

## MEJORANDO LA CALIDAD DE LOS MAPAS DE DEFORMACIONES GENERADOS POR LA CADENA PSIG DEL CTTC: LA SUITE DE HERRAMIENTAS VETOOLS

### IMPROVING THE QUALITY OF DEFORMATION MAPS PRODUCED BY CTTC'S CHAIN: THE VETOOLS SUITE

José A. Navarro\*, Anna Barra, Oriol Monserrat, Michele Crosetto

Centre Tecnològic de Telecomunicacions de Catalunya (CTTC/CERCA), Av. Carl Friedrich Gauss 7, 08860 Castelldefels, España.  
[jose.navarro@cttc.es](mailto:jose.navarro@cttc.es); [anna.barra@cttc.es](mailto:anna.barra@cttc.es); [oriol.monserrat@cttc.es](mailto:oriol.monserrat@cttc.es); [michele.crosetto@cttc.es](mailto:michele.crosetto@cttc.es)

#### Abstract:

The PSIG software chain, which has been producing terrain displacement maps for one and a half decades now, has been entirely developed by the members of the Geomatics Division of the Centre Tecnològic de Telecomunicacions de Catalunya (CTTC). Despite its high degree of automation, PSIG offers its users a noticeable degree of flexibility, making possible to control the relevant set of parameters defining how the processes involved in the chain will behave. However, these parameters, which are applied uniformly to the whole area being processed, may not be the best ones when considering specific local, critical areas. Running the PSIG software chain to process each of these critical areas using tailored sets of parameters is the solution to obtain a product with a good overall quality. However, identifying these areas without an appropriate set of analysis tools may prove very complicated. The VETools, the subject of this work, are targeted at solving this problem. With the VETools it is possible to perform the aforesaid tasks, namely, identifying - visually - the areas that should be enhanced and then relaunch their reprocessing with new control parameters, having the possibility to assess (again, visually) the results and thus rejecting or accepting these until a set of satisfactory solutions are found to be merged afterwards in a unique product with the requested overall quality.

**Key words:** geoprocessing, graphical tools, ground deformation maps, DInSAR

#### Resumen:

La cadena PSIG, en producción desde hace ya una década y media, ha sido desarrollada desde sus inicios por el personal de la actual división de Geomática del Centre Tecnològic de Telecomunicacions de Catalunya. Dicha cadena goza de un altísimo grado de automatización, pero aún así es capaz de ofrecer a sus usuarios una gran flexibilidad, dando la posibilidad de definir aquellos parámetros clave que controlan los diferentes procesos de la cadena. No obstante, esos parámetros, los cuales se aplican uniformemente a la totalidad del área a procesar, pueden no ser los más apropiados cuando se tienen en cuenta ciertas zonas locales críticas. Para resolver este problema, sería suficiente con ejecutar la cadena PSIG circunscrita en esas zonas usando un conjunto de parámetros adecuados para ellas, pudiéndose así obtener finalmente un producto cuya calidad global sea la apropiada. No obstante, identificar dichas áreas puede ser muy complicado sin las herramientas de análisis apropiadas. Esta es la tarea de las VETools, las cuales son el objeto de este artículo: con ellas es posible identificar, visualmente, dichas áreas, lanzar su re-procesado mediante la cadena PSIG tantas veces como sea necesario usando parámetros a medida, rechazando o aceptando los resultados así obtenidos (también visualmente) hasta llegar a un conjunto de soluciones que puedan integrarse en una sola y que goce de la calidad global perseguida.

**Palabras clave:** geoprocesado, herramientas gráficas, mapas de deformación del terreno, DInSAR

### 1. Introducción

Hace ya más de quince años que la actual división de Geomática del Centre Tecnològic de Telecomunicacions de Catalunya (CTTC) comenzó el desarrollo de la conocida como cadena PSIG, concebida para generar mapas de deformaciones del terreno basándose en la técnica DinSAR, esto es, "Differential Interferometry" usando imágenes radar de apertura sintética (SAR). Véase (Devanthy *et al.* 2014) para más información.

Esta cadena software ha alcanzado en la actualidad un gran nivel de madurez, siendo por tanto utilizada en numerosos y variados proyectos. Con el paso del tiempo,

no solamente se ha conseguido mejorarla para que sea posible utilizarla en diversos tipos de aplicaciones, sino que también ha alcanzado un elevado grado de automatización, eliminando así en lo posible la interacción con los usuarios, reduciendo así el volumen de errores que pudieran cometerse y al mismo tiempo aumentando su productividad. Es importante comentar que, a pesar de dicho grado de automatización, PSIG ofrece a sus usuarios la posibilidad de decidir el conjunto de parámetros que controlan el comportamiento de todos y cada uno de los componentes (aplicaciones) que integran dicha cadena.

\* Corresponding Author: José A. Navarro, [jose.navarro@cttc.es](mailto:jose.navarro@cttc.es)

Para complementar la cadena, e incluso acercarla a grupos de usuarios con un nivel de experiencia menor, dentro de la división de Geomática se están llevando a cabo desarrollos adicionales, tales como la identificación de “Active Deformation Areas” (ADA), esto es, “áreas de deformación activa” – véase por ejemplo (Barra *et al.* 2017; Tomás *et al.* 2019; Navarro *et al.* 2020) – o la generación de mosaicos de imágenes para datos de coherencia, amplitud o interferogramas (des)enrollados que permitan una visualización y análisis de la situación con mayor eficacia que en la actualidad. Otro trabajo muy importante, todavía en desarrollo, pero muy próximo a su finalización, es el de las VETools, las cuales son el objeto de este artículo, siendo descritas en cierto detalle en la Sección 2.

Con el objetivo de justificar la necesidad de las VETools y para ubicar al lector en el debido contexto, a continuación, se describirán brevemente y de forma muy general los diferentes pasos que ejecuta la cadena PSIG para generar un mapa de deformaciones del terreno – el cual consiste en la velocidad con la cual se mueve un conjunto de puntos con características de retrodispersión estables y sus correspondientes series temporales de deformación.

Las entradas para dicho proceso son (1) un conjunto de  $n$  imágenes SAR corregistradas (2) la dispersión de amplitud y (3) un conjunto de  $m$  ( $m > n$ ) interferogramas enrollados.

Así pues, a *grosso modo*, los diferentes pasos que se ejecutan son los que se indican a continuación. Para más detalles, se remite al lector a (Biescas *et al.* 2007) y (Devanthery *et al.* 2014).

- **Paso 1:** selección de puntos. Se seleccionan aquellos puntos que garantizan una buena calidad de las fases de entrada en el conjunto de interferogramas.
- **Paso 2:** Cálculo de aristas entre píxeles. Los diferentes píxeles se conectan entre sí por medio de aristas, y para cada una de ellas, la diferencia de fase se calcula para cada interferograma enrollado. La velocidad lineal anual (modelo lineal) y el error topográfico para cada arista se obtienen en base a las diferencias de fase calculadas (observaciones): véase (Biescas *et al.* 2007). La bondad (calidad) del ajuste de estos datos así obtenidos se representa por medio del conocido como factor gamma ( $\gamma$ ).
- **Paso 3:** reordenación. Para estimar la velocidad se selecciona un determinado conjunto de aristas. Para ello se usa como criterio los factores gamma de las aristas en cuestión, así como y un valor umbral que se emplea como referencia para aceptarlas o rechazarlas. La selección del valor umbral puede ser problemática, ya que en función de éste se eliminarán ciertas aristas y algunos puntos podrían quedar desconectados de otros, de manera que no aparecerían en el mapa final de velocidades. Estos grupos de puntos no conectados conducen a la imposibilidad de obtener una solución única.
- **Paso 4:** integración. Como el nombre de este paso indica, aquí se trata de obtener la velocidad de todos y cada uno de los puntos integrando las velocidades diferenciales asociadas a cada una de las aristas en las que está involucrado. El mismo tipo de proceso

se efectúa para obtener el error topográfico. Téngase en cuenta que esto solamente se puede llevar a cabo con grupos de puntos conectados entre sí. Tal y como se decía en el paso 3, si existen varios grupos inconexos entre ellos, no es posible obtener una solución única, con lo cual serán necesarios tantos pasos de integración como grupos de este tipo existan.

Para cada uno de los pasos anteriores, el usuario experto puede seleccionar los parámetros que los controlan; hay que tener en cuenta, no obstante, que dicha selección tendrá una influencia directa en el resultado final, en aspectos tales como la calidad de la estimación de la velocidad (fiabilidad o nivel de ruido), cobertura y densidad de los puntos medidos y, tal y como acaba de comentarse, la conexión entre diversos grupos de puntos y por tanto la posibilidad de obtener una única solución.

La selección de dichos parámetros no es sencilla en absoluto, debido a que la zona a procesar es habitualmente muy amplia y de características muy variadas. Esto hace que escoger un único conjunto de parámetros para procesarla pueda conducir a una solución de calidad inferior a la buscada. Es, por tanto, deseable, poder escoger diferentes conjuntos de parámetros que se ajusten lo mejor posible a aquellas zonas locales del área de interés donde el procesado tendría que realizarse de una forma diferente a la global. Así pues, la obtención del producto final consistiría en la integración tanto del procesado inicial como el de los diferentes procesados de áreas locales críticas que lo precisaran para, finalmente, obtener una solución única.

El gran problema es la identificación de dichas áreas. Y aquí es donde entran en juego las VETools, el conjunto de herramientas de análisis y re-procesado que se discute en este artículo.

La idea de las VETools es, precisamente, facilitar la identificación de las áreas críticas que precisan de un tratamiento diferente, por medio de la visualización de los resultados (velocidad, error topográfico, grupos, aristas existentes, etc.) obtenidos en primera instancia, haciendo posible su re-procesado (integrado como parte de la aplicación para una mayor ergonomía) de acuerdo con un conjunto de parámetros seleccionados a medida para dichas zonas.

La Sección 2 describe a continuación las VETools en detalle.

## 2. La suite de herramientas “VETools”

La suite de herramientas adopta su nombre de los elementos con los que trata, vértices y aristas – en inglés, “Vertices (V)” y “Edges (E)”, esto es, “Vertex & Edge Tools” o VETools.

Tal y como se ha descrito en la sección anterior, el objetivo de dicha suite es el de permitir a los usuarios de PSIG analizar visualmente qué zonas de una determinada área de interés necesitarían ser re-procesadas con un conjunto de parámetros diferente del global para obtener así una solución de mayor calidad.

En *strictu sensu*, solamente una de las dos herramientas que constituyen las VETools, la conocida como VEGrapher, permite llevar a cabo lo que se acaba de enunciar; no obstante, y como se describe a

continuación, el uso del otro componente de la suite, VEReady, es del todo imprescindible. No obstante, y antes de proceder a dicha descripción, se ofrece un resumen muy sintético del flujo de trabajo que hay que respetar para el correcto funcionamiento de dichas herramientas.

El flujo en cuestión es el siguiente:

- *Paso 1:* el área de interés se procesa globalmente utilizando un conjunto de parámetros único por medio de la cadena PSIG. Esto genera una serie de archivos de salida.
- *Paso 2:* la herramienta VEReady entra entonces en juego, no para facilitar al usuario el análisis de los datos, sino por el contrario para reorganizar estos de manera que la segunda herramienta, VEGrapher, pueda funcionar de forma correcta. La Sección 2.1 explica en qué consiste esta reorganización; baste aquí decir que (1) VEReady no precisa de ningún tipo de intervención humana, ya que es una herramienta totalmente automatizada y (2) que su papel es crucial para el correcto desempeño de VEGrapher.
- *Paso 3:* en este punto ya puede ser usada VEGrapher, la segunda herramienta, para analizar los resultados del paso 1, identificándose las zonas problemáticas, lanzando su re-procesado con parámetros específicos tantas veces como sea necesario; y todo ello, asistiendo al usuario con las herramientas de visualización que permitan realizar este trabajo.
- *Paso 4:* finalmente, VEGrapher procederá a la combinación del procesado inicial más tantos re-procesados locales como se hayan efectuado para lanzar la cadena PSIG de manera que se obtenga una solución única con un nivel de calidad suficiente para la totalidad del área de interés.

## 2.1. VEReady

El nombre de esta herramienta, “VEReady”, es un juego de palabras fonético en inglés, ya que intenta sonar como “Be ready” o “prepárate”, dando a entender que su tarea es la de preparar los datos para un ulterior uso eficiente por parte de VEGrapher.

Como se ha comentado en el flujo de trabajo descrito en la sección anterior, este es precisamente su objetivo: transformar la información generada por la cadena PSIG para hacer posible la tarea de su compañera, VEGrapher. Esto es así porque la salida de la cadena PSIG puede incluir millones de puntos y muchísimas más aristas. Si esta información no se organiza de manera que pueda ser accedida rápidamente, entonces la tarea de VEGrapher deviene imposible.

En resumen, VEReady no es más que una herramienta que cambia el formato de los datos. Esto es cierto, pero dicho cambio de formato ha de organizar la información de una manera que haga que su uso por VEGrapher no sea solamente posible, sino que requiera el mínimo tiempo para ser accedida.

A continuación, se detallan las entradas de VEReady, las cuales son, como ya se ha indicado, la salida de la primera ejecución de la cadena PSIG:

- Aristas, constando para cada una de ellas, de la fila y columna (del píxel) tanto del origen como del destino de ésta, así como de los identificadores de los dos puntos (píxeles) involucrados, juntamente con los parámetros asociados a ella, a saber: velocidad, error topográfico y el factor gamma ( $\gamma$ ) cuantificando su calidad.
- Atributos para los puntos, es decir, su velocidad, error topográfico y grupo al cual pertenecen.
- Imagen de fondo. La imagen SAR original convertida a cualquier formato visualizable (como, por ejemplo, JPG) para mejorar la visualización de los puntos y las aristas y ayudar a la comprensión de la situación analizada.

Todas estas entradas vienen en forma ya sea de archivos de texto plano o binarios, pero sin ningún tipo de estructura que permita su acceso aleatorio eficaz, por ejemplo, a partir del identificador de un punto o una arista.

Una vez ejecutado, VEReady reorganiza dicha información produciendo las siguientes salidas:

- Un índice espacial para los puntos. Es importante empezar diciendo que este índice es crucial para el funcionamiento de VEGrapher, ya que, sin él, su operativa no sería posible. Un índice espacial es una estructura de datos que asocia coordenadas (las de un punto, por ejemplo) con un cierto conjunto de atributos (como, en este caso, su velocidad, error topográfico o grupo). Dicha estructura es capaz de responder de forma muy eficiente gracias su especial organización, a preguntas tales como: “¿cuáles son los elementos (puntos) que caen dentro de un determinado rectángulo?”. La capacidad de responder rápidamente a esta pregunta es la piedra angular en que se basa la política de visualización de VEGrapher, como podrá verse en la Sección 3.2.1.
- Si el índice anterior se ocupa solamente de los puntos y sus atributos, un nuevo archivo de salida gestiona los datos relativos a las aristas. El formato de dicho archivo es binario, y es accesible aleatoriamente de forma muy sencilla y rápida a través del identificador de la arista que se desea consultar – identificador, por cierto, generado por VEReady para este propósito.
- Para relacionar los puntos con las aristas, es necesaria una base de datos tal que, dado el identificador de un punto cualquiera, permita recuperar la lista de identificadores de las aristas que en dicho punto se originan. Así pues, una vez que el usuario selecciona un punto para su análisis, es posible saber de forma prácticamente inmediata cuales son las aristas que de él dependen.
- Finalmente, la imagen de fondo se fragmenta en teselas para hacer posible la visualización de escenas arbitrariamente grandes. Esto se hace así para evitar que las imágenes de fondo correspondientes a áreas de interés extremadamente grandes sean, asimismo, muy grandes, de manera que puedan llegar a plantear un problema de visualización a VEGrapher. La Sección 3.2.2 describe con más detalle en qué consiste esta técnica.

Es conveniente hacer notar que tanto el índice espacial como la base de datos relacionando puntos con aristas son capaces de soportar cantidades arbitrariamente grandes de datos, posibilitándose así el trabajo con volúmenes de estos que, de otra manera, serían inmanejables – véase nuevamente la Sección 3.2.1 para más información. Asimismo, su velocidad de desempeño permite tiempos de respuesta razonables al gestionar este tipo de datos en VEGrapher. El índice espacial se ha implementado por medio de la librería libspatialindex (Hadjieleftheriou 2019) y la base de datos de puntos a aristas por medio de la librería levelDB (LevelDB development team 2021).

### 3. VEGrapher

VEGrapher es la herramienta gráfica que permite el análisis de los resultados obtenidos por medio de la ejecución de la cadena PSIG, así como el re-procesado de las ya comentadas áreas críticas que necesiten ser corregidas según el criterio del especialista. Como ya se ha comentado en varias ocasiones, VEGrapher se apoya en la reorganización de los citados resultados realizada por VEReady para realizar esta tarea.

El flujo de trabajo es el siguiente: el usuario revisa las áreas que puedan resultar problemáticas. Cuando identifica una o más de ellas puede lanzar la cadena PSIG directamente desde VEGrapher (mejorándose así la ergonomía de la aplicación) especificando el conjunto de parámetros que, a su juicio, se ajusta mejor a las necesidades de la zona. Hay que notar que este re-procesado se produce dentro de los límites de la zona en cuestión. El re-procesado, no obstante, no es definitivo, ya que el usuario tiene la oportunidad de revisar los resultados y dar su visto bueno o no en función del análisis que realice una vez que dicho re-procesado acabe. Asimismo, no hay límite en cuanto al número de zonas a tratar; pueden ser tantas como se considere necesario. Una vez que se considere que ya se han corregido las zonas locales necesarias, se puede iniciar desde el propio VEGrapher un proceso de fusión de todos los conjuntos de datos (los resultados originales más los relativos a las áreas identificadas como problemáticas) para que estos sean, finalmente, procesados de nuevo por la cadena PSIG obteniéndose así un producto final.

Una herramienta como esta, que ha de manejar enormes cantidades de datos, se enfrenta a una serie de desafíos de diseño e implementación importantes; por ello, y como primera medida, las VETools en su totalidad se han desarrollado en C++, ya que este lenguaje permite crear aplicaciones mucho más eficientes que cuando se emplean otros. La calidad de la interfaz gráfica (ergonomía, velocidad) también es de suma importancia: por ello, se ha utilizado la “Qt framework” (The Qt company 2020) debido a las prestaciones que este marco ofrece.

No obstante, la simple elección de un lenguaje de programación es insuficiente para garantizar el conjunto de funcionalidades necesarias, ni tampoco basta para hacer frente a los diversos desafíos que tales aplicaciones plantean. Ejemplos de tales desafíos son el manejo de ingentes cantidades de datos de forma eficaz, no ralentizando los tiempos de respuesta para no degradar la experiencia del usuario o, por otro lado, el tener que embeber una instancia de la cadena PSIG para

llevar a cabo el re-procesado de las zonas que así lo requieren, facilitando así que el usuario no tenga que trabajar a caballo de múltiples entornos de trabajo.

Las secciones a continuación se extienden en la descripción estas funcionalidades y desafíos.

#### 3.1. Las funcionalidades

En esta sección se describe el subconjunto de funcionalidades más relevante de VEGrapher con el objetivo de permitir al lector hacerse una idea de cuáles son las capacidades de esta herramienta.

##### 3.1.1. Coloreado de puntos según varios criterios

Ésta es la primera gran herramienta a disposición del usuario para ayudarle en su trabajo de análisis. Como se ha comentado anteriormente, los puntos están caracterizados por una serie de atributos, a saber: velocidad, error topográfico y grupo al cual pertenecen.

Poder distinguir esos puntos en función de esos atributos es posible en VEGrapher mediante un coloreado específico. Es decir, es posible distinguir mediante diferentes colores qué valores de dichos atributos definen cada uno de los puntos. Que se utilice el coloreado por uno u otro atributo dependerá del objetivo que en cada momento persiga el analista.

Además, los colores a emplear son definibles por el propio usuario mediante un editor de tablas de colores incluido en la propia aplicación; con este editor, es posible seleccionar, por ejemplo, qué color emplear ya sea para los diferentes intervalos de velocidad o de error topográfico, o para los diferentes grupos de puntos que integran el conjunto de datos. Dichas tablas de colores pueden ser guardadas como archivos externos y ser reutilizadas cuantas veces sea necesario.

Asimismo, y en función de cada proyecto, es posible utilizar diferentes tablas de color, para así mejor adaptarse a la variación en cuanto a los rangos de datos en cada uno de ellos.

Finalmente, es posible alternar entre los diferentes modos de coloración (velocidad, error topográfico, grupo y coloración por defecto no basada en atributos) cada vez que así se desee, para observar los datos de acuerdo con los diferentes criterios existentes. Cada modo de coloración puede emplear su propia tabla de colores.

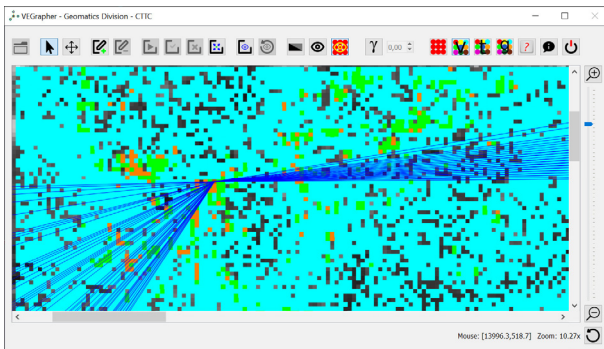
La Figura 1 muestra una zona (con una densidad de puntos altísima) que se ha coloreado de acuerdo con el criterio de velocidad. Pueden apreciarse hasta tres grupos distintos (azul claro, verde, naranja). Las aristas en azul oscuro que aparecen en esta figura se comentarán en la sección siguiente.

##### 3.1.2. Selección y filtrado de aristas en tiempo real

Por defecto VEGrapher no muestra las aristas que conectan los puntos entre sí, ya que el número de estas haría del todo incomprensible el conjunto de datos mostrados. Hay que tener en cuenta que, si los puntos pueden llegar a ser millones, las aristas multiplican dicho número notablemente.

Por esta razón, dichas aristas solamente se muestran cuando el usuario selecciona, haciendo clic sobre ellos, alguno de los puntos mostrados en pantalla. La Figura 1

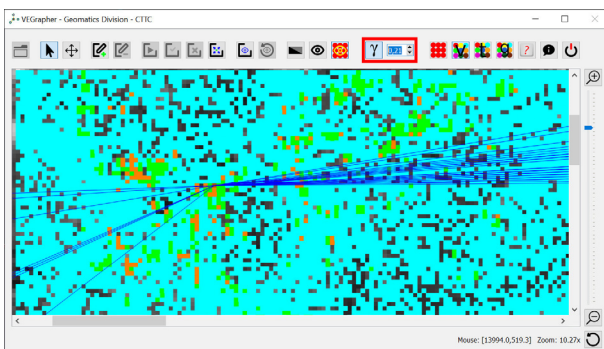
muestra esta situación. Por medio de esta opción el usuario puede comprobar cual es la conectividad del punto seleccionado.



**Figura 1:** Los puntos están coloreados según el atributo de velocidad. Se muestran las aristas que se originan en el punto seleccionado.

Ya se había avanzado, además, que cada arista está etiquetada con el parámetro que mide la calidad de la conexión que representa, es decir, gamma. Dicho parámetro puede oscilar entre cero y uno, siendo el cero el valor correspondiente a la peor calidad posible y uno el asignado a la máxima. VEGrapher permite discriminar qué aristas mostrar modificando de forma interactiva y en tiempo real el valor umbral para el parámetro gamma; cambiando este umbral se mostrarán en pantalla solamente aquellas aristas cuyo valor de gamma sea igual o superior al de dicho umbral. Esto permite observar qué conexiones alcanzan un nivel determinado de calidad, siendo así posible evaluar dichas conexiones.

La Figura 2 muestra exactamente la misma zona y el mismo punto que la Figura 1. No obstante, en esta última figura se ha seleccionado un valor de gamma de 0,21. Esa es la razón por la cual el número de aristas ha disminuido ostensiblemente, como puede comprobarse comprobando ambas ilustraciones. Nótese que el umbral para filtrar según el parámetro gamma está recuadrado en rojo en la Figura 2, estando este ubicado en la barra de herramientas principal de la aplicación.



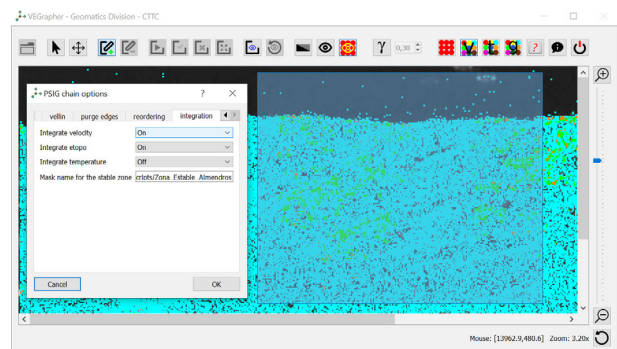
**Figura 2:** Se ha reducido el número de aristas visibles por medio del umbral del parámetro gamma.

### 3.1.3. Re-procesado local de áreas críticas

La identificación de problemas en determinadas zonas del área de interés es solamente una parte del trabajo que el analista realiza con VEGrapher, ya que el resultado de esta actividad puede llevar a la necesidad de volver a procesar localmente algunas zonas dentro del área de interés.

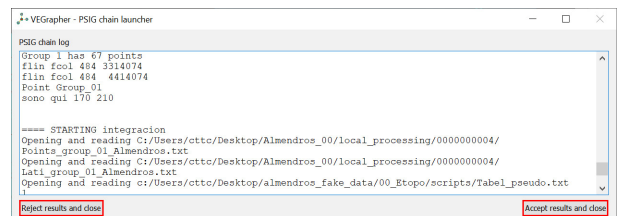
VEGrapher incluye la posibilidad de realizar este re-procesado sin abandonar la propia aplicación, ya que con ella se ha embebido una copia de la cadena PSIG. Desde el punto de vista ergonómico esto es muy conveniente, ya que (1) no es necesario cambiar de entorno de trabajo mientras se realiza todo este proceso y (2) los resultados del re-procesado local pueden ser, a su vez, analizados inmediatamente dentro de VEGrapher una vez que los cálculos han finalizado.

Así pues, y una vez que el usuario ha identificado que hay que re-procesar una determinada zona, éste solamente tiene cambiar el modo de trabajo de la aplicación a “re-procesado” por medio de un botón provisto a tal efecto en la barra de herramientas de VEGrapher; inmediatamente podrá seleccionar el área afectada dibujando un rectángulo que la recubra con el cursor (véase la parte derecha de la Figura 3). Inmediatamente aparecerá una ventana incluyendo diferentes pestañas, una para cada una de las aplicaciones involucradas en la cadena PSIG (véase otra vez la Figura 3, esta vez en su parte izquierda). Es en estas pestañas donde el usuario puede modificar los valores ofrecidos por defecto de los parámetros que controlan el comportamiento de dichas aplicaciones. Una vez que se haya cumplimentado este paso es posible poner en marcha la cadena PSIG pulsando un solo botón.



**Figura 3:** Re-procesando localmente un área identificada como problemática.

La acción anterior hace que aparezca una nueva ventana donde se muestra el registro (“log”) de las diferentes aplicaciones que, sucesivamente se ejecutarán hasta, finalmente, completarse el re-procesado; esto queda ilustrado en la Figura 4.

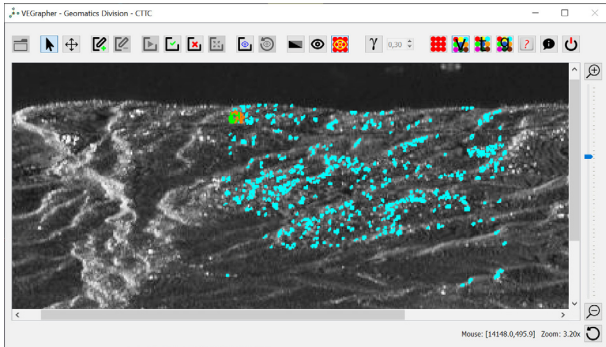


**Figura 4:** El registro de la ejecución de la cadena PSIG. Ya posible rechazar o aceptar preliminarmente los resultados del proceso.

Por medio del estudio de este registro el analista ya puede tomar una decisión preliminar sobre si los resultados de dicho re-procesado son satisfactorios o no; caso de no estar conforme con ellos, puede rechazarlos pulsando sobre el botón provisto a tal efecto (recuadrado en rojo en la parte inferior izquierda de la Figura 4). Si por

el contrario considera que dichos resultados son, aparentemente correctos, puede aceptarlos preliminarmente pulsando el otro botón disponible (también recuadrado en rojo, esta vez en la parte inferior derecha de la misma figura).

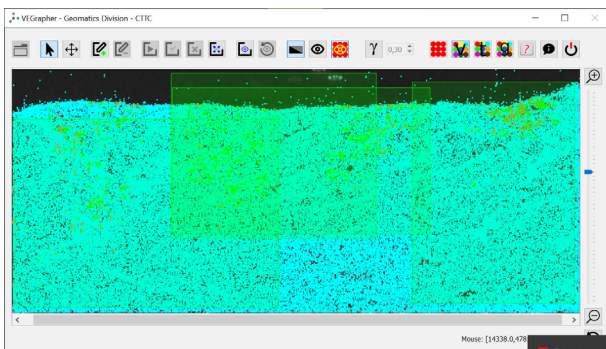
En el supuesto de que se acepten preliminarmente los resultados, VEGrapher mostrará los puntos afectados que pertenecen, exclusivamente, a la zona que se acaba de re-procesar, tal y como se muestra en la Figura 5.



**Figura 5:** Visualización exclusiva de un re-procesado local de un área crítica.

En este punto, el analista puede aplicar de nuevo las técnicas ya descritas (coloreado de puntos, filtración de aristas por gamma) para decidir si dichos resultados le satisfacen de forma definitiva. Caso de ser así, y pulsando el botón adecuado en la barra de herramientas superior, puede aceptar dichos resultados, los cuales se guardarán para un proceso ulterior de integración. Si por el contrario decide rechazarlos, estos se eliminan, restituyéndose la situación anterior.

Para facilitar el control sobre qué se ha re-procesado y qué no, VEGrapher muestra las áreas que ya han pasado por este paso resaltándolas por medio de rectángulos semitransparentes de color verde pálido que actúan como marcadores, tal y como puede comprobarse en la Figura 6. Caso de que molesten, el usuario puede activar o desactivar la visualización de dichos marcadores a voluntad.

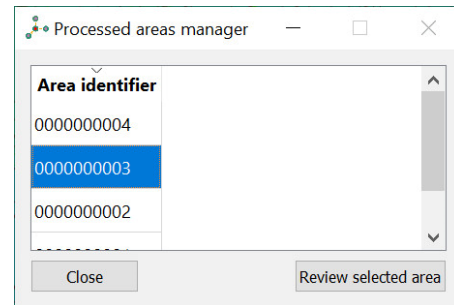


**Figura 6:** Los marcadores semitransparentes resaltan las áreas críticas que ya se han re-procesado localmente.

Pueden re-procesarse localmente tantas áreas críticas como sea necesario hasta que se considere que no son necesarias más mejoras.

Nótese que VEGrapher incluso ofrece la posibilidad de revisar un re-procesado previamente aceptado por el analista, aumentándose así no solo la flexibilidad de la

herramienta sino también reduciéndose la incertidumbre de dicha persona, ya que incluso habiendo dado por bueno un re-procesado, es posible revisarlo si se considera necesario. A tal efecto, VEGrapher ofrece la herramienta de revisita; pulsando el correspondiente botón en la barra de herramientas aparece un diálogo que muestra las diferentes zonas ya trabajadas para que puedan ser seleccionadas y, si se desea, volver a lanzar un nuevo re-procesado. Dicho diálogo se muestra en la Figura 7.



**Figura 7:** El diálogo que permite visitar e incluso re-procesar las áreas críticas ya procesadas anteriormente.

Una vez que éste se ha lanzado, el mecanismo funciona de forma idéntica al re-procesado ya descrito; si éste se acepta, se eliminan los resultados anteriores substituyéndose estos por los nuevos, pero si por el contrario se aceptan, lo que sucede es que se eliminan los resultados anteriores conservándose los últimos.

### 3.1.4. Preparando el resultado final: armonización y fusión de datos

Para obtener un resultado final cuya calidad sea la buscada, falta todavía ejecutar una vez más la cadena PSIG con todos los conjuntos de datos disponibles; esto incluye tanto el procesado global original como aquellos relativos a las diferentes zonas críticas identificadas y re-procesadas tal y como se ha descrito.

VEGrapher, a pesar de que podría ocuparse de ello, no asume dicho paso final, ya que, al tratarse de un procesado global, éste consumirá mucho tiempo y recursos; por el contrario, este paso se hará externamente con la cadena PSIG.

No obstante, y antes de poder proceder así, es necesario realizar una fusión de todos los datos disponibles hasta el momento, armonizando estos.

¿En qué consiste dicha armonización? Un ejemplo sería la necesidad de unificar los identificadores de las aristas; éstas pueden aparecer múltiples veces, ya sea en el procesado global o en uno o más de los re-procesados locales. Hay que re-codificar dichos identificadores de manera que la misma arista, aparezca donde aparezca, sea siempre etiquetada con el mismo identificador para evitar errores. Y si aparecen aristas nuevas, sus identificadores no deben entrar en conflicto con los ya existentes. Continuando con las aristas, otro ejemplo sería que cuando aparecen varias instancias de la misma arista, tal y como se comenta en el ejemplo anterior, es preciso conservar solamente aquella cuyo valor del parámetro gamma sea mejor.

La armonización es un proceso muy sencillo de ejecutar en VEGrapher; eso sí: para que esté disponible, es

necesario que al menos se haya re-procesado localmente un área crítica – ya que, de no ser así, no existiría nada que armonizar. Para iniciarla, basta con pulsar el botón correspondiente y tras la preceptiva confirmación por parte del usuario, el proceso se inicia sin más requisitos. La Figura 8 muestra la ventana con el registro de dicha operación una vez que ésta ha acabado.

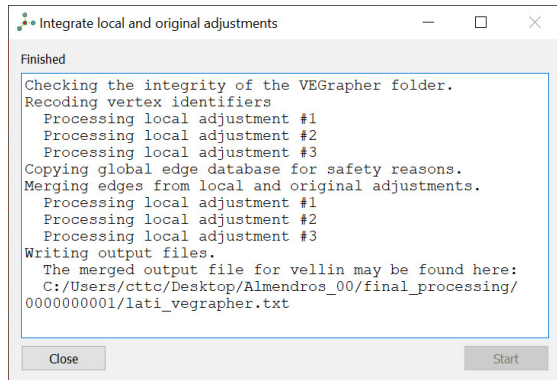


Figura 8: Armonización de resultados para el procesado final.

Hecho esto, el usuario puede tomar los resultados de este proceso para alimentar la ejecución final de la cadena PSIG.

No obstante, es interesante comentar que si el analista deseara proseguir re-procesando localmente más áreas críticas, esto es posible; asimismo, también es posible volver a armonizar resultados nuevamente. De esta manera, no hay límite a la experimentación que el usuario puede hacer para mejorar el producto final.

### 3.2. Los desafíos

Para implementar las funcionalidades descritas y conseguir que éstas reaccionen con tiempos de respuesta razonables que no entorpezcan el trabajo del usuario, ha sido necesario hacer frente a algunos desafíos.

El volumen de datos que han de gestionar las aplicaciones incluidas en las VETools puede llegar a ser muy grande; el número de puntos gestionado puede ser de (decenas de) millones; el de aristas, consecuentemente, aún puede llegar a límites mayores.

Esto plantea inmediatamente el problema de cómo trabajar con esas cantidades de datos y, además, de que el rendimiento de las aplicaciones que trabajan con ellos (especialmente VEGrapher) no se convierta en un obstáculo insalvable que anule su posible utilidad.

La aproximación para resolver este problema es doble. En primer lugar, los datos generados por la cadena PSIG deben ser reorganizados para que su uso de acuerdo con patrones de acceso aleatorio sea posible. Esto se implementa por medio del programa VEReady, descrito en la Sección 2.1.

El segundo vector de ataque consiste en racionalizar la manera en que los datos se cargan y muestran en VEGrapher. Sería absurdo pretender cargar en memoria y listos para mostrarse todos los puntos y aristas existentes en el conjunto de datos; con volúmenes muy inferiores a los comentados el colapso de una aplicación que pretendiera manejarlos estaría garantizado. Por ello,

VEGrapher utiliza el concepto de “viewport” (zona visible) para reducir en lo posible el volumen de datos con los que tenga que tratar.

Cuando se utiliza VEGrapher (o cualquier otra aplicación de visualización) es normal utilizar las funcionalidades de “panning” y “zooming” para mostrar, en mayor o menor detalle los datos a estudiar. Por lo tanto, el área que realmente se muestra es una, mientras que el área de estudio (potencialmente visualizable) es otra y mucho mayor. Si la aplicación solamente carga y visualiza, en todo momento, aquella información que debería presentarse en el área que realmente se está observando, el volumen de datos a gestionar se reduce drásticamente (pasando, por ejemplo, a “solamente” decenas de miles de objetos). Pretender gestionar aquellos elementos que son invisibles por caer fuera de dicha zona visible no reporta ningún beneficio.

La contrapartida es que es necesario actualizar constantemente esa zona visible descargando (eliminando) aquello que no se ve y cargando lo que deviene visible. De ahí que la forma en que se organiza la información (VEReady) sea tan importante.

Las secciones a continuación explican como se gestionaron tanto puntos y aristas como la imagen de fondo que se usa como respaldo para una mejor interpretación de los datos.

#### 3.2.1. Gestionando los puntos y las aristas

VEGrapher trabaja de acuerdo con el modelo arriba presentado de mostrar exclusivamente aquellos datos que son visibles en todo momento, ignorando los que no lo son. Para ello, descansa en la información generada por VEReady, la cual permite un acceso inteligente y rápido a los datos buscados. En particular, la primera pieza clave es el índice espacial implementado con la librería libspatialindex. El lector puede encontrar más información sobre esta librería en (Hadjieleftheriou 2019).

Gracias a este índice, que permite almacenar cantidades arbitrariamente grandes de datos juntamente con un índice que permita recuperarlos, es posible formular la consulta esencial que permitirá conocer qué elementos (puntos, y consecuentemente, aristas) forman parte de la zona visible en todo momento. Dicha consulta no es otra que discernir cual es el conjunto de puntos que caen dentro de un determinado rectángulo (véase la Figura 9).

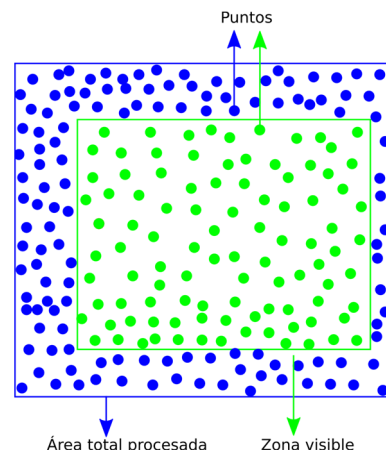


Figura 9: El índice espacial permite discernir muy rápidamente qué puntos pertenecen a la zona visible (en verde).

El índice espacial permite responder a esta cuestión en un tiempo negligible desde el punto de vista humano y, posteriormente, recuperar el conjunto de datos implicados también muy rápidamente. Esto hace posible el refresco continuo de la zona visible.

La importancia de disponer de un índice espacial es tal, que de no existir este, la búsqueda de los elementos presentes en la zona visible implicaría un tiempo de respuesta tan grande que haría del todo imposible implementar una herramienta como la que se está discutiendo.

Tal y como ya se ha comentado, las aristas no se muestran de entrada, ya que esto solamente serviría para generar confusión al ser su volumen tan grande que no sería posible distinguir los elementos individuales dentro de la zona visible. Éstas solamente se muestran cuando se selecciona (haciendo clic) sobre un punto determinado.

En este momento entra en juego la base de datos que conecta un punto con los identificadores de las aristas que parten de dicho punto. Esta base de datos (en realidad, un diccionario relacionando claves – los identificadores de los puntos – con su “significado” – la lista de identificadores de las aristas) también es capaz de almacenar un volumen arbitrariamente grande de datos y ofrece tiempos de acceso igualmente negligibles desde el punto de vista humano.

Esta base de datos está implementada con la librería levelDB, sobre la cual puede obtenerse información en (LevelDB development team 2021).

Es importante comentar que, si bien la funcionalidad de “diccionario” de esta base de datos está disponible en C++ utilizando librerías estandarizadas, es la posibilidad de levelDB de guardar en disco diccionarios de un tamaño arbitrariamente grande lo que posibilita esta función de localización de las aristas relacionadas con un punto. De haberse tenido que utilizar los diccionarios habituales (no persistentes, es decir, sin soporte en disco) el volumen de información a manejar también habría imposibilitado la implementación de VEGrapher.

Una vez obtenidos los identificadores de las aristas, éstas se recuperan del archivo binario que las contiene accediendo de forma aleatoria (por contraposición a secuencial). Este archivo es estándar, y solamente está afectado por las posibles limitaciones de tamaño que el sistema operativo de base (Windows 10 en este caso) pueda imponer.

Todo este proceso (selección del punto, obtención de los identificadores de las aristas, lectura de las aristas propiamente dichas, dibujado en pantalla) sucede en un tiempo mínimo, tan bajo que la experiencia del usuario es que todo ello sucede de forma instantánea.

### 3.2.2. La gestión imágenes de fondo de gran tamaño

VEGrapher permite mostrar la imagen de fondo correspondiente al área de interés con la que se está trabajando con el objetivo de ayudar al usuario en el proceso de interpretación de los datos (puntos, aristas) que se muestran en pantalla.

Esta característica de VEGrapher no debería ser en absoluto objeto de comentarios si no fuese por la técnica

que se utiliza para mostrar dichas imágenes: el teselado y visualización selectiva. Hay que decir, no obstante, que teniendo en cuenta los tamaños de las imágenes que se emplean hoy en día, esta funcionalidad no sería necesaria; no obstante, se ha implementado ya en previsión del aumento de dichos tamaños, cosa que, en un futuro, podría comprometer el rendimiento de la aplicación.

Por ejemplo, aplicaciones muy reputadas de manipulación de imágenes tales como The GIMP – véase (The GIMP Team 2021) - tienen serios problemas para manejar aquellas imágenes que se mueven en rangos próximos a los 50.000 x 50.000 píxel<sup>2</sup>. Es en previsión de que el aumento de las capacidades de los sensores pueda llegar a proporcionar imágenes en este rango de tamaños por lo que VEGrapher utiliza, ya, la técnica del teselado y su carga y visualización selectiva.

Es VEGrapher quien transforma la imagen de fondo original en tantas teselas de tamaño manejable como sea necesario, teselas que se incluyen con el resto de información transformada por esta aplicación. Entonces VEGrapher, empleando el concepto de zona visible ya descrito en las Secciones 3.2 y 3.2.1, carga en memoria y visualiza solamente aquellas teselas que son visibles en cada momento, deshaciéndose de aquellas que devienen invisibles al cambiar la zona en la que el usuario está trabajando. La Figura 10 ilustra este concepto.

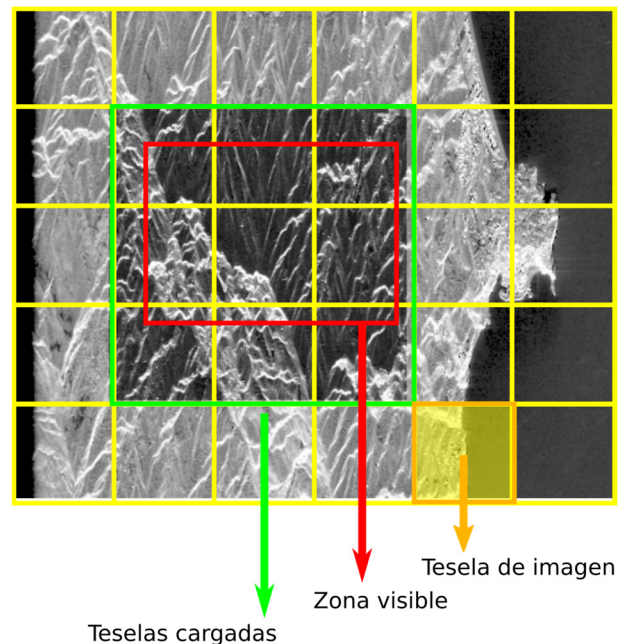


Figura 10: Zona visible: teselado y carga y visualización selectiva.

En dicha figura puede apreciarse (1) cómo la imagen original ha sido teselada en fragmentos más pequeños, los representados por la malla amarilla así como por la zona sombreada en el mismo color, (2) cuál es la zona visible en pantalla, en este ejemplo concreto aquella enmarcada por un rectángulo de color rojo y, finalmente (3) cuáles son las únicas teselas que hay que cargar para garantizar que dicha zona visible sea, realmente, visible; dichas teselas son las comprendidas dentro del recuadro de color verde.



Esta técnica puede llegar a ahorrar mucha memoria a la hora de mostrar imágenes de gran tamaño. Por el contrario, la gestión de la visualización de estas teselas no consume apenas recursos, siendo estos incluso negligibles, y quedando, por tanto, disponibles para la gestión de los puntos y las aristas descrita en la Sección 3.2.1.

#### 4. Discusión y conclusiones

En primer lugar, hay que recordar que las VETools son un conjunto de herramientas que todavía están en fase de desarrollo, si bien su conclusión está próxima.

Se ha comentado en la Sección 1 que la cadena PSIG está altamente automatizada, ofreciendo aún así a sus usuarios la posibilidad de controlar multitud de parámetros que controlan su comportamiento; estos parámetros, en ocasiones, generan diferentes flujos de trabajo.

Son, por un lado, esta variabilidad en lo que se refiere a cómo se ejecuta la cadena PSIG y por otro el hecho de que ésta esté embebida en VEGrapher, las razones por las cuales todavía no se pueda contar con una versión final de dicha herramienta en el momento de escribir este artículo; mientras que un usuario humano no tiene problemas en ajustar lo que sea necesario cuando PSIG se ejecuta de manera independiente, implementar esta lógica de forma automatizada dentro de una aplicación implica una complejidad que está todavía por resolver.

Está previsto, no obstante, tener resuelto este problema dentro del año 2021.

Por esta razón no se han presentado resultados que permitan respaldar de forma absolutamente objetiva los beneficios de las VETools.

Comoquiera que sea, y a pesar de no estar completamente acabadas, su uso por parte de los usuarios ya indica claramente que las VETools pueden llegar a ser unas herramientas de gran utilidad para complementar la cadena PSIG. Esto es así gracias a la implementación, ya disponible en VEGrapher, de uno de los flujos de trabajo de dicha cadena: se ha podido comprobar tanto el rendimiento como su ergonomía, así como su adecuación para asumir las tareas que se esperan de ellas.

Asimismo, esta implementación, aunque aún incompleta, ha permitido validar que estas herramientas satisfacen las expectativas iniciales, lo cual permite creer en un futuro flujo de trabajo en el cual la cadena PSIG se vea acompañada de las VETools para ayudar al experto en su tarea de obtener mejores mapas de deformación del terreno.

#### Agradecimientos

Este trabajo ha sido co-financiado por el Fondo Europeo de Desarrollo Regional (“European Regional Development Fund”) a través de Interreg V-A España, Francia y Andorra (POCTEFA 2014-2020), proyecto EFA295/19.

#### 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., and MORETTI, S., 2017. A Methodology to Detect and Update Active Deformation Areas Based on Sentinel-1 SAR Images, *Remote Sensing*, **9**(10), 1002. DOI: 10.3390/rs9101002
- BIESCAS, E., CROSETTO, M., AGUDO, M., MONSERRAT, O., and CRIPPA, B., 2007. Two radar interferometric approaches to monitor slow and fast land deformation. *Journal of Surveying Engineering*, **133**(2), pp. 66-71. DOI: 10.1061/(ASCE)0733-9453(2007)133:2(66)
- DEVANTHÉRY, N., CROSETTO, M., MONSERRAT, O., CUEVAS-GONZÁLEZ, M., and CRIPPA, B., 2014. An approach to persistent scatterer interferometry. *Remote Sensing* **6**(7), pp. 6662-6679. DOI: 10.3390/rs6076662
- HADJIELEFTHERIOU, M., 2019. libspatialindex 1.9.3 documentation. Available: <https://libspatialindex.org/en/latest/> [4/29, 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. Available: <https://github.com/google/leveldb> [4/29, 2021].
- NAVARRO, J. A., TOMÁS, R., BARRA, A., PAGÁN, J. I., REYES-CARMONA, C., SOLARI, L., VINIELLES, J. L., FALCO, S., and CROSETTO, M., 2020. ADAtools: Automatic Detection and Classification of Active Deformation Areas from PSI Displacement Maps. *ISPRS Int. J. Geo-Inf.* **9**(10), 584. DOI: 10.3390/ijgi9100584
- 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., and CASAGLI, N., 2019. Semi-Automatic Identification and Pre-Screening of Geological-Geotechnical Deformational Processes Using Persistent Scatterer Interferometry Datasets. *Remote Sensing*, **11**(14), 1675. DOI: 10.3390/rs11141675
- THE GIMP TEAM, 2021. GIMP — GNU Image Manipulation Program. Available: <https://www.gimp.org/> [4/29, 2021].
- THE QT COMPANY, 2020. Qt | Cross-platform software development for embedded & desktop. Available: <https://www.qt.io/> [4/29, 2021].