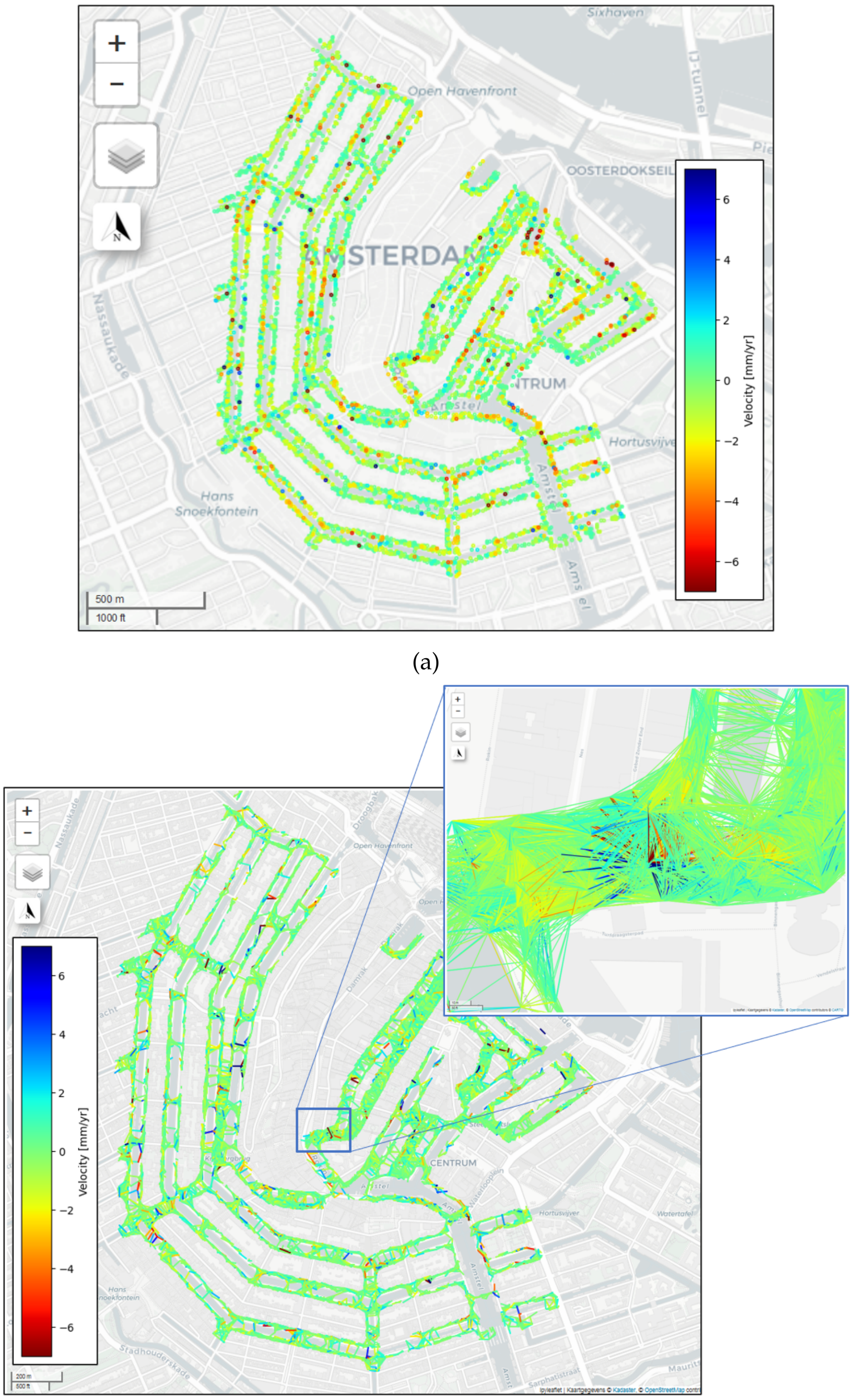
Figure 2.7.: Processing flow for deformation monitoring using arc tuning method.

The processing flow for deformation monitoring using arc-tuning goes by first generating all possible arcs within a user-defined threshold length using KD_tree approach as discussed in section 2.2. After that, we determine the displacement parameters for the selected local arcs. Then, we perform arc-tuning by selecting arcs that meet the criterion of defined arc characteristics. The selected arcs are then visualized for interpretation. Based on whether relevant arcs are selected, the user can further adjust the criterion for arc characteristics to tune into deformation signals.

2.5. Arc clustering

Given that the local arcs are better suited for localized deformation monitoring over conventional times series w.r.t. a common reference point, the local arcs can also be used for application-agnostic monitoring. For use of local arcs in application-agnostic monitoring or an application-aligned case where arc selection is not immediately possible, the problem arises in analyzing information from a lot arcs. This is because when no particular arc selection other than length is done, numerous local arcs can be generated over an area of interest that overlap one another, making it harder to analyze the arcs.

Fig. 2.8a shows the average line of sight velocity using conventional time series w.r.t. a common reference point for PS within a 20 meters of the canal edges, since in the conventional we have one time series per PS it is easy to visualize as well as subsequently analyze them.



(a)

(b)

Figure 2.8.: Example case of monitoring the area in the vicinity of the canals. (a) Conventional method of time series w.r.t. a common reference point. (b) Local arcs. The local arcs overlap one another making it hard to examine them.

But when the local arcs are taken for the same PS, it can be seen in Fig. 2.8b that there are a lot of arcs being overlapped thus making it impossible to analyze all the arcs.

In such cases we propose the use of clustering algorithms to aid the analysis. In clustering analysis, elements are grouped into so called clusters, where elements in a defined group by the algorithm are similar to one another. In here the arcs will be the elements that need to be clustered and the similarity will be determined based on parameters provided for an arc. For example, arcs can be clustered based on a combination of parameters like length, azimuth, elevation difference, displacement parameters or deformation behavior. In a problematic area we expect to see multiple independent arcs with similar arc orientation showing similar temporal behavior that may be distinct from other surrounding arcs. With clustering analysis we aim to find such patterns among a dataset of arcs, thereby narrowing down the amount of arcs to analyze.

The processing flow for arc clustering strategy is given in Fig. 2.9, we consider local arcs since we are interested in localized deformation. After that we estimate the displacement parameters and use it along with arc characteristics as inputs for the clustering algorithm. Many different clustering algorithm exist, but in this project we use the Hierarchical Density based clustering (HDBSCAN) algorithm [Campello et al., 2013], this algorithm was chosen due to its power and flexibility to handle data, particularly in finding the clusters without prior knowledge like the number of clusters to be found which are often required by other clustering algorithms. The hyper-parameters for this clustering algorithms that have to be set by a user are also fairly intuitive, for instance, *min_cluster_size* is the minimum size of the group to be considered as a cluster, and *min_samples* is a parameter based on which the amount of similarity between features (or density) of objects will be determined. HDBSCAN also has the benefit of being flexible to handle clusters of different shapes and sizes, this is particularly useful since we have feature with different units and different ranges which could form clusters of different shapes.

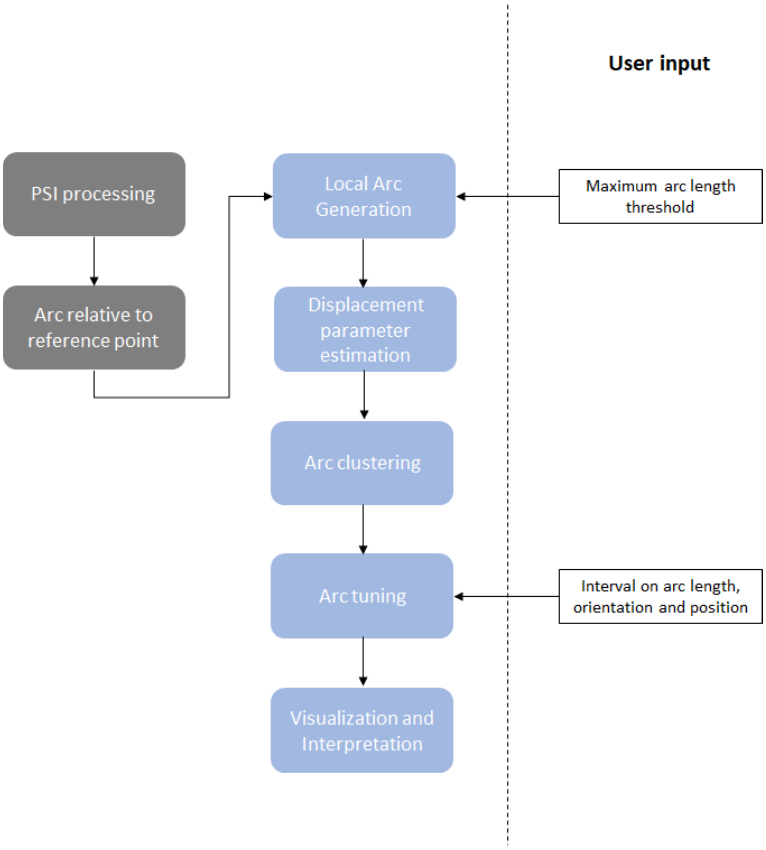

User input
PSI processing
Arc relative to reference point
Local Arc Generation
Displacement parameter estimation
Arc clustering
Arc tuning
Visualization and Interpretation
Maximum arc length threshold
Interval on arc length, orientation and position

Figure 2.9.: Processing flow for deformation monitoring using arc clustering method.

3

Results and Discussion

In this chapter, we present the results obtained using the methodologies discussed in Chapter 2. We describe the study area and data used for our experiments in Section 3.1. Then, in Section 3.2, we compare the time series of local arcs with conventional time series w.r.t. common reference point. In Section 3.3, we show an example case of the application of the arc tuning methodology. Similarly, in the final Section 3.4, we provide an example scenario of applying the arc clustering approach.

3.1. Study area and dataset

The quay walls of Amsterdam represent 200 kilometers of aging infrastructure amidst a densely populated urban environment; Fig. 3.1 shows the collapse and reconstruction efforts of one such quay wall along the canal of Grimburgwal. It is a prime example of a scenario where deformation monitoring is necessary, where the InSAR-based monitoring tools developed in this study might be helpful. For this reason, it was chosen as a study area for experiments and to demonstrate the tools developed in this project.

The dataset used in the project is processed and geo-located data obtained from the acquisitions of the TerraSAR-X satellite over the area of Amsterdam; the dataset was provided by SkyGeo. The dataset contains observations from the descending orbit from track 139. It contains 201 acquisitions from 18th March 2016 to 27th December 2022. The acquisitions are at an 11-day interval, which is the TerraSAR-X satellite's revisit period. The dataset contains displacement time series per point scatterer with respect to a common reference point. The dataset also contains information on the location of the PS points in the RD-system ($EPSG:28992$) and the height of the point in meters. As an example of the dataset used in this project, we plot the point scatterers from the dataset along the Amsterdam canals along with their LoS velocity in Fig. 3.2. Here, for representation of the point scatterers, we clipped the dataset to consider PS within 20 meters of the canal edges; this is done since quay walls are the test area for this project, so we wanted to show the quay wall network as well as the point scatterers in its vicinity.

The Bilderdijksgracht and the Grimburgwal are two regions that were chosen to demonstrate the tools in this study. The region around Grimburgwal was selected because it is a known

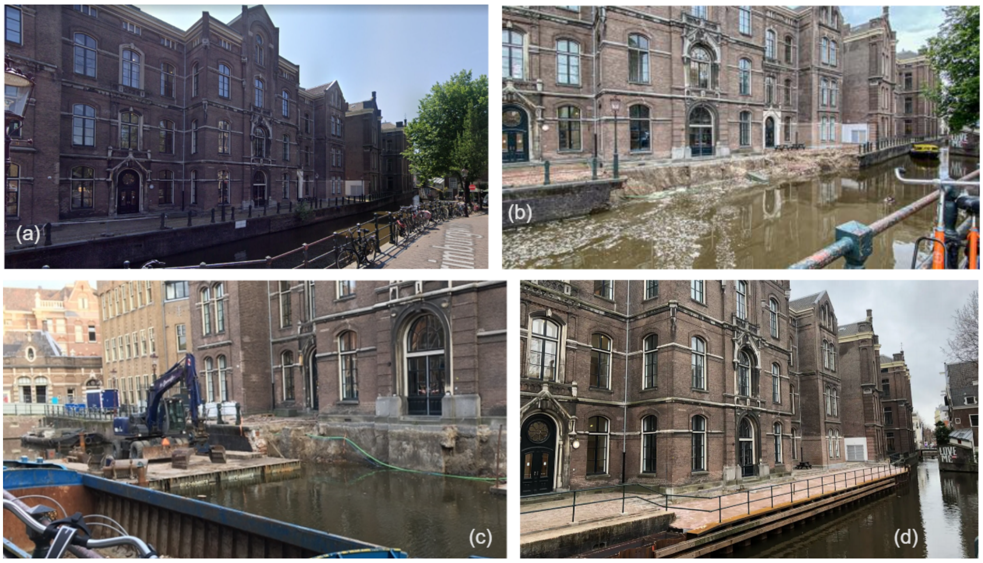


Figure 3.1.: Collapse and reconstruction efforts at the Grimburgwal canal in Amsterdam (area highlighted on Fig. 3.2 by a black rectangle). (a) Before collapse. (b) Aftermath of collapse [GA1, 2020]. (c), (d) Reconstruction efforts.

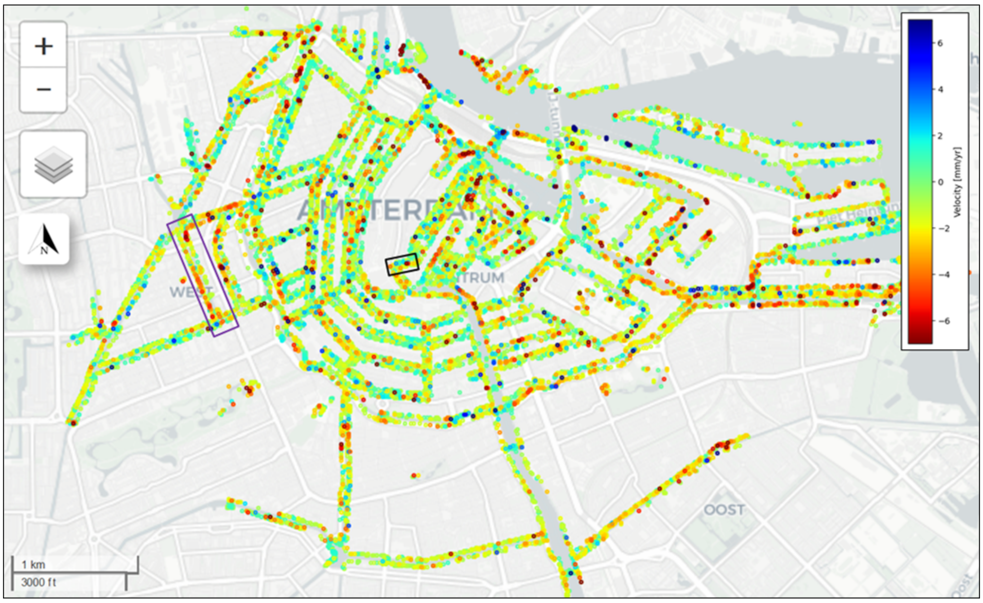


Figure 3.2.: Average LoS velocity of point scatterers (w.r.t. reference point) in the vicinity of the quay walls/canals of Amsterdam. The black rectangular box marks the Grimburgwaal and the purple rectangular box marks the Bilderdijksgracht canal.

case of quay wall instability where a collapse happened, while the region around Bilderdijksgracht was chosen because the PS in this region showed significant subsidence relative to a common reference point. These regions are shown by a purple and black rectangular box, respectively, in Fig. 3.2.

It should be noted that although only cases pertaining to monitoring of quay walls are discussed here, the methodologies that were developed are not limited to this application alone but are intended for universal use.

3.2. Conventional approach versus local arcs

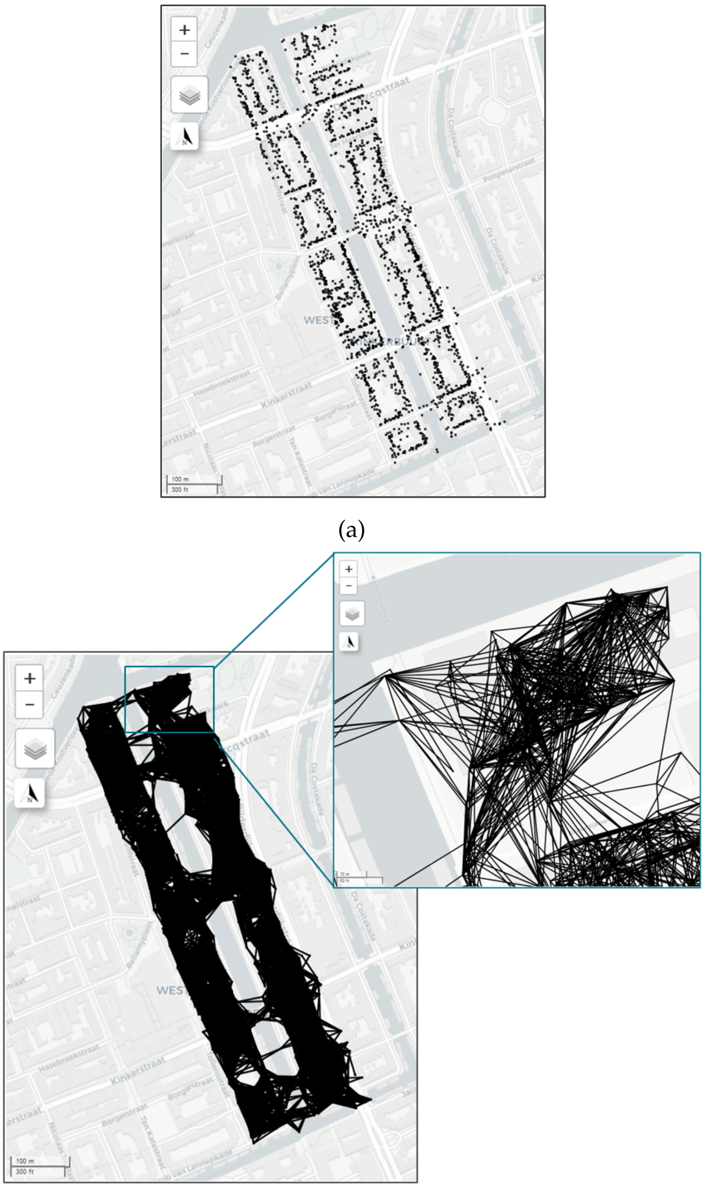
In contrast to the conventional time series of point scatterers w.r.t. a common reference point, which is better suited for wide-scale and application-agnostic monitoring, we take into consideration local arcs which are short and we anticipate that they will be better suited to observing localized differential deformation, and thus particularly useful for cases like infrastructure monitoring. With local arcs, we also expect the atmospheric noise contribution to be low to none; since the atmosphere can be considered homogeneous between PSs close to one another, any effect of atmospheric delay would be similar over both PSs and then cancel out when considering an arc between them. Reduced noise in time series would improve the ability to discriminate between different displacement models (hypotheses). A smaller variance in the displacement estimates translates into a smaller Minimum Detectable Displacement (MDD).

To determine in what ways local arcs could prove advantageous for monitoring deformation, we conducted a comparative analysis of the time series behavior between two cases: local arcs and PSs with respect to a reference point. We analyzed the time series of both cases by using deformation modeling through MHT, as described in section 2.3. For this study, we defined a test area in the region around Bilderdijkgracht. We used a subset of PS surrounding the canal; the location of the canal is indicated by a black dashed rectangle in Fig. 3.2.

For the time series of PSs w.r.t. a common reference point, we chose all the PSs in the defined area of interest. As for the time series of local arcs between PSs, we selected all arcs that were less than 60 meters in length. This was done in order to select highly localized arcs, for which the possibility of noise contribution from atmospheric effects is minimal. Furthermore, it helps limit the number of arcs generated, thus reducing processing time. We did not perform further arc selection for local arcs in this experiment. Fig. 3.3 shows the test area with the chosen PS and arcs.

To perform the MHT procedure for the time series in both cases, we defined a unit variance of $\sigma^2 = 3^2\ mm^2$ and an identity matrix as the co-factor matrix for the stochastic model in Equation (2.6). For both cases, we initialized with a significance level of $\alpha_0 = \frac{1}{2m} = \frac{1}{2*201}\%$ where m is the number of acquisitions, and a detectability power of $\gamma_0 = 50\%$.

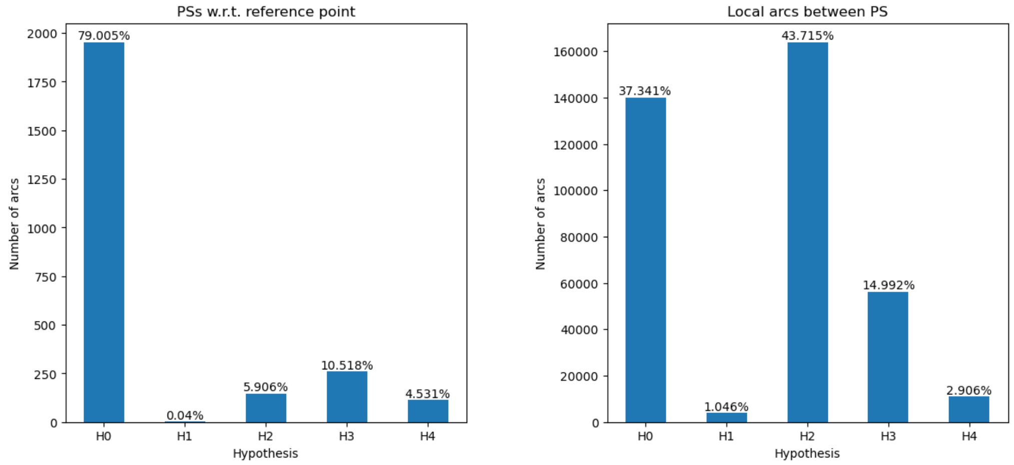
In Fig. 3.4, we can see a bar-plot of the results from the MHT procedure. Here we display the number and percentage of time series that follow each hypothesis in both cases. Since the number of time series is different between the local arcs and conventional approach dataset, we do the comparison based on percentage of time series classified under each hypothesis. In the case of conventional approach of PS w.r.t. a common reference point, for about 79% of the time series the null hypothesis of a linear steady-state model is sustained. On the other



(a)

(b)

Figure 3.3.: Test area around the Bilderdijksgracht canal for experiment in Section 3.2. (a) Point scatterers inside the test area whose time series w.r.t. a common reference point (not shown) is taken for the conventional approach. (b) Arcs less than 60 meters taken for the dataset of local arcs.



(a)

(b)

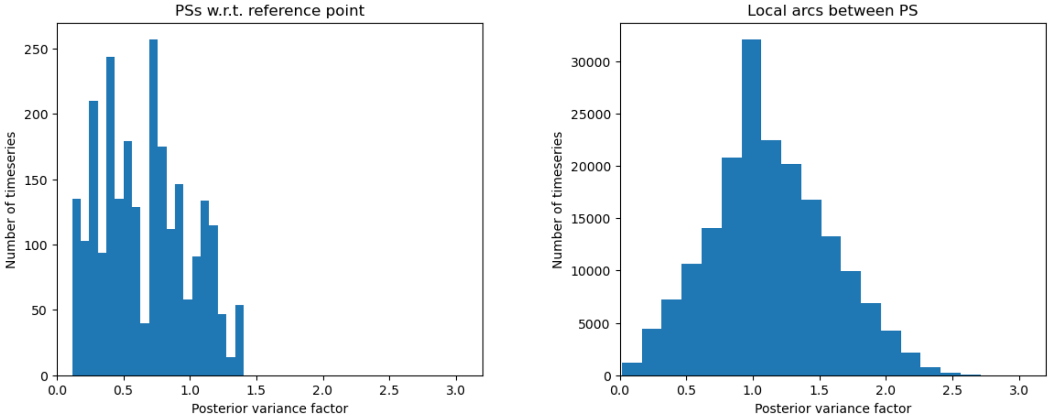
Figure 3.4.: Number of time series following each hypothesis (H0: Linear, H1: Periodic, H2: Periodic+Step, H3: Step, H4: Single Breakpoint). (a) PS w.r.t. a common reference point. (b) Local arcs between PS. The number of time series is vastly different in the two cases. To make comparison easier, the percentages of time series following each hypothesis in both cases are also provided.

hand, for local arcs, the majority of the time series reject the null hypothesis, with alternate hypothesis H_2 (Periodic + Step model) being the most common hypothesis.

Provided that there are localized deformations happening in the area, this result would have been expected since we hypothesized that local arcs would be better at capturing any localized deformations. However, such difference in results can also be an outcome of improper specification of the stochastic model or in our case the unit variance, σ^2. If the chosen variance is too conservative (hence, too high variance), then any inherent non-linear deformation would be considered noise, leading to the null hypothesis always being sustained. On the other hand, an optimistic variance (hence, too low variance) could result in the wrong hypothesis being chosen, as the models will try to fit to the noise. In order to validate the results from the MHT and determine if the defined stochastic model was appropriate we calculate the posterior variance factors.

The posterior variance factor, given by Equation (2.13), shows how well the covariance matrix Q_{yy} actually fits the mathematical model, assuming that the functional model is correct. Since the co-factor matrix R_{yy} is an identity matrix, the posterior variance factor describes the correctness of our assumption of prior unit variance σ^2. A variance factor of one for a time series would indicate that the noise in it has been quantified accurately. A variance factor of less than one indicates that the choice of prior variance has been conservative. A factor greater than one either shows that the prior variance has been optimistic or indicates the presence of unmodeled deformation. Fig. 3.5 shows the distribution of posterior variance factors for the two cases.

Since the local arcs should be less susceptible to atmospheric noise and we employed the same stochastic model in both local arcs and conventional approach, we expect that the posterior variance factors of local arcs will be less than that of conventional time series



(a) (b)

Figure 3.5.: Distribution of the posterior variance factor in (a) PSs w.r.t reference point (conventional method) (b) Local arcs between PS points.

w.r.t. a common reference point. Fig. 3.5 shows that the variance factors distribution differs between the two cases and contrary to our expectation the time series of local arcs often having higher posterior variance factors than the conventional time series of PSs w.r.t. a common reference point.

Investigating such a result that deviates from our expectation, we realized that in the conventional approach where time series of PS are taken relative to a common reference point, a PS with limited noise, i.e., high coherence, is chosen to be the reference point in order to decrease the propagation of noise from the PSs to the time series. While with local arcs, no such selection based on quality of PS is done to generate arcs. Hence, the time series of local arcs are likely to be noisier than the convention method, which explains the difference in the distributions in Fig. 3.5.

An example of such noise propagation is shown in Fig. 3.6. Since the a reference point of limited noise is chosen with the conventional method, the bad quality of estimates (obtained using conventional method) for beginning period between 2016-2019 in Fig. 3.6(a) can be attributed to the PS we are observing. A similar statement can be made for the bad quality towards the end of the time series in Fig. 3.6(b). In Fig. 3.6(c) it is seen that taking an arc between these two PS results in a time series consisting almost entirely of poorer quality estimates.

The difference in posterior variance factors as shown in Fig. 3.5, in turn also explains the results obtained form the MHT procedure in Fig. 3.4. Aside from local arcs having higher posterior variance factors than conventional methods, it is also observed that the posterior variance of most these local arcs are greater than 1.0, which means that chosen a-priori variance $\sigma^2 = 3^2 mm^2$ has also been too optimistic for the time series of most of the local arcs. Since, for an optimistic a-priori variance the MHT procedure tends to support the alternative hypothesis as the have additional parameters that results in the models fitting the noise in the data better , resulting in wrong hypothesis being chosen. This also explains Fig. 3.4b, where the majority of the time-series reject the null hypothesis and is also the reason for hypothesis H_2 with the most additional parameters being the most prevalent of all the other hypotheses.

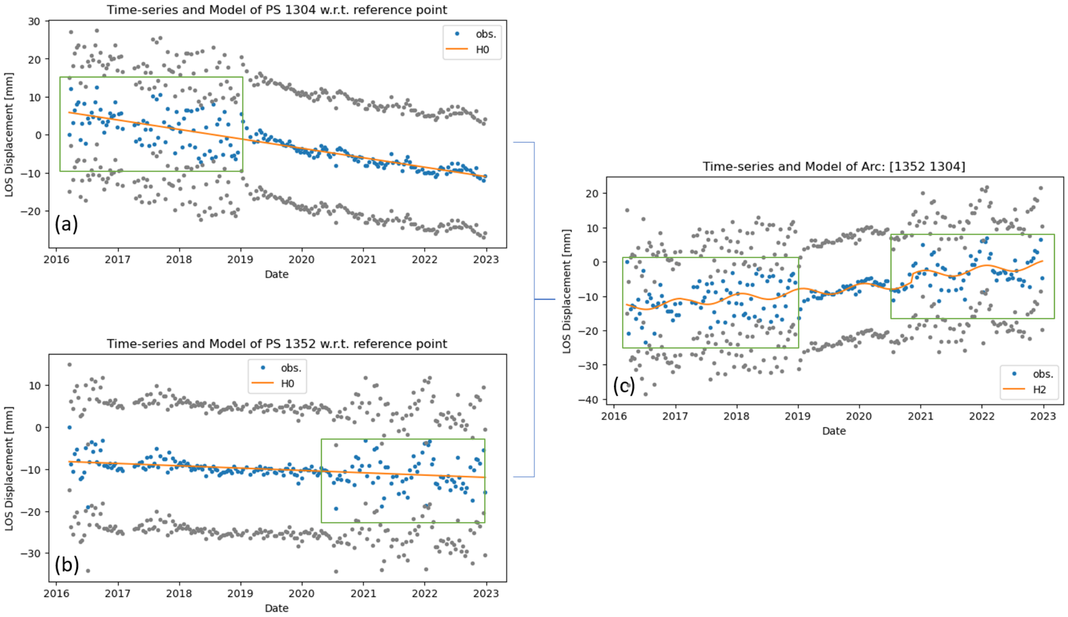


Figure 3.6.: Noise propagation in local arcs. (a) and (b) give the time-series of PSs w.r.t. reference point. (c) shows the time-series of local arcs. The propagation of noise in between PS and arc is highlighted with a green rectangular box.

Table 3.1.: Parameter estimates and supported hypothesis of the time series of the points shown in the left column of Fig. 3.7.

Point	Model	$\hat{v}$	$\hat{a}$	$\hat{\sigma}^2$
		[mm/y]	[mm]	[mm]
2153	**H0**	−0.29	−2.00	3.13
2257	**H0**	0.53	0.75	2.89
2371	**H0**	−0.86	−0.05	3.12
2458	**H0**	0.14	1.72	3.13
2008	**H0**	−1.05	3.07	2.90
2217	**H0**	0.51	−1.08	2.97
1696	**H0**	−2.69	0.38	3.53
1945	**H0**	0.05	3.60	2.89

Table 3.2.: Parameter estimates and supported hypothesis of the time series of the arcs shown in the right column of Fig. 3.7.

Arc	Model	$\hat{v}$	$\hat{s}$	$\hat{c}$	$\hat{\Delta}$	$\hat{v}_b$	$\hat{a}$	$\hat{\sigma}^2$
		[mm/y]			[mm]	[mm/y]	[mm]	[mm]
2153-2257	**H1**	−0.86	−1.2	2.33			−0.22	2.98
2371-2458	**H2**	−2.06	−1.10	0.001	5.18		0.03	2.93
2008-2217	**H3**	−0.44			−6.14		2.17	2.99
1696-1945	**H4**	−5.48				3.87	0.55	2.93

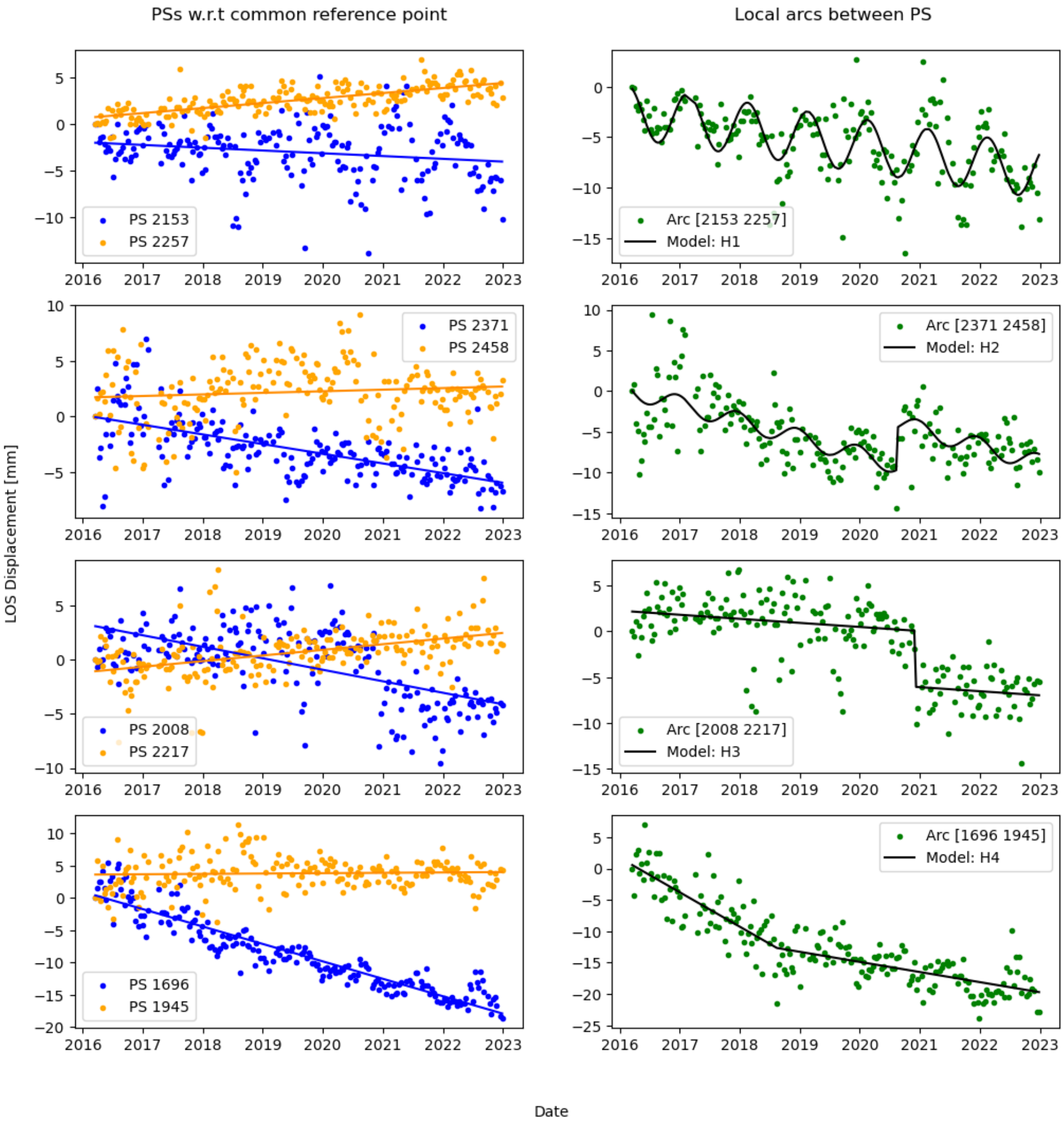


Figure 3.7.: The displacement time seiries of PSs w.r.t. a common reference point (in orange and blue), where the null hypothesis H0 (linear) was sustained. The displacement time series of a local arc formed between the two PSs (in green) where null hypothesis was rejected, showing alternate hypotheses H1 (Periodic), H2 (Periodic+Step), H3 (Step) and H4 (Single Breakpoint).

Investigating the results of the MHT procedure we found cases where the prior variance was well defined (posterior variance factor ≈ 1.0), where that local arcs revealed deformation behavior that wasn't visible in the time series of PS w.r.t. reference point. Fig. 3.7 displays four examples of such outcomes. Each plot in the left column shows the displacement time series and deformation models of two PSs with respect to the reference point. The adjacent plot to the right demonstrates the displacement time series and model for the local arc between the two PSs. We have deliberately selected cases where the behavior of the individual PSs with respect to the reference point remained consistent with the null hypothesis, while the time series of the arc between the two PSs rejected the null hypothesis. The displacement parameters and posterior variances of the results displayed in Fig. 3.7 are given in Tables 3.1 and 3.2.

Even though we found that the displacement time series of local arcs tend to be noisier than PS w.r.t. a common reference point, the local arcs can still be beneficial for localized deformation monitoring with the only downside that the quality of measurements for local arcs will not always be better than PS w.r.t. a common reference point and hence won't be able to detect deformations at a lower minimum detectable displacement as previously anticipated.

3.3. Demonstration of arc tuning

In this section, we provide an example scenario demonstrating the application of the arc tuning methodology for arc selection and deformation monitoring in the region surrounding the canal of Bilderdijksgracht.

We clip out the PS in the area of interest and generated all local arcs within a set threshold arc length. Here, all arc smaller than 100 meters in the between the PS in the area of interest were generated. The displacement parameters for each of the generated arcs are estimated using the MHT procedure. In this case, we use a conservative prior variance of $\sigma^2 = 4.5^2 mm^2$ in Eq. (2.6) for the time series of local arcs. We initialized with a significance level of $\alpha_0 = \frac{1}{2m} = \frac{1}{2*201}\%$ where m is the number of acquisitions, and a detectability power of $\gamma_0 = 50\%$.

The arc tuning methodology allows the selection of relevant arcs based on our monitoring needs. For instance, if a scenarios call for the localized deformation monitoring of the western side quay wall along the canal of Bilderdijksgracht, we can start by selecting arcs orthogonal to the canal and setting threshold on arc length. This way, we can monitor the motion of the points along the canal relative to points away from the canal since points away from the canal can be considered to be unaffected by the deformation of quay walls. This can be done by selecting only arcs that have azimuth angle between $60° - 80°$ so that they are orthogonal to the canal and arc length between $30 - 59m$ so that the points are sufficiently farther away from each other.

Fig. 3.8 shows the estimated velocity of all the selected arcs. Whenever arc tuning is done, if multiple independent arcs are exhibiting similar behavior, such as the ones within the rectangle, it may represent a considerable portion of an object (e.g., the quay wall) or a region facing similar deformation, and if the deformation behavior in these arcs are also significant (high velocity in this case), such regions may require further investigation or mitigation.

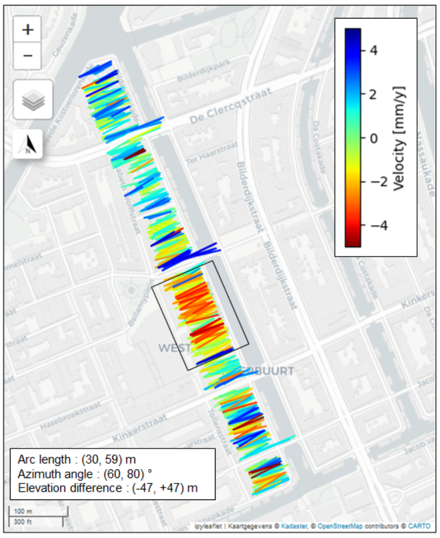
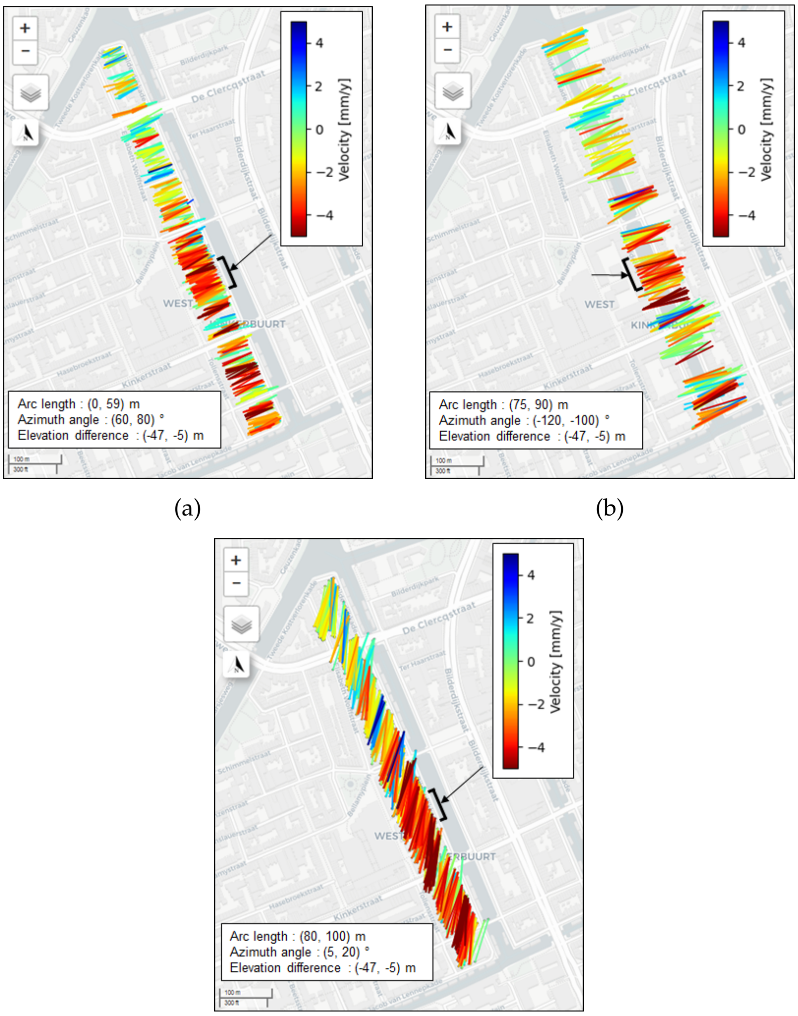


Figure 3.8.: Average LoS velocity of arcs that are orthogonal to the canal. The velocity shown represents the movement of the points along the quay wall (canal) relative to the points on the building.

As the arc tuning tool allows us the flexibility to select arcs based on convenience, any monitoring objective can be examined from different perspectives with different sets of arcs, which can aid in the interpretation of deformation signals. We monitor the canal from three new perspectives in Fig. 3.9. For the first case in Fig. 3.9a, we choose the same orthogonal arcs as before but with the additional criteria that the elevation difference between the points of an arc should be less than -5 meters so as to ensure that all the selected arcs represent the relative motion of points on the canal respective to points on the buildings. For the following case in Fig. 3.9b, we adjust the criteria of azimuth angle by 180 ° and accordingly increase the criteria for arc length to monitor the motion of the canal respective to the buildings on the opposite side of the canal. Finally, in Fig. 3.9c, we take arcs that still represent the relative motion of points on the canal respective to points on the building on the same side. However, now we have increased the arc length and slightly changed the azimuth angle so that the arcs are not orthogonal like earlier.

These different perspectives obtained due to arc tuning allow us to tune and check for deformation signals and aid in interpreting the deformation source. As an example, in Fig. 3.9, we see that the arcs formed along the section highlighted by the black arrow show similar behavior in all three cases. Such consistent behavior in the three cases allows us to confidently attribute the source deformation observed to the quay wall, rather than the buildings.

According to the arc tuning result based on arc geometry parameters, we further investigate the deformation behavior of the arcs. Fig. 3.10 shows the arc selection based on arc tuning and deformation behavior. While, for the MHT procedure, most of the arcs here sustain the null hypothesis of linear steady-state behavior, the other few arcs showing non-linear behavior still provide valuable information for monitoring. Also, arc selection based only on arc geometry often leads to overlapping arcs, where problems arise when the arcs need to be visualized on a map as information from underlying arcs is hidden by the overlapping

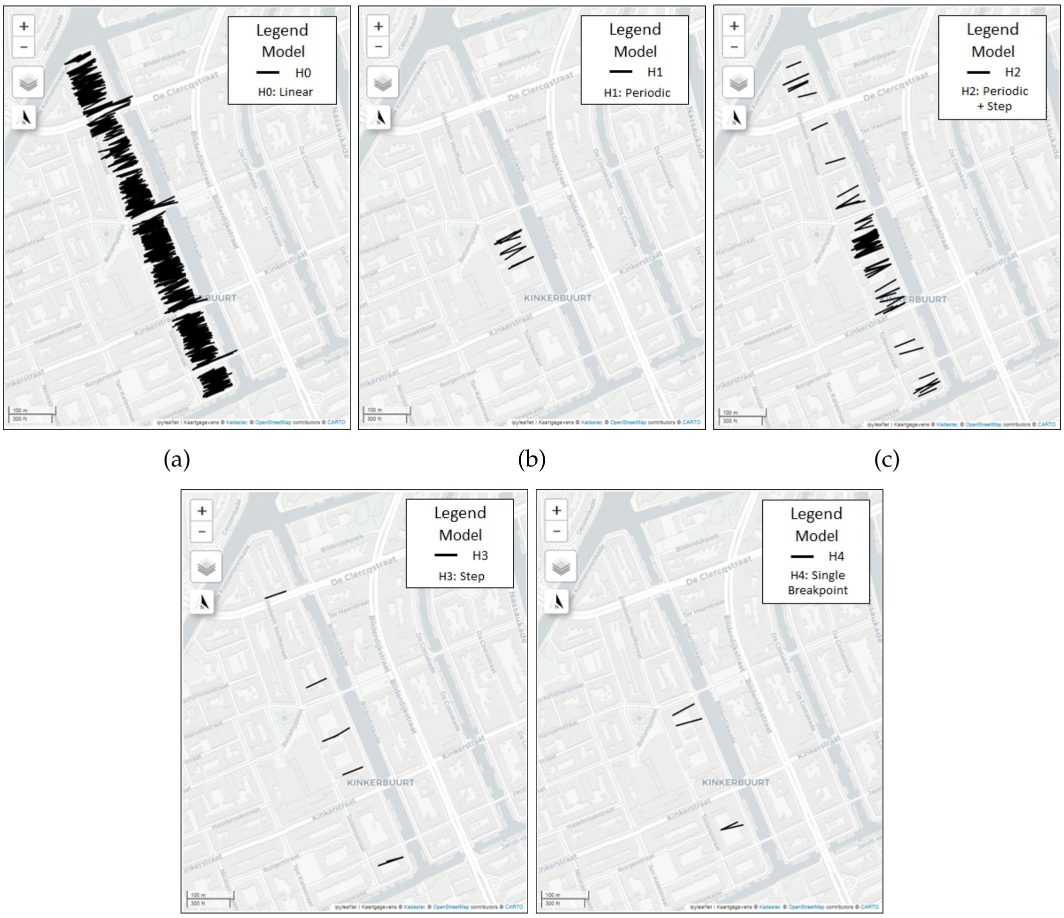

Arc length : (0, 59) m
Azimuth angle : (60, 80) °
Elevation difference : (-47, -5) m
Arc length : (75, 90) m
Azimuth angle : (-120, -100) °
Elevation difference : (-47, -5) m
Arc length : (80, 100) m
Azimuth angle : (5, 20) °
Elevation difference : (-47, -5) m
Velocity [mm/y]
4
2
0
−2
−4
De Clercqstraat
Bilderdijkstraat
Kinkerstraat
WEST
KINKERBUURT
100 m
300 ft

(a)

(b)

(c)

Figure 3.9.: Arc tuning applied to monitor the quay wall form different points of reference, which are (a) relative to points on the building on the same side of the quay wall, (b) relative to points on the building on the opposite side of the quay wall and (c) relative to points on the building on the same side of the quay wall but the arcs are not orthogonal.



(a)

(b)

(c)

(d)

(e)

Figure 3.10.: Arcs orthogonal to the canal selected as a result of arc tuning in order to observe motion of points along the canal relative to points away from the canal (Azimuth angle: (60, 80)°, Arc length: (30, 60)m). Using MHT the behavior of the time series of the arcs were found to be supporting the hypothesis (a) H0 (Linear + Constant) , (b) H1 (Periodic) , (c) H2 (Periodic + Step), (d) H3 (Step) and (e) H4 (Single Breakpoint).

arcs. In such scenarios, additional arc selection criteria based on deformation behavior or displacement parameters would allow us to be more selective and thus aid interpretation.

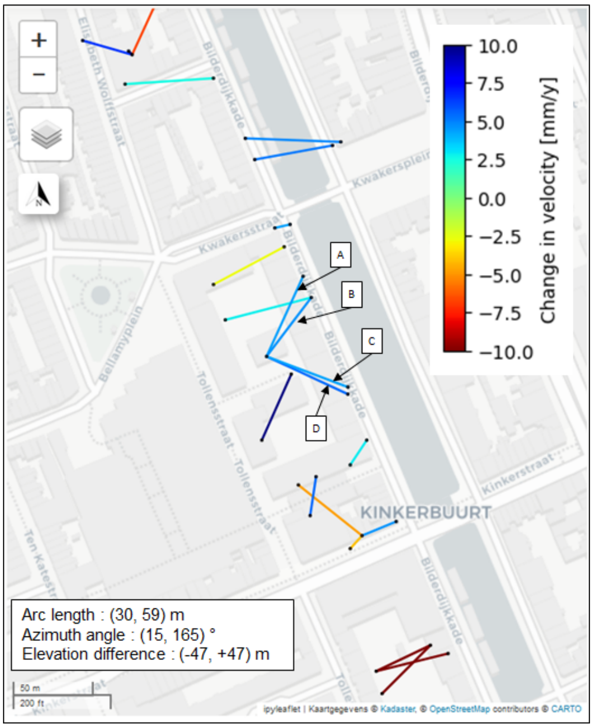
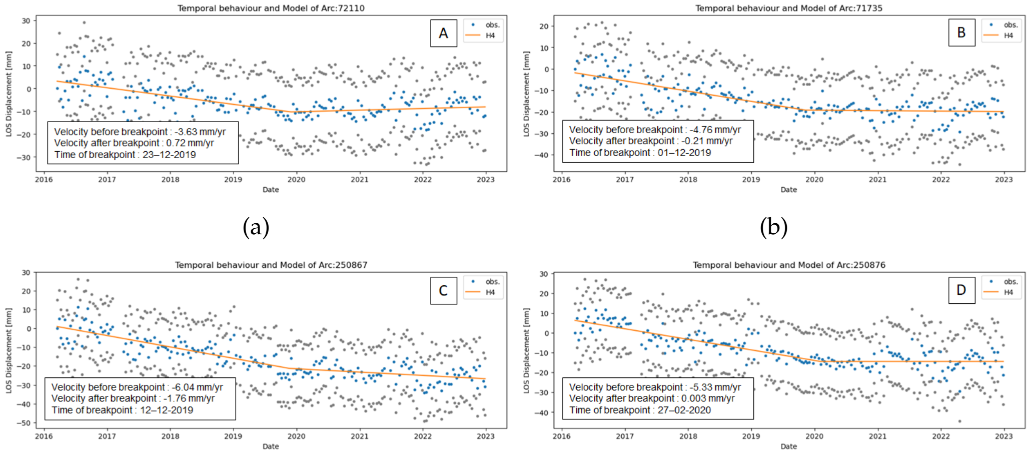

10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−10.0
Change in velocity [mm/y]
A
B
C
D
KINKERBUURT
Arc length : (30, 59) m
Azimuth angle : (15, 165) °
Elevation difference : (-47, +47) m
50 m
200 ft
ipyleaflet | Kaartgegevens © Kadaster, © OpenStreetMap contributors © CARTO

Figure 3.11.: Arc selection done based on arc behaviour of single breakpoint model in addition to selection by arc tuning.

Fig. 3.11 shows the case where arc tuning was done based on arc geometry and estimated deformation behavior. In this case, since only a few arcs in the region show single breakpoint behavior, we could set a wide criterion for the azimuth angle of arcs (from 15° to 165°). In doing so, we could observe the arcs highlighted as A, B, C, and D in Fig. 3.11, which arose from the same PS on a building. The time series of these highlighted arcs along with relevant parameters is shown in Fig. 3.12, it can be seen from the figure that the amount of velocity change and time of occurrence of breakpoints is similar between the four arcs. Since the four arcs are connected with the same PS located on a building, cases such as this allows us to interpret with a high likelihood that the source of the behavior is from the building.

With this demonstration, we showed how arc tuning can be used for arc selection and how a monitoring objective can be monitored from different perspectives by selecting different sets of arcs, which could aid in the interpretation of results. The arc selection need not be limited to arc geometry parameters alone; we showed that in combination with the MHT procedure, criteria for arc selection can be applied based on deformation behavior or to estimated displacement parameters.



(a)

(b)

(c)

(d)

Figure 3.12.: Time series and estimated displacement parameters of the arcs highlighted in Figure 3.11.

3.4. Demonstration of arc clustering

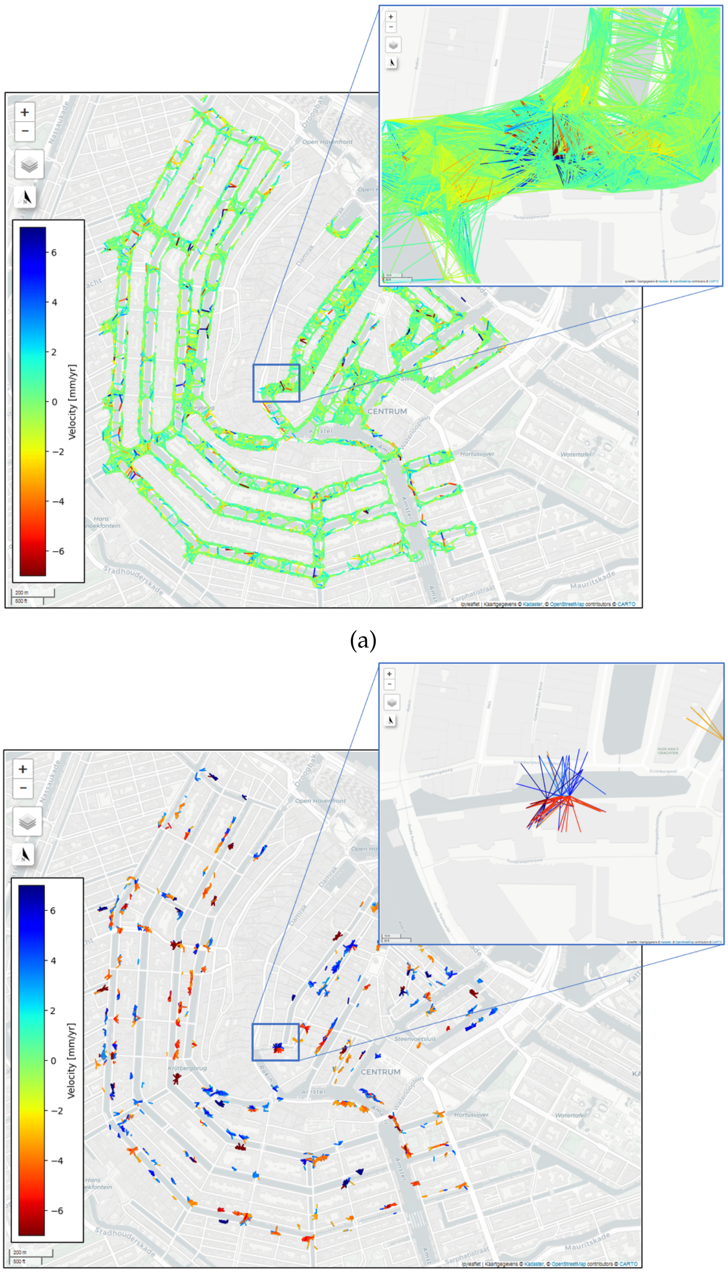
In the previous section, we discussed the arc tuning tool, which was designed for a targeted diagnostic scenario where in-depth analysis is necessitated for a specific area of interest. However, using this tool for monitoring on a larger scale would not be practical. For example, if we want to monitor all the quay walls in the city using local arcs instead of just one specific quay wall as shown in the previous section, we would need to perform the arc tuning process manually for each canal since each canal has unique orientations, which might not be feasible.

For cases of monitoring on a large scale where arc tuning is not feasible or application-agnostic (non-targeted) monitoring using local arcs, we propose to find risk areas based on patterns such as group arcs showing similar commonalities. This is referred to as arc-clustering.

3.4.1. Realization

This method involves using the clustering algorithm, in order to identify such patterns among data set of arcs. These patterns consist of multiple arcs with similar orientations that are in close proximity to each other and also exhibit similar deformation behavior and the behavior of these arc groups being in contrast to rest of the arcs in the surrounding area. An example of such a pattern was observed in the previous section, highlighted by a rectangle in Fig. 3.8. Such patterns may indicate that a particular region or object is experiencing significant deformation in comparison to its surrounding thus becoming a region of interest for further analysis. Multiple independent arcs showing similar behavior also signify that a deformation phenomenon might indeed be present and is not due to other factors such as phase unwrapping imperfections.

An example demonstration of a test scenario where arc clustering could be beneficial. For this consider a situation where localized deformation monitoring in necessitated for moni-



(a)

(b)

Figure 3.13.: Map of Amsterdam with inset over Grimburgwal. (a) Local arcs generated that are less then 50 meters in length. The high density of arcs can be seen in the inset. (b) Clusters of anomalous local arcs obtained as a results from the Arc clustering method. The inset shows the five anomalous clusters over Grimburgwal, also see Figure 3.14.

toring the quay walls, roads and buildings in the vicinity of the entire canal of network of Amsterdam. Using local arcs would be beneficial for localized deformation monitoring in such a case.

For this test case since we are only interested in the area in around the canal we considered all point scatterers within a 20 meter buffer zone of the canals. To generate only local arcs an arbitrary length threshold of 50 meters was set for the arcs and all arcs between the considered points scatterers were that meet the criteria were generated. All the generated local arcs around the canal is shown in Fig. 3.13a, it can be seen from the inset that the arcs generated are dense and overlap one another making interpretation harder. The displacement parameters for all of the generated arcs are also estimated using the MHT procedure. In this case, we use a conservative prior variance of $\sigma^2 = 4.5^2 mm^2$ in Eq. (2.6) for the time series of local arcs. We initialized with a significance level of $\alpha_0 = \frac{1}{2m} = \frac{1}{2*201}\%$ where m is the number of acquisitions, and a detectability power of $\gamma_0 = 50\%$.

As mentioned in Section 2.5, the clustering of arcs works by grouping them based on the similarity of given arc parameters. Similarity among arcs is found by the algorithm based on the arc parameters provided to it. In this test case, the similarity is found based on the parameters arc length, azimuth, elevation difference, the position of the arc, and the estimated average velocity of each arc. These parameters for all the generated arcs are given as input for the HDBSCAN (clustering) algorithm.

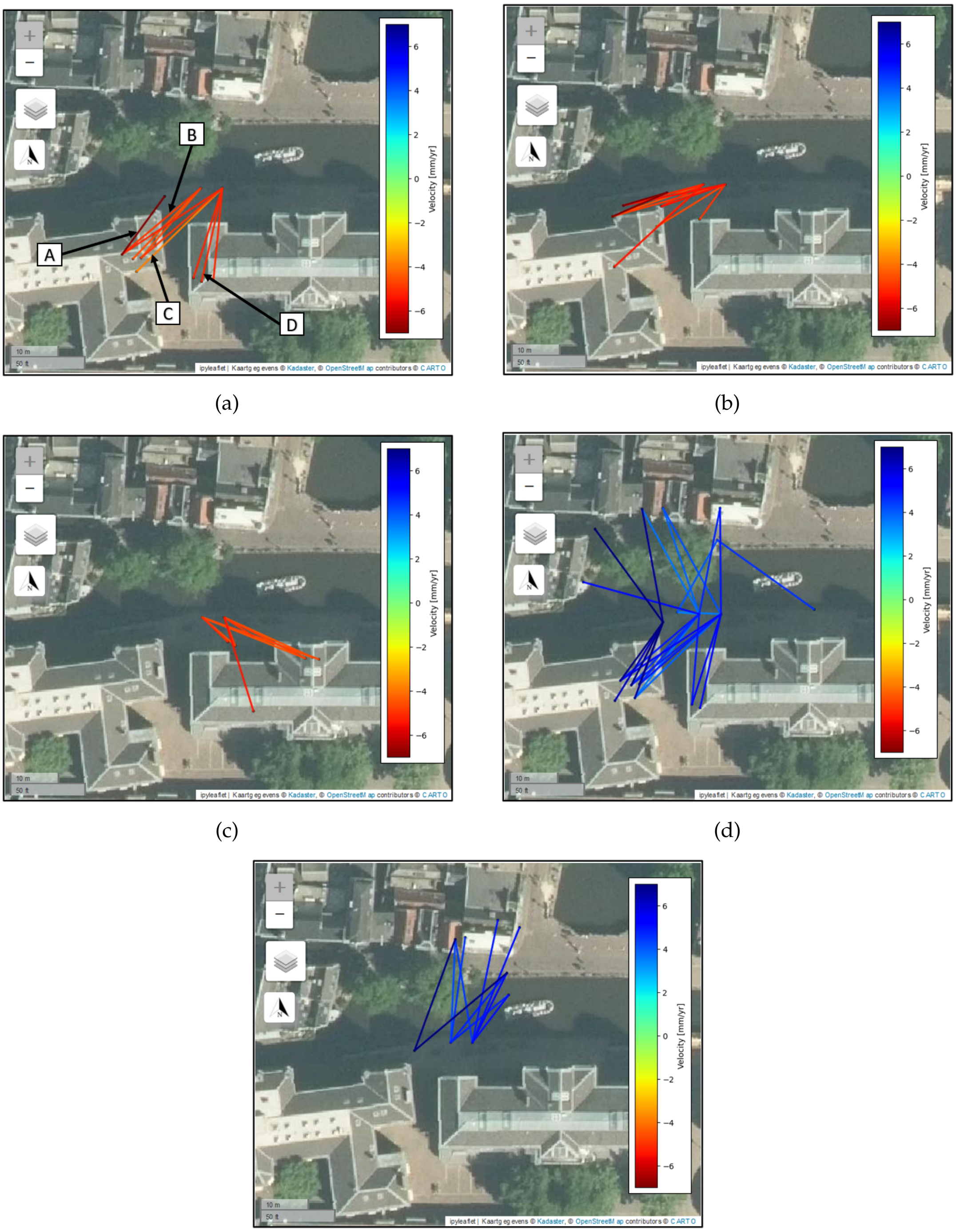
Fig. 3.13b shows arcs from all the clusters found by the HDBSCAN algorithm. Where arcs within the same clusters should have similar orientations and velocity while also being in close proximity to each other.

3.4.2. Discussion

It can be seen from Fig. 3.13b that the clustering algorithm aids in narrowing down the number of arcs that need to be analyzed, as analysis can now be limited to the location of the arc formed in clusters. A key point to note is that by using clustering algorithms, it's highly unlikely that all arcs that may be relevant or show anomalous behavior are selected to form clusters. Hence, loss of information should taken into account. Despite this, clustering provides an effective way to explore and analyze a large set of arcs.

The clusters symbolize potential areas for further analysis. One of the highlights from the results of the clustering was that algorithm was able to identify five clusters around the Grimburgwal, as shown in Fig. 3.14. The Grimburgwal is a known area of quay wall instability, see Section 3.1. The arcs in Figs. 3.14a to 3.14c are in red because they represent subsidence, indicating the movement of the quay wall concerning the buildings. On the other hand, the arcs in Figs. 3.14d to 3.14e are in blue and represent the movement of the buildings respective to the quay walls, which explains the different anomalous behavior observed in each cluster. Thus, all the deformation observed in the arcs of the five clusters was due to the subsidence at the location of the points on the quay wall along the Grimburgwal canal. The difference in behavior was only due to how the arcs were taken. The subsidence observed at Grimburgwal from these clusters are at the exact location where a quay wall collapse happened in September 1st, 2020.

By using the arc clustering method, we were able to find anomalous deformation signals in local arcs from a verifiable case of infrastructure collapse, thus demonstrating the potential of the arc clustering method. Clustering algorithms like the HDBSCAN are meant to be used for exploratory data analysis, and thus, they should be used in the same capacity, preferably


A
B
C
D
Velocity [mm/yr]
6
4
2
0
−2
−4
−6
10 m
50 ft
ipyleaflet | Kaartg eg evens © Kadaster, © OpenStreetM ap contributors © C ART O

(a)

(b)

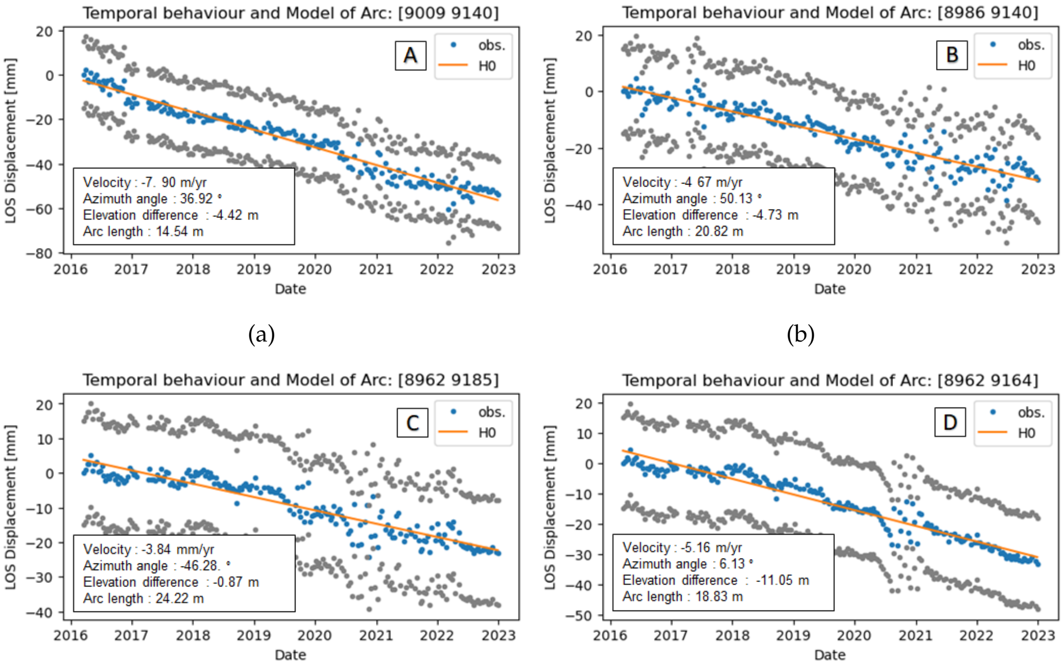
(c)

(d)

(e)

Figure 3.14.: Clusters of arcs showing of anomalous deformation at the Grimburgwal canal, obtained as a result from HDBSCAN algorithm. The cluster of arcs represents pattern such as multiple arcs with similar orientation and similar anomalous behavior in close proximity denoting occurrence of significant deformation in the area.

as a precursory tool for arc tuning or further analysis. Thus, the tool should also not be expected to find the occurrence of all such anomalous patterns in a given dataset of arcs.



(a)

(b)

(c)

(d)

Figure 3.15.: Time series, estimated deformation model and displacement parameters of the arcs highlighted in Fig. 3.14a.

4
Conclusions and Recommendations

This chapter contains the conclusions that were derived from our work along with recommendations for future work.

4.1. Conclusions

The main research objective of this study is to answer the following question:

How can application-aligned monitoring and interpretation be enabled using satellite radar interferometry?

Application-aligned monitoring was enabled by the selection of relevant arcs. Key findings of this research are subsequently answered in the sub-questions:

- *How do localized arcs compare to conventional time series with respect to a common reference point for deformation monitoring?*

  Comparison of displacement time series was done by modeling deformation behavior using multiple hypothesis testing methodology.

  By comparing the behavior of time series between local arcs and conventional time series w.r.t. a common reference point, we found cases where, for the same set of PS points, the conventional time series w.r.t. a common reference point was determined to show linear behavior while the arc between those points was shown to follow non-linear behavior, thus determining the capacity of arcs taken between two PS to provide additional information on deformation behavior over conventional time series w.r.t. a common reference point for the same two PS.

  In the multiple hypotheses testing procedure, we used the same stochastic model for local arcs and conventional time series w.r.t. a common reference point. However, since noise in each PS might differ, the noise in the resulting time series in local arcs as well as time series w.r.t. a common reference point might also differ. We calculated the posterior variance factor to assess how well the stochastic model was described. Based on the obtained posterior variance factors, we found that local arcs generally have higher noise than conventional time series w.r.t. a common reference point, contrary to our expectation. In the conventional method, the common reference point is chosen such that there is less noise propagating into the time series formed. With local arcs, no such selection is made based on the quality of arcs, which resulted in time series of local arcs being noisier than the conventional approach.

Thus, local arcs are helpful for monitoring by providing additional information on deformation behavior over conventional methods. However, the noise in the time series of local arcs is likely to be higher than in the time series from conventional methods.

- *How can user-initiated arc selection be implemented for given case of deformation monitoring?*

  We introduced arc tuning for application-aligned monitoring using InSAR. In arc tuning, necessary arcs can be selected by setting criteria on arc parameters: the length of an arc, its azimuth, the elevation, and its position. This way, we have a user-initiated method where arcs are generated according to monitoring needs. Additionally, arc tuning enables the monitoring of an object from different perspectives, which could help in a better understanding and interpretation of the deformation behavior of arcs. For example, with arc tuning, a bridge can be monitored relative to an adjacent road or relative to another adjacent building. Observing similar anomalous behavior of the bridge from multiple perspectives would substantiate the interpretation that anomalous deformation is indeed happening on that bridge. Furthermore, the criteria for arc selection need not be limited to arc geometry parameters alone; with the combination of Multiple Hypothesis Testing procedure, criteria for arc selection can also be done on the estimated displacement parameter.

  For application-aligned monitoring on a larger scale, i.e., for monitoring a whole canal network instead of just one canal, using local arcs for monitoring might prove beneficial, but performing arc tuning for the whole network might not be optimal. In such cases, a dataset of all local arcs can initially be generated, and patterns in the dataset, such as multiple independent arcs in proximity showing similar anomalous behavior, could be taken as an indication of an anomalous region or object. We introduced arc clustering, where we use the HDBSCAN algorithm to input the arc parameters to find such patterns among the datasets of arcs. While demonstrating the arc clustering for a dataset of arcs over Amsterdam, the approach managed to detect clusters of arcs showing anomalous behavior at Grimburgwal in the exact location where a quay wall collapse occurred in September 2020, showcasing the potential of the arc clustering methodology. Since with arc-clustering arcs that are not determined to be a part of a cluster by the algorithm are disregarded there is a loss of information. For this reason, arc clustering is recommended as an exploratory tool but not as an end product.

4.2. Recommendations

The key feature of arc tuning is that relevant arc can be selected by tuning (or experimenting) with the criterion on arc parameter. Therefore, proper visualization of arcs is paramount to determine if the arcs from the set criterion correspond to the monitoring application. While we developed an interactive visualization tool as a solution for this, the tool is inefficient and prone to crash when arcs greater than the order of thousands need to be visualized; this is primarily because in the current tool each arc is individually visualized iteratively. The use of GeoDataFrame data structure in Python and GeoData class in the Ipyleaflet Python visualization library could make this process more efficient by eliminating the need of iteration. Furthermore, since the arcs are three-dimensional, a three-dimensional visualization would be more appropriate given the availability of building models [Peters et al., 2022] or point clouds such as Actueel Hoogtebestand Nederland.

Arc tuning based on setting criteria on arc geometry parameters or displacement parameter estimates has been discussed in this study, However, the criteria need not be limited to these parameters and can also be set based on other metrics, e.g., quality of an arc.

The displacement time series of arcs were obtained by subtracting the time series (w.r.t. a common reference point) of the two point in an arc. We observed that unwrapping errors were prevalent in the time series of most of the arc obtained with this current method. Therefore, it is crucial to detect and correct unwrapping errors or implement unwrapping techniques for a more accurate interpretation.

For the arc clustering strategy, only the potential of using clustering algorithm with arcs is discussed in this study that too only for one HDBSCAN algorithm. Future direction of research for clustering would include assessing the choice of input parameters for the algorithm, comparison against other clustering algorithms and feature engineering, where the input data for the algorithm, e.g., arc parameters, are modified in order to improve the performance of the clustering algorithm.

For the MHT procedure used in this study only a limited set of functional models based on prevalent deformation behaviors is used to estimate the displacement parameters. However, if there is prior knowledge on expected behavior or there is a specific monitoring objective for a particular behavior, more relevant functional models can be used.

While estimating the displacement parameters of arcs, for the stochastic model, in our current approach, we use a generalized assumption of a unit variance and an identity matrix as co-factor matrix for all arcs, which unfortunately falls short as the quality of observations varies between arcs and also might vary in time for the same arc. Thus, a more proper stochastic model per arc is needed in order to get more reliable displacement parameter estimates.

Bibliography

(2020). Quay wall in Univ. Amsterdam collapse was repaired in May; In trouble for weeks.

(2021). Amsterdam and NWO join forces for urban bridges and quay walls . *NWO*.

(2021). Vernieuwing oudere infrastructuur is urgent. *TNO*.

(2023). InSAR at a glance.

ASCE (2021). ASCE's Infrastructure Report Card Gives U.S. 'C-' Grade, Says Investment Gap Trillion, Bold Action Needed. *American Society of Civil Engineers*.

Baarda, W. (1968). *A testing procedure for use in geodetic networks*. Netherlands Geodetic Commission.

Campello, R. J. G. B., Moulavi, D., and Sander, J. (2013). Density-Based Clustering Based on Hierarchical Density Estimates. In *Advances in Knowledge Discovery and Data Mining*, pages 160–172. Springer Berlin Heidelberg.

Chang, L., Dollevoet, R. P. B. J., and Hanssen, R. F. (2018). Monitoring line-infrastructure with multisensor sar interferometry: Products and performance assessment metrics. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 11(5):1593–1605.

Chang, L. and Hanssen, R. F. (2016). A Probabilistic Approach for InSAR Time-Series Post-processing. *IEEE Transactions on Geoscience and Remote Sensing*, 54(1):421–430.

Crosetto, M., Monserrat, O., Cuevas-González, M., Devanthéry, N., and Crippa, B. (2016). Persistent Scatterer Interferometry: A review. *ISPRS Journal of Photogrammetry and Remote Sensing*, 115:78–89.

Ferretti, A., Monti-Guarnieri, A., Prati, C., and Rocca, F. (2007). *Insar Principles Guidelines for SAR interferometry processing and interpretation*. ESA Publications Division.

Ferretti, A., Prati, C., and Rocca, F. (2000). Nonlinear subsidence rate estimation using permanent scatterers in differential sar interferometry. *IEEE Transactions on Geoscience and Remote Sensing*, 38(5):2202–2212.

Hanssen, R. F. (2001). *Radar Interferometry*. Springer Netherlands.

Kampes, B. (2005). *Displacement parameter estimation using permanent scatterer interferometry*. PhD thesis.

Maneewongvatana, S. and Mount, D. M. (1999). Analysis of approximate nearest neighbor searching with clustered point sets.

Peters, R., Dukai, B., Vitalis, S., van Liempt, J., and Stoter, J. (2022). Automated 3d reconstruction of lod2 and lod1 models for all 10 million buildings of the netherlands.

Skolnik, M. I. (1980). *Introduction to radar systems: Merrill I. Skolnik*. McGraw-Hill.

Teunissen, P., Simons, D., and Tiberius, C. (2004). *Probability and observation theory: AE2-E01*. TU Delft.

Ullrich, P. and Zarzycki, C. (2017). TempestExtremes: A framework for scale-insensitive pointwise feature tracking on unstructured grids. *Geoscientific Model Development*, 10:1069–1090.

Van Leijen, F. (2014). *Persistent Scatterer Interferometry based on geodetic estimation theory*. PhD thesis.

van Natijne, A. (2018). Locating PS-InSAR derived deformation using LiDAR point clouds. MSc thesis.

A

Appendix

A.1. Critical values for multiple hypothesis testing

We calculate the reference non-centrality parameter λ_0 with an initial setting of $q = 1$, $\alpha_0 = \frac{1}{2m}$ and $\gamma_0 = 50\%$ as shown in Figure 2.4. The critical value K for the OMT test given for a dimension $q = m - n$, where m is the number of epochs of observations and n is the number of parameters in the null hypothesis.

For the descending dataset with a time series of 201 epochs of measurement ($m = 201$), the critical value for OMT was calculated to be $K = 207.4554$.

Table A.1.: Critical values for MHT for the descending orbit dataset, see section , of $m = 201$ epochs of measurements.

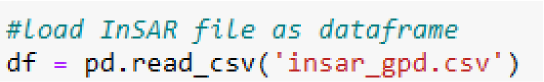
q_j	α_j [%]	$\chi_{\alpha_j}(q_j)$
1	0.25	9.1497
2	0.62	10.1682
3	1.08	11.1847

A.2. Workflow of the arc tuning tool

In this appendix, we will provide a guide on how to use the ’Arc_tuning_tool’ Jupyter notebook which has been created during this study. The notebook can be accessed via Github, see A.3. The objective of this guide is to provide information regarding the code/tool, which will aid in replicating the experiments of this thesis or in adapting the tool for use in other cases. While providing a brief overview of the workflow this guide also serves the purpose to highlight key file/user inputs that can be modified according to users use case.

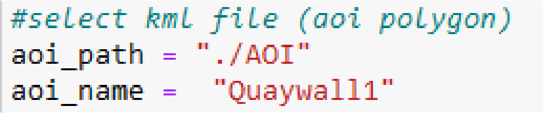
In the beginning the notebook/tool requires two main data files. The first is a file containing a dataset of pre-processed InSAR observations in a Comma Separated Value (CSV) format. Any new InSAR data file should contain data in the same format as the provided ’insar_gpd.csv’ file. The second file contains contains a user defined polygon over an Area of interest in the Keyhole Markup Language (KML) format, which will be used to clip out relevant points scatterers and their accompanying from the rest of the dataset. The polygon in KML format can be created using the “Add Polygon” tool in the Google Earth Pro application.

After clipping the point scatterers over our AOI form the rest of the dataset, the next step is to generate the all arcs within a user specified length in between these points scatterers. This is efficiently done using the local arc connection strategy. As mentioned, the process requires a user input of the maximum arc length based on which arc are formed between the selected point scatterers.



```
#load InSAR file as dataframe
df = pd.read_csv('insar_gpd.csv')
```

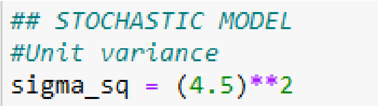
Figure A.1.: Variable to input processed InSAR data file in the code.



```
#select kml file (aoi polygon)
aoi_path = "./AOI"
aoi_name =  "Quaywall1"
```

Figure A.2.: Variable to input Area of Interest KML file in the code.

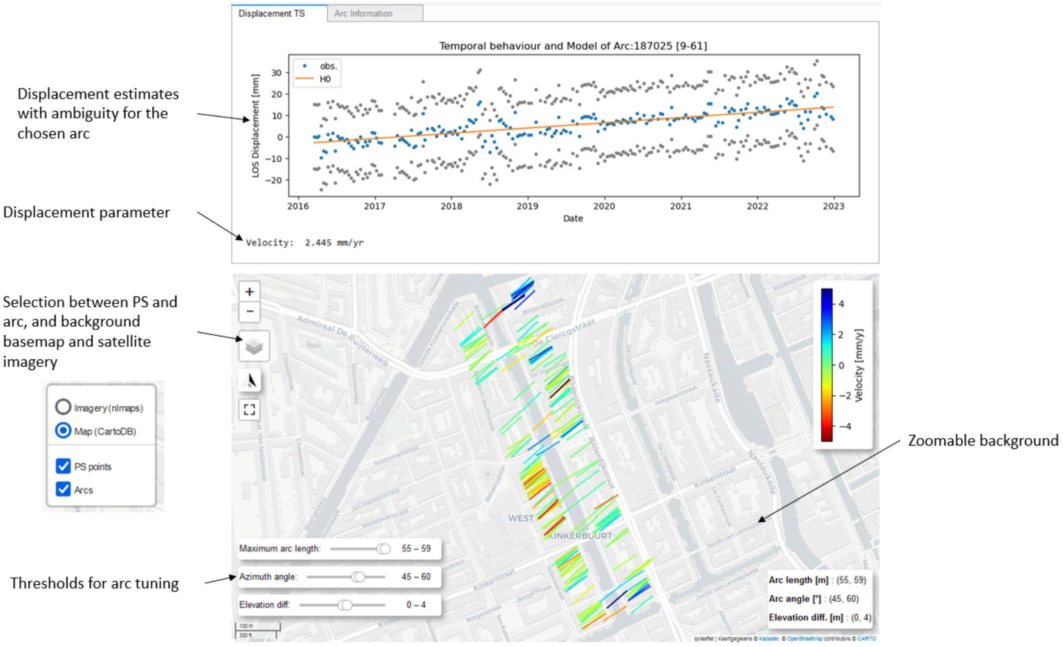
The arc tuning tool is also combined with displacement parameter estimation using Multiple Hypothesis Testing. The code has a steady state liner behavior as the null hypothesis and four non-linear behavior as alternative hypothesis. see . With sigma = 4.5^2 and critical values based on our InSAR dataset. These must be changed in case other datasets are used. It should be noted that the critical values for the MHT were pre-calculated, which should also be changed if other InSAR observations are used.



```
## STOCHASTIC MODEL
#Unit variance
sigma_sq = (4.5)**2
```

Figure A.3.: Variable to input unit variance for MHT.

To perform the arc tuning and efficiently communicate any information on arcs we made a interactive visualization tool that works inside the Jupyter notebook script.



Displacement estimates with ambiguity for the chosen arc

Displacement parameter

Selection between PS and arc, and background basemap and satellite imagery

Zoomable background

Thresholds for arc tuning

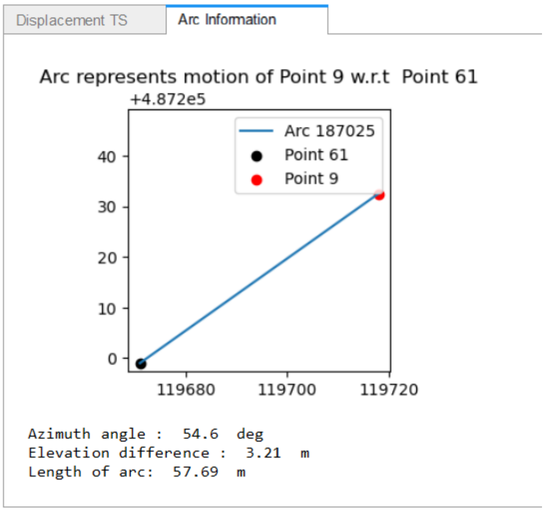
Velocity: 2.445 mm/yr

○ Imagery (nlmaps)
● Map (CartoDB)
☑ PS points
☑ Arcs

Maximum arc length: 55 – 59
Azimuth angle: 45 – 60
Elevation diff: 0 – 4

Arc length [m] : (55, 59)
Arc angle [°] : (45, 60)
Elevation diff. [m] : (0, 4)

Figure A.4.: Interactive visualization tool



```
Azimuth angle :  54.6  deg
Elevation difference :  3.21  m
Length of arc:  57.69  m
```

Figure A.5.: Arc information tab

The features of the tool are:

- Selection of map layers: this includes the ability to set the background of the tool as either a map or satellite imagery and the option to plot PS points, Arcs, or both together.
- An interactive background: The map can be zoomed and dragged. The PS points and Arcs are clickable, and the tool displays relevant information on the chosen object in the tabs shown in the top of Fig. A.4.
- An interactive slider is present in order to set thresholds for arc tuning,
- The tool shows relevant information in multiple tabs. The ”Displacement TS” tab shows the deformation times series of the chosen object along with the optimal model estimated using Multiple Hypothesis testing. The displacement parameters are also provided in this tab. The ”Arc information” tab contains information regarding the arc characteristics and a plot that describing how the displacement time series is obtained. see Figure A.5.

A.3. Codes

This appendix contains the links to data, jupyter notebooks and python enivronment file used to obtain the results of this thesis. All the items refereed above can be found in the Github reprository: (https://github.com/rammohan-c/MSc_Thesis-Arc-Tuning.git)