



MOMIT
**MULTI-SCALE OBSERVATION
AND MONITORING OF RAILWAY
INFRASTRUCTURE THREATS**

Grant Agreement No: 777630

Project Acronym: MOMIT

Project Title:

**Multi-scale Observation and Monitoring of railway Infrastructure
Threats**

This project has received funding from the Shift2Rail Joint Undertaking under the European Union's Horizon 2020 research and innovation programme under grant agreement No 777630

Technical note: ADAFinder - A tool to automate the detection of Active Deformation Areas

This Document is composed by 31 pages, including attachments

Last revision date: 2018-04-09

Dissemination Level: CO = Confidential

Deliverable information

Document Configuration

Programme Name:	Shift2Rail
Call:	H2020-S2RJU-2017
Topic:	S2R-OC-IP3-03-2017
Grant Agreement Number:	777630
Project Acronym:	MOMIT
Project Title:	Multi-scale Observation and Monitoring of railway Infrastructure Threats
Template Version:	v1.2
Deliverable Number:	Technical note
Title of the Deliverable:	ADAFinder - A tool to automate the detection of Active Deformation Areas
Work-Package:	WP3
Due Date:	2018-08-31 (M12)
Issue Date:	2018-07-18
Revision of the document:	v1.0
Dissemination Level:	CO = Confidential
Responsible Beneficiary:	CTTC
Prepared by:	José Navarro (CTTC)
Reviewed by:	Maria Cuevas>(CTTC)
Contributors:	-

Revision History

Revision	Date	Author	Organisation	Description
v0.9	2018-04-16	José Navarro	CTTC	First issue
v1.0	2018-07-18	José Navarro	CTTC	Definition of the format of the defaults file is removed from this document and written as a separate technical note. The necessary ammendments to refer to such technical note have been made in this document. The same applies to the installation procedure, which now is written in a separate document.

Acknowledgement

This project has received funding from the Shift2Rail Joint Undertaking under the European Union's Horizon 2020 research and innovation programme under grant agreement No 777630.

Statement of originality

This document contains original unpublished work except where clearly indicated otherwise. Acknowledgement of previously published material and of the work of others has been made through appropriate citation, quotation or both.

Disclaimer

This publication reflects the author's views. The Joint Undertaking is not responsible for any use that may be made of the information it contains.

The information contained in this document and any other information linked therein is confidential, privileged and it remains the property of its respective owner(s). As such, and under the conditions settled in the MOMIT Grant Agreement and the MOMIT Consortium Agreement, it is disclosed for the information of the intended recipients within the MOMIT Consortium and the European Commission / Shift2Rail Joint Undertaking according to its "Dissemination Level" and may not be used, published or redistributed without the prior written consent of its owner(s).*

** PU = Public; CO = Confidential, only for members of the consortium (including the JU and Commission Services) EU-R/R-UE = Classified, as referred to in Commission Decision 2001/844/EC.*

Table of Contents

LIST OF TABLES	5
LIST OF FIGURES	6
ACRONYMS AND ABBREVIATIONS	7
1 EXECUTIVE SUMMARY	9
2 DISCLAIMER	10
3 THE ADAFINDER TOOL	11
3.1 THE PROCESS	11
3.2 THE GUI.....	13
3.3 THE DEFAULTS FILE.....	14
3.3.1 (A partial list of) The available options.....	15
3.3.2 The extra, batch options.....	20
3.4 THE READ / WRITE MAP FILE	21
3.5 THE INPUT (DATA) FILES	24
3.5.1 The PS (points) file	25
3.5.2 The optional boundaries file.....	25
3.6 THE OUTPUT FILES.....	25
3.6.1 The ADA file	25
3.6.2 The points file	26
3.7 INSTALLING THE TOOLS.....	27
3.8 USING THE TOOL	27
4 CONCLUSIONS	29
REFERENCE DOCUMENTS	30

List of Tables

Table 1: The available options in an ADAFinder defaults file	18
Table 2: Arrangement of a QI table	19
Table 3: Example of QI classes assignment.....	19
Table 4: The batch options in an ADAFinder defaults file.....	21
Table 5: The attributes in the read map file	23
Table 6: The attributes in the ADA shapefile	26
Table 7: The attributes in the output points shapefile.....	27

List of Figures

Figure 1: ADAFinder's GUI..... 13

Figure 2: Selection of the subset of points to write..... 13

Figure 3: QGIS showing a map with ADAs (purple areas) 29

Acronyms and Abbreviations

ADA	Active Deformation Area
AOI	Area of Interest
CRF	Coordinate Reference Frame
GUI	Graphical User Interface
PS	Persistent Scatterers

Technical note:
ADAFinder - A tool to automate the detection
of Active Deformation Areas

1 Executive Summary

This technical note presents ADAFinder, a tool to identify automatically Active Deformation Areas (ADA). The basics of the algorithm used are briefly described, as well as the kind and characteristics of the input, output and configuration files. The Graphical User Interface (GUI) and command line tools, although very concisely, are also explained.

2 Disclaimer

This document is the user's guide of the ADAfinder tool. It is neither a description nor a tutorial of the methodology it implements, which is described in [RD1]. Therefore, previous, solid knowledge of such methodology is required to fully exploit the potential of ADAfinder. A thorough reading of [RD1] is consequently recommended.

3 The ADAFinder tool

The ADAFinder tool is targeted at the automatic detection of Active Deformation Areas (ADA) implementing the methodology thoroughly described in [RD1].

The application has been developed in C++ for efficiency reasons. Both a command line and a Graphical User Interface (GUI) are offered to the user. The GUI simplifies noticeably the operational procedure; the command line version may be used to integrate this tool in a batch production line, if necessary.

The tool accepts and produces Esri shapefiles as input and output for compatibility reasons. These files are easily managed by a wide number of Geographic Information System (GIS) tools. Among these, ArcGIS and QGIS. For a description of the shapefile format, please see [RD2].

Some options controlling the behaviour of the application are not shown in the GUI for simplicity reasons; normally, these extra options may be considered as project-wide ones so it would make no sense changing each value each time the tool is executed (in the context of some given project). Not appearing in the GUI does not mean that these options cannot be modified; these are stored in a defaults file that the tools opens each time it is started. In fact, the default values shown in the GUI when the tool is started (see Figure 1 on page 13) are read from the aforementioned defaults file. This defaults file is, at the same time, the file where all the options are specified for the batch version of the tool.

ADAFinder is able to adapt itself to (slightly) different input shapefile formats. There is a minimal set of data that an input PS shapefile must contain to be processed by this application. Providing that such data is present in the shapefile, their positions in the file (“column”) may differ depending on the origin of the data. A configuration file (“read map”) may be used therefore to define where to find the necessary data without having to modify the application.

On output, ADAFinder creates two files: the first one contains the polygons defining the areas covered by the ADAs identified by the application. The second one stores a subset of the input points, those that meet some criteria.

The following sections describe ADAFinder in more detail. This encompasses the GUI, the formats of the configuration, input and output files as well as guidelines about how to use it.

3.1 The process

As already stated, ADAFinder implements the methodology described in [RD1]; such methodology is briefly sketched below for reference purposes only.

- Read the input files. These are (1) an Esri shapefile containing the Point Scatterers (PS) to analyse and *optionally* (2) another Esri shapefile including one or more polygons defining the Areas of Interest (AOI) where the actual process must take place
- Compute the standard deviation of the velocities of the PS in the point shapefile (σ_v)
- If a shapefile containing the boundaries of the AOI has been provided, then clip the point

dataset to remove those PS falling outside the requested limits.

- Remove isolated PS. A PS is an isolated one when no other PS exist within a distance defined by the user.
- Remove stable PS (a PS is considered as stable or not comparing its velocity with the standard deviation, corrected by a multiplier provided by the user, of the velocities σ_v),
- Look for clusters of points that are close enough (the distance is set by the user). Those clusters with a number of PS below a given threshold (set by the user) are discarded. This is the second isolation filter.
- Look for ADAs. An ADA is a set of PS that are close enough to each other. Again, the distance defining the meaning of “close enough”, that is, the radius of influence, aka the clustering radius, is set by the user.
- Compute ADA statistics.
- Compute polygons representing ADA’s areas (concave hulls).
- Write output ADA and PS shapefiles.

For more information about the details concerning the algorithm above, please refer to [RD1].

3.2 The GUI

Figure 1 depicts the graphical interface of the ADAFinder (GUI version) tool.

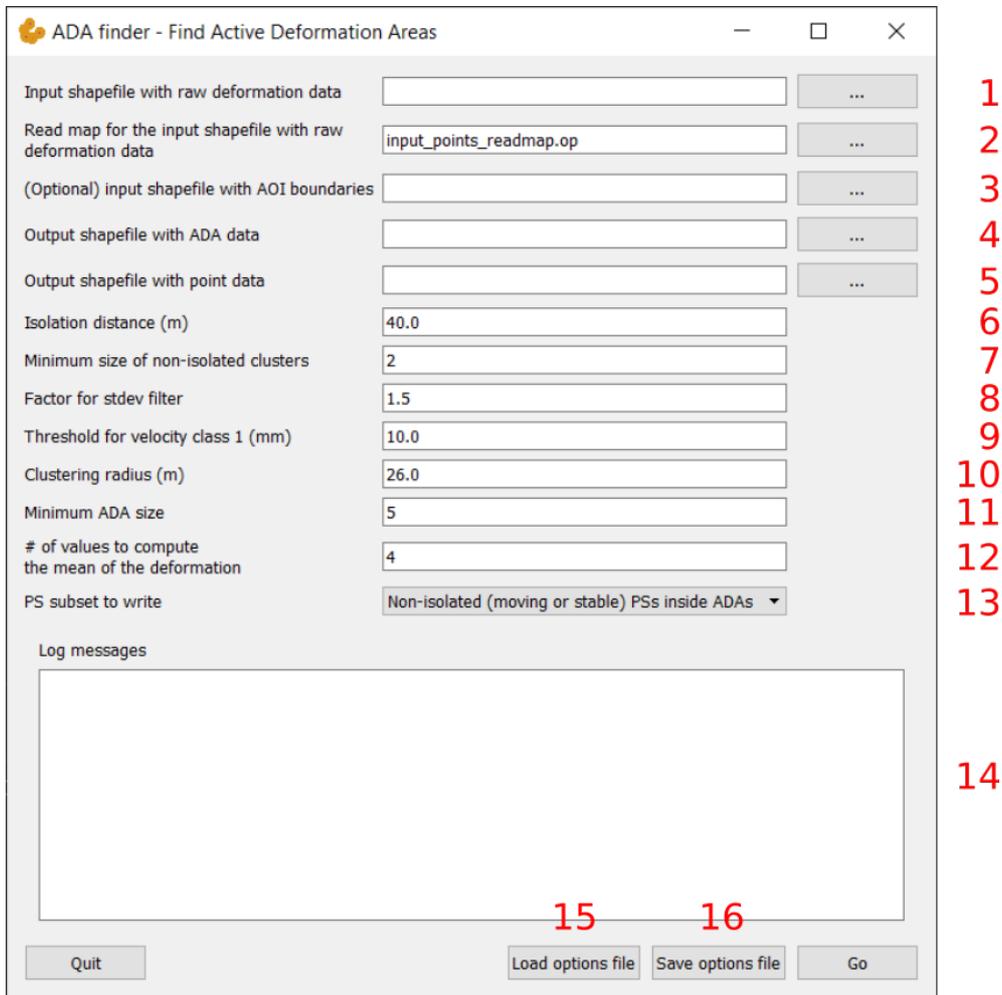


Figure 1: ADAFinder's GUI

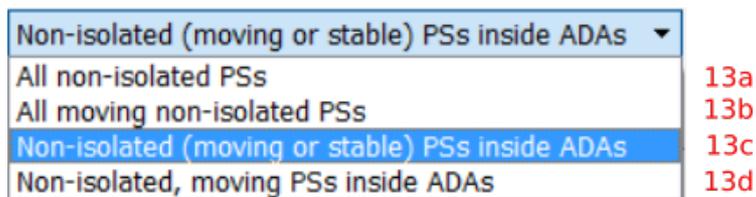


Figure 2: Selection of the subset of points to write

Note that:

- Items 1 to 5. The names of the several input and output files. These include the input PS (raw deformation data), the read map defining how to read the PS file (see section 2.4 for details), the optional boundaries defining the AOI and the two output files, one containing the ADAs and other storing the PSs.

- Items 6 and 7. These are the parameters defining the meaning of “isolated point”. The first one, the isolation distance, defines the radius of isolation around a PS. Points with no neighbours within such radius are considered isolated and thus removed. The second parameter is used when there are some other PS within the previous area; if there are less points than the number specified there (“Minimum size of non-isolated clusters”) then the PS being checked is removed. In the figure this value is two, meaning that groups of 2 points will be considered isolated and thus removed.
- Item 8. This is a multiplier to apply to the standard deviation of the velocities of the PS. The resulting value is used as the threshold to consider a point as stable or not.
- Item 9. Velocity threshold to consider a PS as belonging to class 1 or 2.
- Items 10 and 11. The radius of influence to consider the points surviving to the several filters applied to the original dataset as belonging to the same cluster. That is, the points that, directly or indirectly are within the “Clustering radius” are going to be considered as members of an ADA, but only (item 11) if the number of PS in such ADA is greater than or equal to the “Minimum ADA size”.
- Item 12. Number of elements in the deformation time series of each PS to take into account to compute the mean of the deformation. Given a number n , then the last n elements in each time series are used to compute the statistic.
- Item 13. This dropdown list is used to select the *subset of points* in the input PS file that will be written to the output one. The several options are shown in Figure 2; these are:
 - Item 13a: the whole set of PSs that survive to the isolation filter. That is, isolated PSs are excluded from this subset. Any other PSs are in.
 - Item 13b: Like 13a, but removing those PSs that are considered stable (velocity filter).
 - Item 13c: Like 13a, but in this case the PSs must belong to an ADA. That is, any non-isolated PS inside an ADA is written, no matter whether it is stable or not.
 - Item 13d: Like 13c, but in this case the stable PSs, even when inside ADAs, are excluded from the subset to write.
- Item 14. Log area. The messages describing the progress of the process will be shown here.
- Items 15 and 16: these two buttons let the user load and save, respectively, files with options controlling the behaviour of the application.

For more details on the meaning of the several fields described above (especially for fields 6 to 13), please refer to [RD1].

Note that some of the options controlling the behaviour of ADAFinder are not shown in the GUI but stored in the so-called defaults file. Please refer to section 2.3 for details about it.

3.3 The defaults file

When using the GUI version of the tool, a defaults file must reside in the same directory than the application’s executable. A defaults file is just a plain text file with a set of defaults options that the GUI application will read on start up and show to the user. When processing numerous datasets for which the parameters controlling the behaviour of the application are the same or almost the same, providing the appropriate defaults file may help to save time and reduce the number of human errors.

The command-line tool accepts a full path and name to this defaults (options) file as an input parameter.

When working with the GUI version of the tool, only a subset of the options included in this file may be modified by means of the graphical interface of the application. Some others are considered either too complex to be offered to the regular user or too immutable as to be changed each time the application is run, so these are not shown in the GUI and may be changed in the defaults file only. By immutable parameters the reader must understand, essentially, “project wide parameters”, that is, those whose values will not change when the tool is used in some specific project. Not showing these in the GUI makes it clearer and less burdensome.

When using the GUI version of the tool, the defaults file must reside in the same directory than the tool’s executable and its name must be exactly `ADAFinder_default_configuration.op`.

Note, however, that the defaults values loaded from the defaults file may be overridden by loading (button 15 in Figure 1) another options file. Such options file adhere to the same format that defaults files (in fact, these are exactly the same).

The **syntax** of the ADAFinder defaults (or user-saved options) file is **described in [RD3]**.

3.3.1 (A partial list of) The available options

The options that may be present in an ADAFinder’s defaults file are summarized in Table 1. Note that ADAFinder expects to find **all** the options listed in the aforementioned table. Missing parameters or incorrect values of these will make the application complain about the error. Note also that in the description of each option, references to the GUI items in Figure 1 are made, so it is possible to link the names of such options in the options file and their counterparts in the GUI.

Label	Description	Comments
INPUT_SHAPEFILE_POINTS_READMAP	(Full path and) file name of the read map file defining how the input file with points must be read. See section 2.4. Item 2.	Valid (full path and) file name.
ISOLATION_DISTANCE	Distance (in meters) to consider a point as isolated because of the lack of neighbors. Usually, it is two times the longitude of the longest pixel side. Item 6.	Floating point number. Greater than 0.
ISOLATION_CLUSTER_SIZE	Minimum number of points in a cluster that makes it "not isolated". That is, even when a PS has a certain number of points around it (see ISOLATION_DISTANCE) , this point will be considered as isolated (and therefore, removed) if the total number of points is not greater than or equal than this value. Item 7.	Integer value. Greater than 0.

Label	Description	Comments
STDEV_FACTOR	Multiplier that applied to the global standard deviation of the whole velocities set will produce the threshold to consider a point as stable or not. Item 8.	Floating point number. Greater than 0.
THRESHOLD_FOR_VELOCITY_CLASS_1	Threshold (mm) to consider a velocity as belonging to class one. (if $ v >$ this value) then velocity class = 1. Item 9.	Floating point number. Greater than 0.
ADA_MINIMUM_CLUSTER_SIZE	Minimum number of points in a cluster to make it an ADA. Clusters with less points will be discarded. Item 11.	Integer value. Greater than 2.
ADA_RADIUS	Radius of influence (in meters) to build the ADA (usually, 1.3 times the longest pixel side). To be part of an ADA, a point must be close enough to at least another point. This distance sets the threshold for this proximity criterion. Item 10.	Floating point number. Greater than 0.
ADA_SHIFT_TO_CM	Number of decimal places to shift the decimal point in such a way that the units of the projected coordinates of the input points become, at least, cm. This is necessary for the buffering operation applied to the ADA polygons; the library used to perform the buffering works with integer coordinates only, so scaling these to finer units is a must to avoid losing meaningful precision.	Integer value. Greater than or equal to 0.
ADA_DISPLAY_MODE	Display mode. ADAs may be represented as a series of circles whose centers are located at the points making the ADA (namely, the "influence area") or building the (buffered) concave hull surrounding these points. For influence areas, set the ADA_DISPLAY_MODE to 1. For concave hulls, set it to 2. Other values are illegal.	Integer value. Either 1 or 2.
ADA_CIRCLE_STEPS	Number of steps to approximate the circle representing the area of influence of every point in an ADA cluster. The higher this number, the better circle approximation obtained, but also the more CPU time to compute it. Note that this parameter makes sense only when the ADA_DISPLAY_MODE is set to 1 ("influence area"). When ADA_DISPLAY_MODE is set to 2 ("concave hull") this parameter is ignored, although it must be present in this defaults file.	Integer value. Greater than 3.

Label	Description	Comments
N_VALUES_MEAN_DEFORMATION	<p>Number of values in each time series that will be used to compute the accumulated mean of the deformation. This number may never be greater than the actual number of values in the time series. If n is the value of this parameter, then only the LAST n values in each time series are used.</p> <p>Item 12.</p>	Integer value. Greater than 1. Less than or equal to the actual number of values in the time series in the input points file.
POINT_SUBSET_TO_WRITE	<p>Subset of points in the input PS file that have to be written to the output one. Legal values and their meanings are:</p> <ol style="list-style-type: none"> 1. Write all non-isolated points, no matter whether these are stable or moving. 2. Write all moving, non-isolated points. 3. Write only those non-isolated points (either moving or stable) that belong to an ADA. 4. Write only those moving, non-isolated points that belong to an ADA. 	Integer. Value in the range [1..4].
TNITABLE_N_VALUES	Number of threshold values in the TNI (Temporal Noise Index) table. See the explanations in the current section for more details about tables.	Integer value. Greater than 1.
TNITABLE_THRESHOLDS	<p>The TNI threshold values themselves. Note that there must be as many values as stated in parameter TNITABLE_N_VALUES above. Note that these thresholds are used to check autocorrelation values; therefore, these must be in the range (-1, 1) and sorted in ascending order.</p> <p>See the explanations in the current section for more details about tables.</p>	List of floating point values. Each value in the range (-1, 1). List sorted in ascending order.
SNITABLE_N_VALUES	Number of threshold values in the SNI (Spatial Noise Index) table. See the explanations in the current section for more details about tables.	Integer value. Greater than 1.
SNITABLE_THRESHOLDS	<p>The SNI threshold values themselves. Note that there must be as many values as stated in parameter SNITABLE_N_VALUES above. Note that these thresholds are used to check correlation values; therefore, these must be in the range (-1, 1) and sorted in ascending order.</p> <p>See the explanations in the current section for more details about tables.</p>	List of floating point values. Each value in the range (-1, 1). List sorted in ascending order.

Label	Description	Comments
QITABLE_CLASSES	<p>The Quality Index (QI) class table.</p> <p>This table must have a total of $(\text{TNITABLE_N_VALUES} + 1) * (\text{SNITABLE_N_VALUES} + 1)$ values.</p> <p>See the explanations in the current section for more details about tables.</p>	List of integer values.

Table 1: The available options in an ADAFinder defaults file

For a full understanding of the parameters in Table 1, please refer to [RD1].

The parameters describing tables (temporal & spatial noise indices, quality index class) deserve a special discussion.

The temporal and spatial noise index tables work exactly in the same way. Therefore, only the first case (spatial noise index table) will be described. The rationale behind the explanation below may then be applied literally to the temporal noise index table.

The temporal noise index is based on the value of the autocorrelation of the deformation time series. The possible range for the autocorrelation value (-1, 1) is split in several segments, each of these defined by a threshold value. Thus, if for instance x thresholds are defined, this implies $x+1$ segments in the autocorrelation range.

Assuming that the threshold values are -0.4, 0.5 and 0.9 (three thresholds) then four segments exist: [-1, -0.4], (-0.4, 0.5], (0.5, 0.9], (0.9, 1]. Each of these segments correspond to a temporal noise class. The identifier of these classes starts at 1 for the first segment and ends at $x + 1$ (the number of segments), 4 in this example. Thus, the classes would be 1,2,3 and 4.

The parameters `TNITABLE_N_VALUES` and `TNITABLE_THRESHOLDS` are used to define the thresholds, and, indirectly, the segments and classes used to classify the spatial noise. In the example above, `TNITABLE_N_VALUES` should be set to 3 (there are 3 thresholds) and the list of values in `TNITABLE_THRESHOLDS` should be -0.4, 0.5 and 0.9. The correct way to write this in the defaults options file would be:

```
TNITABLE_N_VALUES      = 3
TNITABLE_THRESHOLDS  = -0.4 0.5 0.9
```

The same discussion applies to `SNITABLE_N_VALUES` and `SNITABLE_THRESHOLDS`.

Finally, the quality index class (QI) table explains how the combination of the classes derived from the spatial and temporal quality indices produce another class classification. For clarity reasons, an example will be used to describe how the QI table is built and stored in the defaults file.

The assumption is that there exist four classes for the temporal noise index (namely, 1, 2, 3 and 4) and only three for the spatial noise index (that is, 1, 2 and 3). Then, the QI table must be arranged as shown in Table 2. Note that in this table the spatial noise classes are arranged from left to right while

the temporal ones are deployed from top to bottom.

In this example, up to $3 \times 4 = 12$ QI classes are possible. At any rate, the number of classes is decided by the user of the application; however, the maximum number of these is limited by the product of the number of spatial and temporal noise classes. Of course, it is also the user who assigns a QI class to each combination of spatial and temporal classes.

		Spatial Noise Class		
		1	2	3
Temporal Noise Class	1			
	2			
	3			
	4			

Table 2: Arrangement of a QI table

In this example, 6 QI classes will be assumed. Table 3 shows how the user decided to assign these classes depending on the temporal and spatial ones.

		Spatial Noise Class		
		1	2	3
Temporal Noise Class	1	1	2	3
	2	2	3	4
	3	3	4	5
	4	4	5	6

Table 3: Example of QI classes assignment

This table must be written in the defaults file. To do it, parameter `QITABLE_CLASSES` must be used. The class data in the matrix must be written row-wise, starting at the top of the table and then going downwards. The example below shows how Table 3 must be written to the defaults file:

```
QITABLE_CLASSES = 1 2 3 2 3 4 3 4 5 4 5 6
```

Note that the number of elements (12) in `QITABLE_CLASSES` is directly derived from the product `TNITABLE_N_VALUES` \times `SNITABLE_N_VALUES`, so there is no need for a parameter in the defaults file specifying its size.

A full example of an ADAFinder defaults file is shown below. Note that no comment lines have been written in this example.

```

INPUT_SHAPEFILE_POINTS           = input_points.shp
INPUT_SHAPEFILE_POINTS_READMAP    = input_points_readmap.op
INPUT_SHAPEFILE_BOUNDARIES       = -
OUTPUT_SHAPEFILE_ADAS            = output_adas.shp
OUTPUT_SHAPEFILE_POINTS          = output_points.shp
ISOLATION_DISTANCE                 = 40.0
ISOLATION_CLUSTER_SIZE             = 2
STDEV_FACTOR                        = 1.5
THRESHOLD_FOR_VELOCITY_CLASS_1    = 10.0
ADA_MINIMUM_CLUSTER_SIZE           = 5
ADA_RADIUS                          = 26.0
ADA_SHIFT_TO_CM                    = 4
ADA_DISPLAY_MODE                   = 1
ADA_CIRCLE_STEPS                   = 40
N_VALUES_MEAN_DEFORMATION           = 4
POINT_SUBSET_TO_WRITE              = 3
TNITABLE_N_VALUES                  = 3
TNITABLE_THRESHOLDS                = 0.53 0.70 0.84
SNITABLE_N_VALUES                  = 3
SNITABLE_THRESHOLDS                = 0.53 0.70 0.84
QITABLE_CLASSES                    = 1 1 2 4 1 2 3 4 2 3 3 4 4 4 4 4

```

Note that in the example above there are a few extra options (those that have been **boldfaced**) that have not been discussed yet.

3.3.2 The extra, batch options

Besides the options discussed in the previous section, there are a few more that are ignored¹ when the ADAFinder application is started in **interactive** (GUI-based) mode. These are the batch options, targeted at defining some extra parameters when the tool is run in batch (command line) mode. The extra batch options – see Table 4 – define, essentially, the names of the several input and output files intervening in the process. Note that these files are provided manually by the user when using the GUI version of the application. When it is run in batch mode, the defaults file is the way to specify their names.

¹ Even if these options are not loaded in the interactive (GUI-based) mode, they must be present in the default files for completeness reasons. Their values may be nonsense, but they must be correctly written. For instance, an options pointing to an input file must actually point to a well-written input file name, although this file didn't exist.

Label	Description	Comments
INPUT_SHAPEFILE_POINTS	Name of the input file containing the points to process. Item 1.	It must be an Esri shapefile containing points.
INPUT_SHAPEFILE_BOUNDARIES	This file is optional . It must contain either the name of the input file defining the boundaries of the areas of interest or the character “-“ (quotes not included). In the second case, the dash indicates that there is no boundaries file to clip the input data. Item 3.	It must be an Esri shapefile containing one or more polygons (the boundaries).
OUTPUT_SHAPEFILE_ADAS	Name of the output file containing the ADAs. Item 4.	Esri shapefile containing one polygon per ADA.
OUTPUT_SHAPEFILE_POINTS	Name of the output file containing the points. Item 5.	Esri shapefile containing the filtered points.

Table 4: The batch options in an ADAFinder defaults file

3.4 The read / write map file

The points that ADAFinder will process must be stored in an Esri shapefile. Although the shapefiles (.shp, .shx) containing points are always the same, this is not so for the attribute table (.dbf file) accompanying the former.

The set of attributes in the .dbf file may vary. ADAFinder needs a minimal set of attributes to be present in order to carry out its work. If at least one of these attributes does not exist, ADAFinder may not use the file.

The mandatory attributes are:

- X coordinate (easting).
- Y coordinate (northing).
- Velocity (in mm / year).
- Deformation time series (mm / year).

The X & Y coordinates must be projected coordinates. The number of values in the deformation may vary from one file to other, but must be the same for all the points in the same file.

Additionally, two more optional fields may be present for each point:

- Lambda coordinate (deg) and
- Fi coordinate (deg).

Assuming that a points shapefile’s attribute table contains at least the mandatory ones (and, eventually, the optional ones too) the positions where these appear may vary between different

datasets (because, for instance, have been produced by several providers in different projects). Obviously, this variability poses a problem that must be solved if ADAFinder must be able to read these different shapefiles.

This is why the tool includes a mechanism to define where to find the mandatory (and, eventually optional) fields listed above. By providing such information, ADAfinder will be able to read many different point shapefiles with no changes in the software itself.

The way to do this is to assign, to each field, the position in the shapefile where the field sought may be found. For instance, assuming a shapefile whose attribute table is defined as follows:

```
identifier X Y lambda fi gamma row column velocity def01 def02 ... def20
```

then the positions for the X, Y, velocity and deformation attributes would be, respectively, 1, 2, 8 and 9 (note that the first column, identifier in the example, uses index 0). Since the number of values in the deformation time series is also user-definable, it is necessary to state that such number; in this example, it is 20. The positions of the optional lambda and fi attributes would be 3 & 4.

To define such positions, a simple options file, whose syntax is defined in [RD3], is used (in fact, is exactly the same kind of file used by the defaults file).

Finally, when writing the output file with points the names of the (variable) number of fields storing the values of the time series with deformations must be defined. These fields may have any name(s) the user wish to assign, although the usual practice is to label these including the date of the measurement.

For instance, a series consisting of four values measured the 30th of April of 2010, 2011, 2012 and 2013 could use the strings “20100430”, “20110430”, “20120430” and “20130430” to label the corresponding deformation fields in the output points file, where the pattern to build these identifier adheres to the schema “YYYYMMDD”, being YYYY the year, MM the month and DD the day.

In fact, the usual convention prepends a “D” in front of these dates to build the field names – where the literal letter “D” stands for “deformation”; thus, the previous example would mutate to “D20100430”, “D20110430”, “D20120430” and “D20130430” respectively.

Note, however, that this is an accepted convention and not a rule; the user of ADAfinder may decide to name these fields at his / her will.

What is true is that it is necessary to provide names for the deformation fields. The way to do it is the label named OUTPUT_TS_FIELD_NAMES.

All the labels to use (for both read and write maps) are defined in Table 5.

Label	Description	Comments
POSITION_X	Position (column number) of the X projected coordinate.	Mandatory, 0.based index.
POSITION_Y	Position (column number) of the Y projected coordinate.	Mandatory, 0.based index.
POSITION_VELOCITY	Position (column number) of the velocity.	Mandatory, 0.based index.
POSITION_TIME_SERIES	Position (column number) of the first value of the deformation time series.	Mandatory, 0.based index.
N_VALUES_TIME_SERIES	Number of values in the deformation time series.	Mandatory, 0.based index.
HAVE_LAMBDA_FI	Flag indicating whether the optional lambda and fi coordinates are available.	Mandatory. Either 0 (no lambda and fi exist) or 1 (lambda and fi are available).
POSITION_LAMBDA	Position (column number) of the lambda (longitude) coordinate.	Optional if HAVE_LAMBDA_FI is zero. Mandatory otherwise. 0 based index.
POSITION_FI	Position (column number) of the fi (latitude) coordinate.	Optional if HAVE_LAMBDA_FI is zero. Mandatory otherwise. 0 based index.
OUTPUT_TS_FIELD_NAMES	Names of the fields (in the output points file) containing the time series with the deformations.	Mandatory. List of string values. As many values in the list as indicated by label N_VALUES_TIME_SERIES.

Table 5: The attributes in the read map file

The following is an actual example of a read map file, including a few comments.

```
# =====
#
# This is and ADAfinder's read map file.
#
# =====

# -----
#
# MANDATORY attributes. Positions (0 is the origin) for:
# - X
# - Y
# - Velocity
# - First value of the deformation time series.
#
# Additionally, number of values in the deformation time series.
#
# -----
```

```

POSITION_X          = 5
POSITION_Y          = 6
POSITION_VELOCITY   = 9
POSITION_TIME_SERIES = 11
N_VALUES_TIME_SERIES = 26

# -----
#
# OPTIONAL attributes. Flag indicating the availability of the lambda and
# fi coordinates and the positions of the coordinates themselves.
#
# -----

HAVE_LAMBDA_FI      = 1
POSITION_LAMBDA     = 4
POSITION_FI         = 3

# -----
#
# The names for each of the fields containing a deformation series value.
#
# Remember that the number of values in this option must match the value
# given in label N_VALUES_TIME_SERIES.
#
# -----

OUTPUT_TS_FIELD_NAMES = D20141105 D20141117 D20141129 D20141211 D20141223 \
                        D20150104 D20150116 D20150128 D20150209 D20150221 \
                        D20150305 D20150317 D20150329 D20150609 D20150703 \
                        D20150715 D20150727 D20150808 D20150820 D20150901 \
                        D20150913 D20150925 D20151007 D20151019 D20151031 \
                        D20151112

# End of file.

```

3.5 The input (data) files

ADAFinder's input files are (up to) four:

1. The defaults file,
2. the read map file.
3. the PS (points) file and
4. an optional boundaries file, setting the AOI to actually process.

The first two (defaults and read map files) have been thoroughly described in sections 2.3 and 2.4 respectively. Next sections talk about the PS and boundaries files.

3.5.1 The PS (points) file

Input PS (point) files must be an Esri shapefile (see [RD2]). Its type, since it will contain points, must be 1 (“Point”). The coordinates of the points must be projected (x & y, that is, easting & northing).

Points in the input shapefile must have a minimum set of attributes for ADAFinder being able to carry out its work. These attributes, stored in the attribute table (.dbf file in the shapefile file set) are:

- X coordinate (easting).
- Y coordinate (northing).
- Velocity (in mm / year).
- Deformation time series (mm / year).

Additionally, the following optional attributes may also be present, but their absence will not prevent the normal operation of the tool.

- Lambda coordinate (deg) and
- Fi coordinate (deg).

Both the mandatory and optional attributes may appear in any order in the attribute table. The read file map (see Section 2.4) will be used by ADAFinder to ascertain the actual position of every attribute in the input .dbf file.

3.5.2 The optional boundaries file

Again, the input boundaries file is an Esri shapefile (see, once more, [RD2]). In this case, however, it must contain one or more polygons defining the AOIs where the actual process must take place. This means that the type of the shapefile must be 5 (“Polygon”).

The coordinates defining the aforementioned polygons must be projected (that is, x or easting and y or northing), using the same coordinate reference frame (CRF) than the projected coordinates used in the PS file.

No table of attributes (.dbf file) is expected. If present, ADAFinder will ignore it.

3.6 The output files

ADAFinder creates two output files: the first one will store the polygons defining the surface covered by the ADAs; the second one will include those points from the input PS file that have passed a set of filters. Both are shapefiles, so refer to [RD2] for details on the format of these Esri files.

3.6.1 The ADA file

The output ADA file is an Esri shapefile containing polygons, that is, its type is 5 (“Polygon”). There are as many polygons in the file as ADAs has been detected by the tool.

The coordinates used to define the polygons are projected (x & y, that is, easting & northing) and refer to the same CRF than the one used in the input points file.

The fields included in the attribute table (related .dbf file) are shown in Table 6.

Field	Description
ADA_ID	ADA identifier. This is a sequential number starting at 1 for the first ADA found. Used to link points in the output points file with the ADAs in this one.
N_APS	Number of active points that are included in the ADA.
X_MEAN	Mean X coordinate for the ADA. Computed as the mean of the X coordinates of the active points included in it.
Y_MEAN	Mean Y coordinate for the ADA. Computed as the mean of the Y coordinates of the active points included in it.
VEL_MIN	Absolute value of the minimum velocity of the active points included in the ADA.
VEL_MEAN	Absolute value of the mean velocity of the active points included in the ADA.
VEL_MAX	Absolute value of the maximum velocity of the active points included in the ADA.
VEL_CLASS	Velocity class. See [RD1] for more information about the possible values and their meanings.
DEF_MEAN	Mean of the last <i>n</i> values of the deformation time series of the active points included in the ADA. The value <i>n</i> is defined by means of the label <code>N_VALUES_MEAN_DEFORMATION</code> in the tool's defaults file (see Section 2.3.1 for details).
TNI_CLASS	Temporal Noise Index Class. The number (and labels) of the classes depend on the number of temporal noise thresholds set by label <code>TNITABLE_THRESHOLDS</code> in the application's defaults file.
SNI_CLASS	Spatial Noise Index Class. The number (and labels) of the classes depend on the number of spatial noise thresholds set by label <code>SNITABLE_THRESHOLDS</code> in the application's defaults file.
QI_CLASS	Quality Index Class. The values of the classes are defined by label <code>QITABLE_CLASSES</code> in the tool's defaults file.

Table 6: The attributes in the ADA shapefile

3.6.2 The points file

The output points file is, as usual, an Esri shapefile (type 1, "Point").

A point included in the input file will be saved in the output file providing that it passes the following filters:

- It must be inside any of the clipping polygons (if a boundaries input shapefile has been provided, otherwise no clipping takes place and all points pass this filter),
- it must not be isolated (no neighbours within the isolation distance set by the user) and
- it cannot belong to a too small cluster (the number of points found within the isolation distance is not big enough).

On the contrary, these points do not need, for instance, neither to pass the stability (standard deviation) filter nor be part of an ADA.

The coordinates of the points are projected ones (x & y, that is, easting and northing) and refer to the same CRF than the one used by the input points file.

The fields included in the attribute table (.dbf file) are described in Table 7.

Field	Description
E	Easting (x coordinate).
N	Northing (y coordinate)
LAMBDA	Longitude. WARNING: this field is optional, and will only appear when it is present in the attribute table of the input points file.
FI	Latitude. WARNING: this field is optional, and will only appear when it is present in the attribute table of the input points file.
ADA_ID	Identifier of the ADA the point belongs to. When a point belongs to no ADA, this identifier is set to -1. On the contrary, it is set to an strict positive value (1 or greater) when the point belongs to one of the ADAs found.
DEF_MEAN	Mean value of the deformation time series.
Deformation value X.	There are as many fields like this as values in the deformation time series. That is, if the time series include, for instance, 20 values, then there will be 20 of these fields. Each field contains one of the values in the deformation time series. The names of these files are provided in the read / write map file, using label OUTPUT_TS_FIELD_NAMES (see section 2.4 and Table 5)

Table 7: The attributes in the output points shapefile

3.7 Installing the tools

To install the tool please refer to [RD4].

3.8 Using the tool

The tool has two incarnations that must be used in different ways.

To use **the GUI version** it is enough to start the application by whatever means at hand (start menu, desktop shortcut...), fill the form shown by the tool (see Section 2.2 and Figure 1) and click on the “Go” button. Once the process is over, a message box warning about the result (success, failure and its reason) will show up. The process may be interrupted at any moment by closing the GUI window.

Since some of the options may not be changed using the GUI, it is important to remember that these may be changed editing the application’s default file located in the same folder than the tool’s executable. In the same line, the read map file is located in the same place; this file must be modified to adapt the tool to the specific characteristics of the input point shapefiles to use. See sections 2.3 for details about the contents of defaults (options) files.

To use the **batch version**, a command line must be opened. To start it, type the following command:

```
adafinder_cmd <defaults_file>
```

where <defaults_file> stands for the full path and name of the defaults file. See again section 2.3

for details.

Note that in the case of the command line version, the defaults file does not need to stay in the application's directory. This is so because it is the way to control the application and it is much more convenient let the users manage it at his / her will.

Note also that if the tool is not located in the current (windows) path, then it will be necessary to prepend the path to the directory where the tool is stored (or to add the path to the tool to the Windows PATH environment variable).

4 Conclusions

ADAFinder is a tool that computes Active Deformation Areas using the methodology reported by [RD1]. It is a fast tool: $O(k \cdot \log(\#points))$; it is able to process more than 900.000 points in less than five minutes in a regular computer (i5, 8Mb RAM). This makes ADAFinder a truly production tool. It may be integrated as a toolbox (ArcGIS) or plugin (QGIS). Moreover, the kind of inputs and outputs managed may be obtained from GIS tools as well, being therefore no obstacle in a “production” workflow based on GIS tools. The same may be said about the outputs.

The application is configurable, this meaning that it may be adapted to read different kind of input shapefiles, providing these include the minimum set of attributes needed to operate.

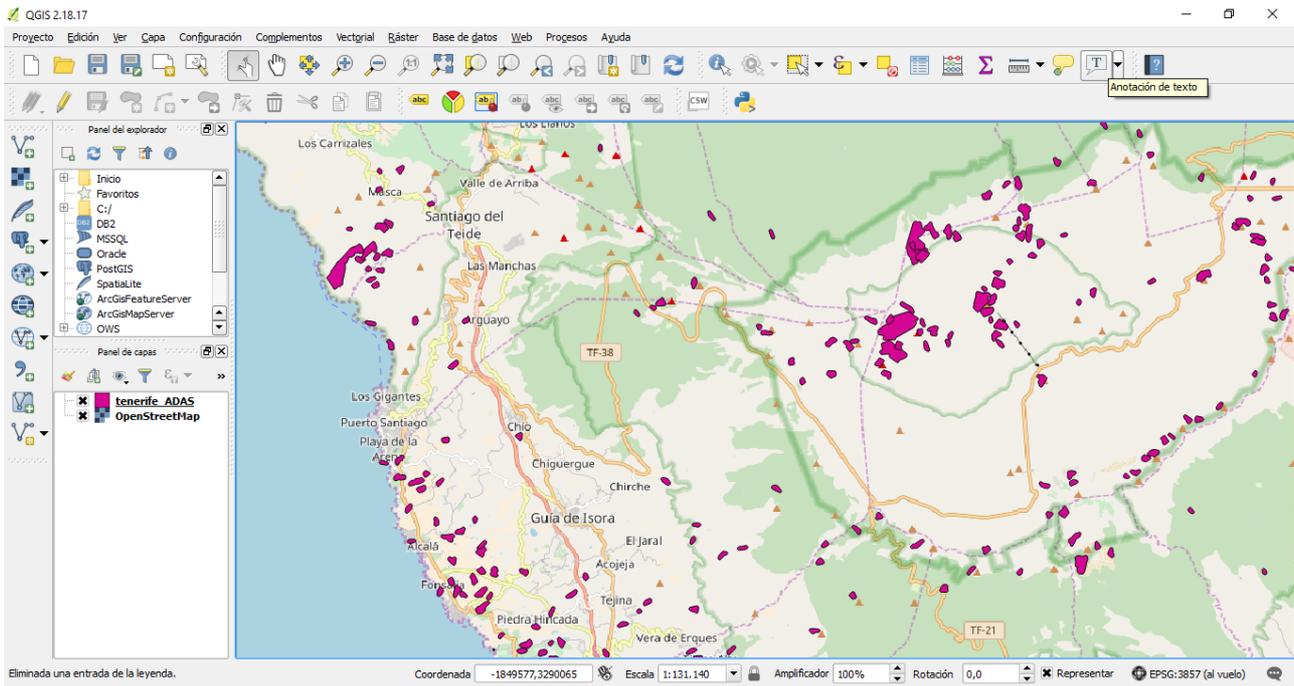


Figure 3: QGIS showing a map with ADAs (purple areas)

Reference Documents

- [RD1] Barra, A.; Solari, L.; Béjar-Pizarro, M.; Monserrat, O.; Bianchini, S.; Herrera, G.; Crosetto, M.; Sarro, R.; González-Alonso, E.; Mateos, R.M.; Ligüerzana, S.; López, C.; Moretti, S. *A Methodology to Detect and Update Active Deformation Areas Based on Sentinel-1 SAR Images*. *Remote Sensing*. 2017, 9, 1002.
- [RD2] Esri. Esri shapefile description. 1999. <https://www.esri.com/library/whitepapers/pdfs/shapefile.pdf>. Last visited: March 13th, 2018.
- [RD3] Navarro, J.; Cuevas, M. MOMIT technical note – Simple options file: technical description.
- [RD4] Navarro, J.; Cuevas, M. MOMIT technical note – Installing the software.

End of the document
