

USING TAILORED GRAPHICAL TOOLS TO IMPROVE THE QUALITY OF DISPLACEMENT MAPS AT CTTC: THE VETOOLS

J. A. Navarro^{1*}, A. Barra¹, O. Monserrat¹, M. Crosetto¹

¹ Centre Tecnològic de Telecomunicacions de Catalunya (CTTC/CERCA), Av. Carl Friedrich Gauss 7, 08860 Castelldefels, Spain -
(jose.navarro, anna.barra, oriol.monserrat, michele.crosetto)@cttc.cat

KEY WORDS: DInSAR, ground deformation maps, graphical tools, geoprocessing

ABSTRACT:

The Geomatics Division of the Centre Tecnològic de Telecomunicacions de Catalunya (CTTC) has been producing terrain displacement maps for more than 15 years using the PSIG software chain, developed entirely by the members of the aforesaid group. PSIG has reached a high level of maturity, being highly automated but also offering the user the possibility to fine-tune the set of critical parameters controlling the process. However, large areas with a high level of diversity often pose problems in finding a good quality solution using a single set of parameters. Improving the quality of the final terrain displacement map means being able to process very local and critical areas using specific sets of parameters; however, identifying such areas is a difficult process without suitable analysis tools. The VETools, a new software project still under development, but very close to its completion, target this problem. With the VETools it is possible to visualize the results produced by a previous PSIG global processing, analysing their quality, thus making possible to identify the local, critical areas, allowing the user to interactively experiment with specific sets of parameters for these areas, reprocessing them and reviewing the local results as many times as desired and, finally, merging all of them in a single, unique solution whose level of quality is appropriate for the whole area of interest. This work presents the current state of development of the VETools, describing their features and the challenges overcome.

1. INTRODUCTION

The Geomatics Division (GD) of the Centre Tecnològic de Telecomunicacions de Catalunya (CTTC) (<http://geomatics.cttc.es/remote-sensing/>) has a long experience in generating terrain displacement maps from satellite SAR Differential Interferometry (DInSAR) techniques.

In the frame of this research line, the Persistent Scatterer Interferometry chain of the Geomatics division (PSIG) (Devanthery et al., 2014) has been developed. The PSIG processing chain and software are under continuous improvement, and adaptation for a wide range of related applications. Nowadays the GD is also focusing in developing complementary tools to improve (1) the practical exploitation of the PSI derived displacement maps for any (even not expert) final users (Barra et al., 2017), (Tomás et al., 2019), (Navarro et al., 2020), and (2) the “user experience” using the PSIG chain itself: the VETools, which are the subject of this work.

The PSIG processing chain includes several steps to, starting from a set of interferograms, estimate the velocity map and the deformation time series over selected points (representing targets with stable back scattering characteristics). The PSIG chain has reached a high level of automation, but at the same time it maintains the flexibility of a “white box” approach, allowing the user to tune every step or parameter depending on the case of study and to have an inspection of all the internal-intermediate outputs. The counterpart is the difficulty in the full exploitation of this flexible system. The idea behind the development of the VETools was to enhance the capacity of the user to improve the quality of the results thanks to a visual and interactive approach to the processing. VETools allows to generate and integrate local results obtained over several critical zones of the target area (thus having a local tuning of the parameters) and to visualize, check, and edit the results. These tools are very useful processing areas that are difficult from the point of view of the DInSAR technique application, also

allowing to connect separate zones within a unique solution. The actual version of VETools includes only the PSIG related to the estimation of the displacement velocities. In this work, after a short description of the main steps for the velocity estimation of PSIG (section 2), the tools will be presented in all their aspects (section 3).

2. THE PSIG SOFTWARE CHAIN

This section describes the key features of the PSIG chain to generate the velocity map. Its inputs include a stack of N co-registered SAR images, the amplitude dispersion (DA), and M wrapped interferograms, with $M \gg N$. We refer to (Biescas et al., 2007) and (Devanthery et al., 2014), for deeper explanation of the velocity calculations.

The main steps are resumed below.

- Points selection: selection of the pixels that guarantee a good quality of the input phases in the stacks of used interferograms.
- Pixel edges calculation: the selected pixels are connected by edges, and the phase difference of each edge is calculated for each wrapped interferogram. Based on the calculated differential phases (observations), the differential annual linear velocity (linear model) and the topographic error – see (Biescas et al., 2007) – are estimated over each edge. Each edge estimation has associated the so called γ (gamma) factor that assess the goodness of fit of the model with the given set of wrapped phase observations.
- Reordination: it consists of the selection of the edges, to be used in the velocity estimation, based on the gamma values. Depending on the gamma threshold, some points will be disconnected from the others and

* Corresponding author: J. A. Navarro, jose.navarro@cttc.cat

thus those points will not be present in the final velocity map. It can happen that there are more groups of interconnected points, but without connections among them, excluding the possibility of having a unique solution.

- Integration: the last step consists in the reconstruction of the velocity for each selected pixel integrating the differential velocities obtained over the edges. The same is done for the topographic errors. This step is possible only over a connected group of points. If there are more than one group, more integrations are needed.

Within these main steps, several parameters for each step can be chosen by the expert user, knowing that each decision will influence the final result in terms of: i. Quality of the velocity estimation (reliability or noise level), ii. Spatial coverage and density of measurement points and iii. Connection between groups of points, i.e., obtaining a unique result over the whole area. The main difficulty can be found where the area to be processed is wide and variegated; in this case finding a unique solution in terms of parameters setting, can reduce the quality of the results over the whole area. On the contrary, the idea of the VETools was to make a more flexible selection of the parameters, based on a graphical visualization of the results (velocities, edges, groups, etc.), to be adapted to very local and critical areas, and to be integrated in the global result in an automatic way. VETools improve the user control and facilitate the approach to reach better results.

3. THE VETOOLS

The VETools (Vertex & Edge Tools) consist of two applications, VEReady and VEGrapher. They are targeted at helping the users of the PSIG software chain to analyze its results and decide on how to improve these by allowing the reprocessing of critical local areas, being possible to tune the parameters controlling such reprocessing.

The VETools have been designed to work according to the following workflow:

- The PSIG software chain is run, which produces some results after processing the area of interest.
- Then, VEReady takes these results and reorganizes them in such a way that VEGrapher may use them efficiently. This step is fully automated and requires no user intelligence. However, it is crucial to make possible the task of VEGrapher later.
- After that, the user may use VEGrapher to review the results, identifying troublesome areas. With VEGrapher it is then possible to reprocess these zones, thus obtaining additional local results. Users may reprocess as many areas as desired.
- VEGrapher may be used to merge the whole set of inputs (the original ones plus those produced by the reprocessing of local areas).
- Finally, the data set created in the previous step will be used as the input of a final run of the PSIG chain combining everything in a unique solution.

Although very close to its completion, the VETools are still under development. The following sections describe the current

state of the VETools, explaining their characteristics, features, and challenges.

3.1 VEReady

VEReady takes care of changing the way the information produced by the PSIG software chain is organized to facilitate the task of VEGrapher. Since such data may comprehend millions of PS (aka, the points) and still much more edges, providing a clever and fast way to manage them is vital not only to make possible a tool like VEGrapher, but also to make it productive.

VEReady takes the following inputs (which are the outputs of the first run of the PSIG software chain):

- Edges: a text file including, in each line the column and row of the starting and ending points for the edge, the identifiers of these two points, and then three attributes characterizing the edge (velocity, topographic error and the gamma quality parameter).
- Attributes for the points (velocity, topographic error, and group identifier). Several non-indexed, plain text files contain the values for such attributes in the form column, row, value.
- Background image. The original SAR image, converted to some visible format (such as TIFF or JPG), to back up the visualization of the area of interest in VEGrapher.

Then, it produces the following outputs:

- A spatial index for the points. A spatial index is a specialized indexing structure where the indexing key is the spatial location (coordinates) of the objects indexed. The type of searches on a spatial index is the set of spatial queries, such as "find the objects whose distance to some other object is less than some given distance" or "find the points that fall inside some area". Its purpose is to be able to identify – quickly, thanks to its internal indices – what are the points that are included the rectangle defining VEGrapher's viewport (see section 3.3.1). Besides their coordinates (the key for the index) the velocity, topographic error and group attributes are stored for the points in this data structure.
- The edges. A binary file including the same information that the input (text) edges file is also produced. The main difference is that in this case, the information related to each edge is easily, quickly, and randomly accessible thanks to an extra edge identifier that allows for a quick computation of the position where such edge is stored.
- The points to edges database. A database that relates the points to the edges that originate from them; that is, once that the identifier of some point is known, it is possible to obtain immediately the set of identifiers of the edges starting at this very point.
- The background image tiles. The background image is divided into small regular tiles. The idea is to make possible the visualization (in VEGrapher) of very big images by loading and displaying only those parts of the scene that are visible in the current viewport (see section 3.3.2).

Because of its importance concerning performance, the spatial index has been implemented by means of the libspatialindex library (Hadjieleftheriou, 2019), and the points to edges database relies on the levelDB library (LevelDB development team, 2021).

3.2 VEGrapher

VEGrapher is a visual tool targeted at helping the SAR specialist to review the results of the PSIG software chain and, if necessary, to start corrective actions to solve the problems that might be detected.

This is done by analyzing visually and interactively the results obtained from a first, initial run of the PSIG software chain – results that have been reorganized by VEReady. This analysis process will eventually lead to the identification of critical areas that, according to the criterion of the expert, could be improved by means of their local reprocessing. Such reprocessing implies running (an embedded version of) the PSIG software chain but constraining the zone to work with to the aforesaid critical areas.

The user may then review the results of reprocessing such areas, and has the opportunity to reject, accept and even reprocess them again as many times as considered convenient until a satisfactory set of local solutions is obtained. Finally, VEGrapher lets the user merge the whole set of results (the original ones coming from the first run of the PSIG software chain plus the local results produced by reprocessing the local, critical areas) to obtain, later, an integrated solution.

Developing a tool of this type implies solving a series of challenges and offering a set of functionalities that allow the user to carry out their work with ease. More specifically, it is necessary to manage big amounts of data, embed the PSIG software chain inside VEGrapher to offer the user an ergonomic, non-disruptive experience. Moreover, tools to help the user to identify and / or correct problems must be included in a tool like this.

VEGrapher has been implemented (as well as VEReady) in C++ for performance reasons. The graphical user interface, being quite complex, relies on the portable Qt framework (The Qt company, 2020.).

The next subsections describe the most important features included in VEGrapher as well as the challenges that the implementation of a tool like this had to face.

3.2.1 Point colorization: Users analysing the data set will sometimes need to rely on one or more of the attributes related to the points. The attributes managed by the VETools are (1) the velocity of the points, (2) their topographic error and (3) the group they belong to. The specific attribute to use will depend on the kind of problem analysed by the user.

Having a visual feedback of such attributes may be a helpful aid to decide how to proceed. More specifically, VEGrapher let users to colorize points according to the values of any of the attributes above. The colour tables to paint the points may be designed using a built-in function in the application. Since the colour tables may be saved to disk, they can be reused as many times as necessary. Figure 1 depicts how a velocity & topographic error colour table is built; both the velocity intervals and the colour selected are shown. Figure 2 shows the same segment of an image using the default colour for points (up, in red) and then using the velocity-based colour table (down, green and yellow).

Users are free to change between the four kinds of coloring schemas (no coloring, velocity, topographic error, groups) at their will.

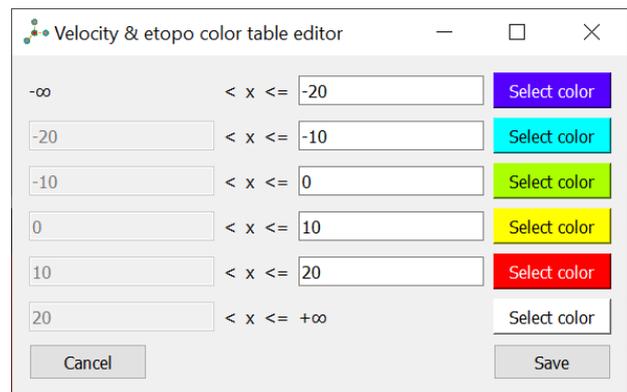


Figure 1: Building a colour table.

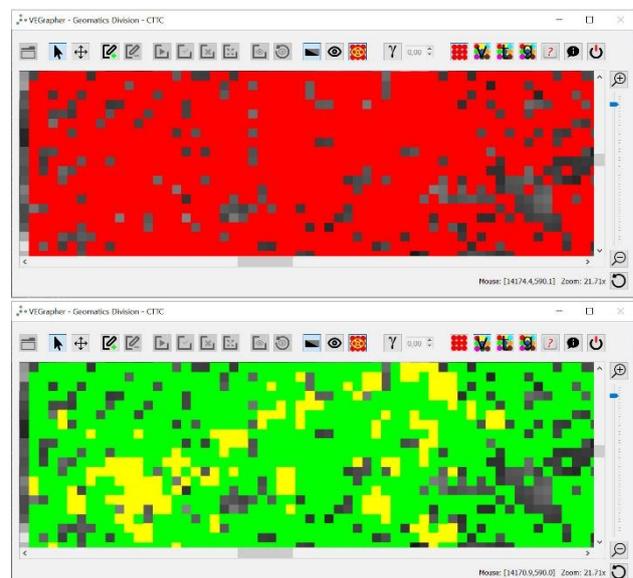


Figure 2: Points shown using the default colouring schema (up, red) and displayed with a velocity colour table (down, green & yellow).

3.2.2 Selecting and filtering edges: If the number of points to display is big, the number of edges is still much bigger. Therefore, these are only shown when the user clicks over one of the points.

The quality of the connection represented by the edge is measured by the so-called gamma parameter, whose value ranges between 0 and 1. Edges with a low gamma value represent poor connections between the points making the edge; the opposite, high values of gamma, correspond to good connections.

VEGrapher let users analyse the quality of the connections to ascertain whether some specific area should be reprocessed to obtain better results. To do this, a real-time filter based on the value of gamma may be used.

When a point is selected, and the edges starting there are shown, it is possible to activate the gamma filter and, in real time, see how these are discarded or included when the value of this parameter is changed. Figure 3 shows how the number of edges is reduced when the gamma filter is applied.

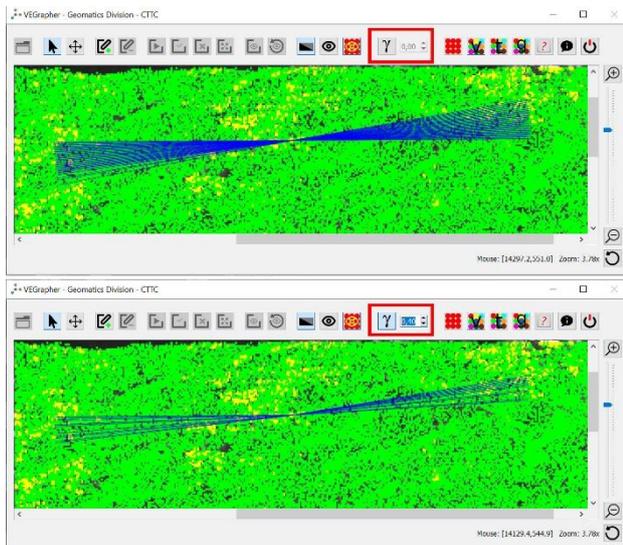


Figure 3: Gamma-filtering edges: the filter is inactive (up) and active (down) with a gamma value 0.4.

3.2.3 Local reprocessing: Once that the SAR specialist has identified problems in one or more areas, it is possible to use VEGrapher to reprocess them, by launching the PSIG software chain directly from this tool.

The procedure to follow is quite simple; first, the suspicious area is selected dynamically with the mouse, defining a rectangular zone that covers it. Then, clicking on the appropriate button in the top toolbar, a dialog box including the parameters controlling the behavior of the several applications making the PSIG software chain is shown. At this point, users can modify these parameters according to the kind of reprocessing they intend to do; once satisfied, clicking on a confirmation button will start the process. See Figure 4.

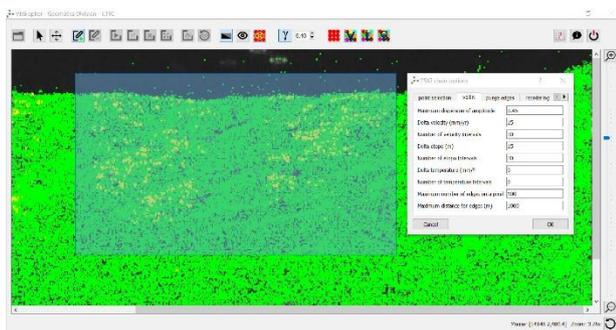


Figure 4: Local reprocessing - selecting the critical area and setting the control parameters.

At this point, a log window is opened (see Figure 5). The output (log) of every tool involved in the process is shown there, so once the job is over, the user may perform a first check of the quality of the results. In fact, it is possible to reject the results of this process using only this log data. Should this happen, no changes would be applied to the data set.

If this step is – just preliminary – accepted, the user is given the opportunity to review the results, this time graphically. This means that the display changes substantially, showing only the data belonging to the area that has just been reprocessed.

Figure 6 depicts such situation. There, it can be observed that the area just reprocessed is highlighted with a semi-transparent greenish rectangle. Only the points produced by the reprocessing are shown to avoid confusion (in this case, using the default coloring scheme).

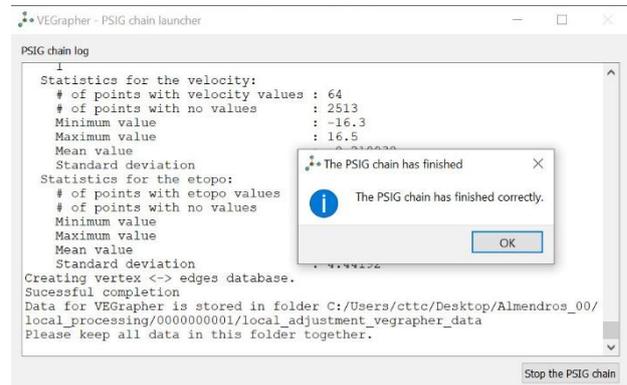


Figure 5: The embedded PSIG software chain log window.

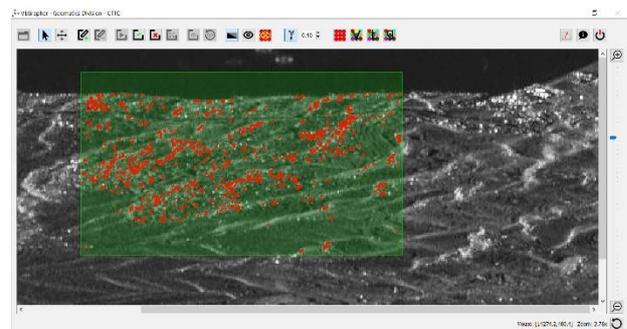


Figure 6: Reviewing the results obtained after reprocessing a local area.

Now, the user may focus on analyzing the area using the same tools described above – as for instance, gamma-based edge filtering – that were applied to analyze the whole scene. After that, and once the analysis work for this specific zone is finished, it will be possible, once more, to either accept or reject the results. A rejection will discard the results of the reprocessing. On the contrary, accepting the results will preserve these for a later, final merging process (see section 3.2.4). The reprocessed area is then highlighted with a semi-transparent, greenish rectangle, as a feedback to remind the user about what has already been processed, especially because the whole workflow just described may be applied to an unlimited number of areas – as many as conflictive ones are detected by the expert.

Nonetheless, and even when the users have decided to accept some reprocessed area(s), they still have the possibility to review and reprocess these as many times as desired – thus increasing the flexibility of the tool. Just clicking on the appropriate button in the top toolbar will open a window listing the whole set of reprocessed areas (see Figure 7).

In Figure 7, three areas – represented by the greenish rectangles – have already been a reprocessed. The small window lists all of them; the rightmost one has been selected – this may be noticed because it is shown in a darker shade of green. If, by whatever the reason, the expert decides to relaunch the PSIG software chain to reprocess again the zone, clicking the appropriate button in that small window will be enough.

The steps to reprocess – again – the selected area are the ones already described in this section. The only difference concerns the final acceptance or rejection: if the users decide to accept the new results, then the old ones will be discarded; on the contrary, a rejection implies that the former results are restored, while the new ones are those that will be destroyed.

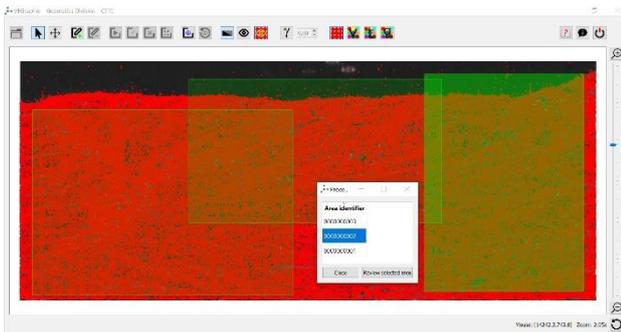


Figure 7: Revisiting areas that have already been processed.

3.2.4 Final merging: The local reprocessing steps described in the previous section serve only to improve the quality of the results for a set of small areas in the global scene. That is, small patches are applied to the global picture.

To finish the process, the original set of results obtained when running the PSIG software chain for the first time – those that were the input of the VEReady tool, see section 3.1 above – must be integrated with those obtained by means of reprocessing the local areas.

Such integration is a two-step process; first, all data sets (initial plus local) must be harmonized to make possible to merge them; then, the PSIG chain must be run once more time to obtain a unique solution encompassing all results. The first step, harmonization, is performed by VEGrapher itself; the second one, the final processing with the complete data sets, will take place out of VEGrapher. This is so because running the PSIG software chain globally – that is, not for a small area - is a costly process; since this is the final step, it is of no use to lock VEGrapher with this kind of task. Consequently, the final processing will be run using the external – non-embedded – PSIG software chain.

Harmonizing means merging appropriately the several data sets available at this point; for instance, some edges may have appeared as the result of several processes, so the one with the better gamma attribute must be selected and the remaining ones discarded; other tasks include the recoding of identifiers to guarantee that the same data item is always referred to by the same name.

To start the harmonizing (merging) process, there must exist at least results from one local reprocessing. Then, clicking on the appropriate button in the interface will automatically run this job. The output thus processed must be used as the input for the final run of the PSIG chain. Figure 8 shows the log window for this process.

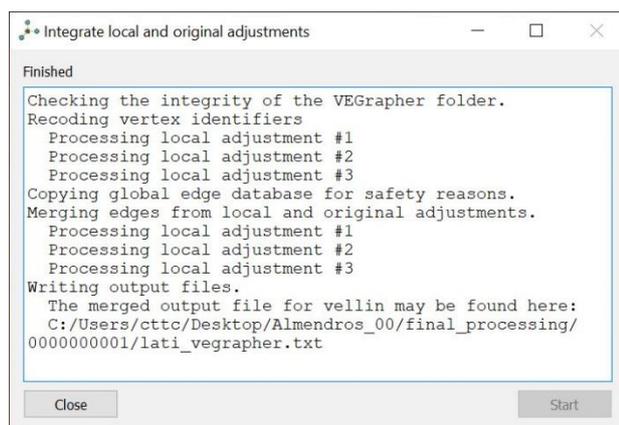


Figure 8: Harmonizing (merging) all the data sets.

3.3 Dealing with high data volumes

The VETools deal with information produced by the PSIG software chain, consisting mainly in points (vertices) and edges. The number of points can be in the millions and that of the edges may be still much bigger. This poses the problem of managing huge volumes of information from the memory and speed standpoints.

The approach adopted by the VETools in general and more specifically by VEGrapher is very simple: manage (load and display) the minimum possible subset of data, at all times, no matter how big the original data set is.

To be able to behave in this way, the outputs created by VEReady are essential, since they allow (1) to select only the information that must be displayed and (2) to retrieve it quickly. The question is: how to tell what to display?

VEGrapher is a data viewer; as any other viewer, it offers its users the possibility of zooming and panning the work scene. This means that the viewport, that is, the area actually being viewed by the users, will normally cover only a subarea of the whole data set, including much less data than the whole scene. Managing invisible data (that is, information out of the area covered by the viewport) is useless, since the user will not be able to see it. Thus, the viewport becomes the essential concept on which VEGrapher's data managing and visualization algorithms relies.

Loading and displaying only data included in the viewport reduces the problem from managing millions (or tens or hundreds of millions) of items to, normally, thousands of these. Obviously, the display may be constantly updated when the viewport is changed. This is why the use of efficient data stores is a must – and this is why the role played by VEReady is so important.

This viewport-based approach applies not only to the points and edges themselves, but also to the background image used to support the users to properly interpret the scene.

3.3.1 Dealing with points and edges: VEGrapher use the spatial data index created by VEReady to load the subset of points that actually lie inside the rectangle defined by the current viewport. Spatial indexes are very efficient when answering spatial queries (such as: “retrieve the subset of elements that fall inside this rectangle”), and their use is the keystone making possible the implementation of the viewport-based approach; should these not be available, finding the points inside the viewport would be an endless, inefficient task invalidating the concept completely.

Figure 9 depicts the concept of viewport-based point selection. In the figure, only red PS, that is, those inside the viewport, would be loaded and shown on the screen.

Edges are not shown by default since this approach would imply such an excess of information that it would become useless. These are shown only when the point of interest is clicked. To be able to ascertain in real time what are the edges starting at some point (the one the user has just clicked), the points to edge levelDB database (see section 3.1) is used; then, once that the list is available, the edge binary file is checked and the data for the edge(s) found read. This takes a negligible time, not noticeable by the user since the refresh of the display is immediate.

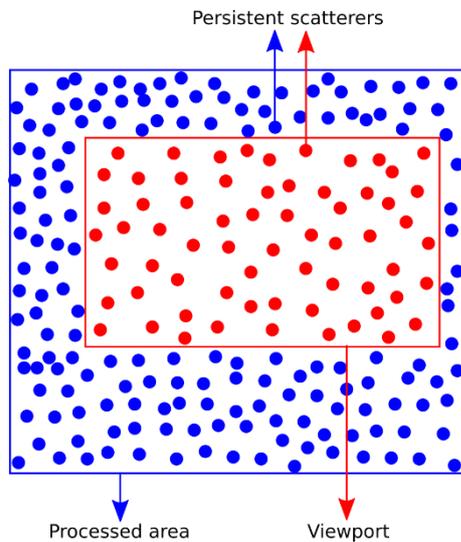


Figure 9: Selection of points inside the application's viewport.

3.3.2 Dealing with (big) background images: Background images are used to help the user of VEGrapher to better understand the problem. Normally, displaying images is straightforward for regular viewers unless these become too big to handle; for instance, specialized, very well-known, well-seasoned image editing tools such as The GIMP (The GIMP Team, 2021) had trouble opening images whose dimensions exceeded 50000×50000 pixel². It is true that these dimensions may seem excessive for the SAR images available nowadays, but VEGrapher has been designed to cope with arbitrarily large imagery to be able to manage change: sensors improve constantly, so images whose dimensions may seem incredible today may be available in a near future.

As stated in section 3.1, one of the outputs of VEReady is a tessellation or the original background image, thus producing regular image tiles. The size of these tiles will always be small enough as to be easily manageable by VEGrapher.

Then, when displaying the background image, VEGrapher will load only the tiles that are strictly necessary to cover the current viewport, as shown in Figure 10.

In Figure 10.:

- the yellow grid shows how the original image has been tessellated; each tile, like the one shown in the lower left corner, may be loaded and displayed independently of the others,
- the red rectangle depicts the viewport, that is the area actually being viewed by the user, and
- the green rectangle stands for the actual set of tiles that must be loaded to guarantee that such viewport includes the part of the background image to show.

This technique guarantees that only the almost strictly necessary amount of memory is loaded: the tiles in memory will usually exceed the limits of the viewport, as shown in Figure 10, so such amount will always be a little bit higher than the minimum required. However, this is a negligible excess.

4. DISCUSSION

The VETools are still under development; although these are very close to be completed, the complexity of embedding the

PSIG software chain as a component of VEGrapher made impossible to fully process a real data set to present some results in this work.

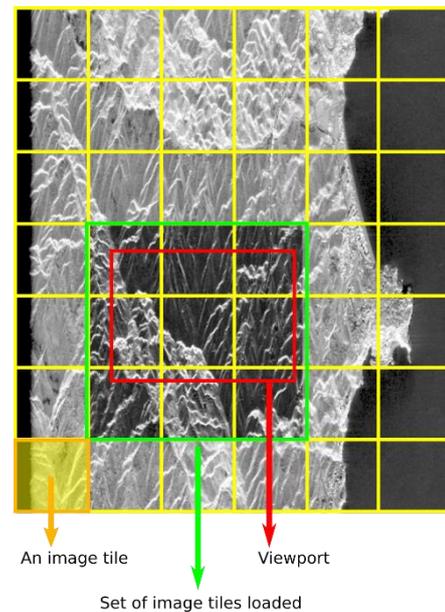


Figure 10: Background image tiles and viewport.

The difficulties come from the variety of sub-workflows that are possible in the PSIG software chain. When such chain is operated manually by humans, they are able, for instance, to set the appropriate options, prepare the necessary inputs or create the required folders. On the contrary, when the chain is run by VEGrapher, the difficulty to manage all these aspects is much higher.

Therefore, at the moment of publishing this work, the aspects related to the organization of data (VEReady), the efficient management of high volumes of information, as well as the polishing of the graphic user interface of VEGrapher are, if not fully finished, very close to the final version. The tasks still to complete are those related to the symbiosis between VEGrapher and the execution of the PSIG software chain. Later publications on this topic will include actual results.

5. CONCLUSIONS

Although still unfinished, the first contact between the users and the VETools prove that these may become very useful tools complementing the PSIG software chain. The main problem to overcome to finish VEGrapher is the integration between this tool and the several workflows present in the PSIG software chain. Nonetheless, the implementation of a single, very limited workflow has allowed to check its performance and ergonomics as well as its suitability to take care of the tasks it has been designed for using a tailored data set.

This incomplete implementation made possible, however, to verify that the tools meet the initial expectations, thus opening the door to a new workflow where the PSIG software chain may be complemented with the VETools to help the expert to obtain quality deformation maps faster.

It is expected to finish the VETools, fully integrating the whole set of PSIG workflows, within the year 2021.

REFERENCES

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., 2017: A Methodology to Detect and Update Active Deformation Areas Based on Sentinel-1 SAR Images, *Remote Sensing*, 9(10), 1002.

Biescas, E., Crosetto, M., Agudo, M., Monserrat, O., Crippa, B., 2007: Two radar interferometric approaches to monitor slow and fast land deformation. *Journal of Surveying Engineering* 133(2), 66-71.

Devanthery, N., Crosetto, M., Monserrat, O., Cuevas-González, M., Crippa, B., 2014: An approach to persistent scatterer interferometry. *Remote Sensing* 6(7), 6662-6679.

Hadjieleftheriou, M., 2019. libspatialindex 1.9.3 documentation. libspatialindex.org/en/latest/ (16 April 2021).

LevelDB development team, 2021. LevelDB is a fast key-value storage library written at Google that provides an ordered mapping from string keys to string values. github.com/google/leveldb (16 April 2021)

Navarro, J. A., Tomás, R., Barra, A., Pagán, J. I., Reyes-Carmona, C., Solari, L., Vinielles, J. L., Falco, S., Crosetto, M., 2020: ADAtools: Automatic Detection and Classification of Active Deformation Areas from PSI Displacement Maps. *ISPRS Int. J. Geo-Inf.* 9(10), 584.

Tomás, R., J. Pagán, J. I., Navarro, J.A., Cano, M., Pastor, J. L., Riquelme, A., Cuevas-González, M., Crosetto, M., Barra, A., Monserrat, O., López-Sánchez, J. M., Ramón, A., Iborra, S., del Soldato, M., Solari, L., Bianchini, S., Raspini, F., Novali, F., Ferreti, A., Constantini, M., Trillo, F., Herrera, G., Casagli, N., 2019: Semi-Automatic Identification and Pre-Screening of Geological-Geotechnical Deformational Processes Using Persistent Scatterer Interferometry Datasets, *Remote Sensing*, 11(14), 1675.

The GIMP Team, 2021. GIMP - GNU Image Manipulation Program. www.gimp.org (16 April 2021).

The Qt company, 2020. Qt | Cross-platform software development for embedded & desktop. www.qt.io (16 April 2021).