

# WEB-BASED INTERACTIVE VISUALIZATION OF PS-INSAR POINT CLOUDS FOR EDUCATION AND TRAINING

Dong Liang, Timo Balz, Ziyun Wang, Lianhuan Wei, Mingsheng Liao

State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University,  
129 Luoyu Road, Wuhan 430079, PR China - liangdong@whu.edu.cn

**KEY WORDS:** PS-InSAR, Visualization, Web-based Visualization, WebGL, Point Clouds

## ABSTRACT:

In radar remote sensing education and training, only few people have a good knowledge of radar remote sensing and geology, which would be necessary fully analyse the surface motions estimated by persistent scatterer interferometry. Using scientific visualization, data can be presented in an intuitive way for surface motion analysis of non-radar experts. In this paper, we introduce a web-based interactive visualization of PS-InSAR point clouds for education and training.

## 1. INTRODUCTION

Synthetic Aperture Radar (SAR) is an active microwave imaging system. Due to its longer wavelengths, compared to optical systems, it can penetrate clouds and smoke. Because it is an active system, it can operate day and night, although most spaceborne system are on sun-synchronous orbits to ensure sufficient energy supply. Interferometric SAR (InSAR) takes advantage of the coherent nature of microwaves (Ferretti *et al.*, 2007) to obtain height information and to obtain surface motions.

Persistent Scatterer Interferometry (PS-InSAR) (Ferretti *et al.*, 2001) is a commonly used method in radar remote sensing to obtain precise surface motion estimations (Colesanti *et al.*, 2003). However, the interpretation of PS-InSAR results is not always easy (Gehlot *et al.*, 2006) and requires a certain amount of knowledge of the radar backscattering behaviour as well as an understanding of the geological and geomorphological processes involved in the analysed surface motions. Typically, many radar experts lack in the geological and geomorphological knowledge for a precise analysis, while geologists often lack in a deeper understanding of the D-InSAR and PS-InSAR procedures.

Scientific visualization tries to close the gap between disciplines by presenting data in an understandable way and by making the data easily accessible without the need for specialized and sometimes expensive software tools. To allow non-radar-experts to analysis PS-InSAR point clouds, we want to make the visualization intuitive and easy to use. Easy accessibility is one of the major goals, making web-based approaches our primary choice. Web access is available on a wide range of systems, from desktop PCs to tablets and mobile phones. Therefore, using web-technology is guaranteeing a good accessibility.

For the interactive 3D visualization of point clouds, WebGL is our choice, because it is a widely adopted standard running on most modern browsers (Loesch *et al.*, 2012). However, only modern browsers and hardware systems support WebGL at the moment (Carton *et al.*, 2012), unfortunately limiting the accessibility. Nevertheless, we believe this is still the most widely useable approach for 3D visualization and will be supported by more and more systems and vendors in the near future.

In this context, we introduce a web-based interactive visualization for the PS-InSAR point clouds with WebGL for education and training. In the next section, we introduce the PS-InSAR technique briefly. In section 3, we give a brief introduction to WebGL and the process of rendering with WebGL. We introduce the design of our web-based visualization briefly in section 4. In section 5, we present the visualization results using Las Vegas as our test site. In the final section, we draw our conclusions.

## 2. PS-INSAR

### 2.1 PS-InSAR technique

SAR Interferometry can be used to detect of surface motions and for the generation of DEM (Bamler *et al.*, 1998). Differential InSAR (D-InSAR) makes use of phase differences between different acquisitions to estimate the surface motion (Gabriel *et al.*, 1989; Wegmuller *et al.*, 1998); the result can be acquired at a high accuracy while covering relatively large areas, making D-InSAR a very efficient technique in surface motion monitoring.

PS Interferometry, also known as permanent or persistent scatters Interferometry was developed by Ferretti and his colleagues (2001) in the late 1990s, at the Politecnico di Milano, Italy. PS-InSAR was developed to overcome the limitations of deformation monitoring using D-InSAR, since the operational usage of the latter is restricted to short baselines in time and space due to the temporal and geometrical decorrelation (Zebker *et al.*, 1992).

### 2.2 Deformation analyze using PS-InSAR

However, PS-InSAR mainly focuses on a subset of pixels characterized by stable scattering properties; hence, these points are not affected by decorrelation noise (Colesanti *et al.*, 2002). Since these points are distributed discretely on the whole scene, they naturally generate a precise grid, which can gradually eliminate the effect of the atmosphere on the phase (Yao *et al.*, 2008), allowing the information needed for surface motion estimation to be extracted after removing those common decorrelation sources. Thus, PS-InSAR can largely improve the monitoring accuracy, comparing to traditional approach, the

monitoring precision is available to mm-level (Ferretti *et al.*, 2007).

PS point clouds are one of the products of PS-InSAR processing, consisting of PS points extracted from interferograms. The scattering properties of points are very stable, usually seen in artificial structures. That also helps to explain why PS-InSAR works rather well for estimating surface motions in urban areas. On the other hand, PS points may usually demonstrate the structure of the city scene, so visualization of PS points cloud plays an important role in judging the PS-InSAR results.

### 3. CLIENT AND SERVER BASED DESIGN

#### 3.1 Client design

The client has to be a modern, compatible browser. Using HTML5 and JavaScript, a WebGL context is being setup.

##### 3.1.1 WebGL Definition

The Web Graphics Library (WebGL) is part of the HTML5 technologies proposed by the Khronos Group (Anyuru, 2012). As described on the official website: "WebGL is a cross-platform, royalty-free web standard for a low-level 3D graphics API based on OpenGL ES 2.0, exposed through the HTML5 Canvas element as Document Object Model interfaces" (Khronos Group, 2014).

WebGL enables modern internet browsers to render 3D scenes on the web in HTML and JavaScript contexts (Khronos Group, 2013). This allows applications to run on computers, tablet PCs, and even smart phones, as long as a WebGL supported browser can be installed on the device. WebGL provides hardware-based rendering, allowing fast and real-time visualizations. WebGL is designed for dynamic web applications (Parisi, 2012), and examples have been developed to implement 3D interactive visualization using WebGL (e.g., Congote *et al.*, 2011; Birr *et al.*, 2012; Henning *et al.*, 2013).

##### 3.1.2 System Requirements

In a web-based application using WebGL, only a WebGL-supported browser is necessary. There is no need to install plugins or other software. However, not all browsers support WebGL.

##### 3.1.3 Basic Components

WebGL contains different components, and they are used in different applications. To create a 3D scene, the following four components are necessary (Cantor *et al.*, 2012):

**Canvas:** Canvas is a standard element in HTML5. It is a rectangular element in the page where the scene will be rendered. Its position and size can be defined in the HTML document.

**Objects:** Entities in the 3D scene rendered on the canvas in WebGL are defined in JavaScript arrays and stored in the WebGL buffers. In WebGL, complex entities shown on the screen are made up of three basic primitives: points, lines and triangles.

**Camera:** The camera plays the role of the viewport to the 3D world. Parameters of the camera, such as position, view direction, field of view, etc., will determine the final image rendered on the canvas.

**Lights:** In WebGL, vertex and fragment shaders are used to create a lighting model.

#### 3.1.4 Rendering in WebGL

To render objects using WebGL, certain steps are necessary:

- 1) Initialize WebGL
- 2) Load data into the WebGL buffer
- 3) Associate shader attributes to objects
- 4) Initialize the viewport and transformations
- 5) Render the scene to the canvas

#### 3.2 Server design

The design of the web-based visualization is based on a server-client structure. The browser is the client in this model, accessing data from server. Our system is using the ASP .NET MVC 4 framework. The Model-View-Controller (MVC) pattern separates the application into three main aspects: the Model, the View, and the Controller (Freeman, 2012; Galloway *et al.*, 2012). The Model contains classes that describe the data and the rules to change and manipulate the data. The View defines how the user interface looks like. The Controller contains classes to manage the relationship between the View and the Model. ASP .NET MVC4 is a good choice for developing new applications on the .NET platform.

### 4. WEB BASED VISUALIZATION FOR EDUCATION AND TRAINING

Understanding the content of PS-InSAR point clouds is often not easy. There are many reasons for this, like the large amount of data, the often rather high noise levels from different noise sources with different characteristics, as well as the SAR looking and measurement geometry, which also needs to be taken into consideration.

To understand the results, it can be necessary to visualize them in detail. This is normally done in 2D for the PS-InSAR results (Li, 2013). However, to better analyse details, and to get a better understanding of the type of objects forming stable PS points in urban areas, a 3D visualization is preferable. In the third dimension, we can clearly see the PS points and their estimated locations. Even more, it is possible to analyse and understand their spatial relationship.

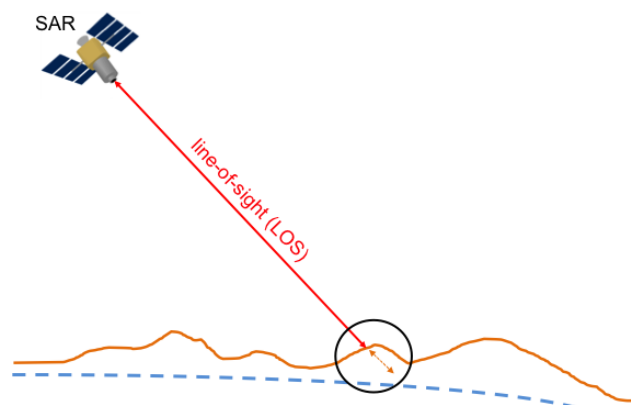


Figure 1. Measurement direction of differential InSAR

A complete visualization of objects in three dimensions requires a dynamic element, allowing, either interactively or pre-scripted, to move the viewpoint and to visualize the objects from different viewpoints. This dynamic element gives us the full three dimensional experience. For training and education, a dynamic visualization is clearly preferable to a pre-scripted animation. The possibility to dynamically change the view, the zoom level, the colour coding, etc., is essential for a deeper understanding of the point clouds structure.

Another common problem in understanding the PS-InSAR results is due to the geometry of the measurement. Differential InSAR methods, with PS-InSAR being one of them, are measuring the surface motion in the line-of-sight of the sensor, i.e. the measured velocity is not in the vertical direction but in the line-of-sight direction, as shown in Figure 1.

However, due to the imaging geometry, not all surface motions can be estimated. Only motion components in the line-of-sight direction are visible and they can be based on horizontal or vertical motions, which are not distinguishable from one measuring direction.

By analysing the velocity information in three dimensions, a better understanding of the underlying surface motion can be achieved, which is essential for a deeper understanding of the PS-InSAR results. A three-dimensional visualization allows us to present the results for non-experts in an understandable way.

## 5. EXPERIMENT AND RESULT

### 5.1 Test Data

For our experiment we use a high-resolution spotlight TerraSAR-X data stack of Las Vegas. Figure 2 shows the test area marked in Google Earth. The data was processed with SARProZ (Perissin *et al.*, 2010) resulting in 610 442 PS points. Each point contains the position, velocity, and coherence information.

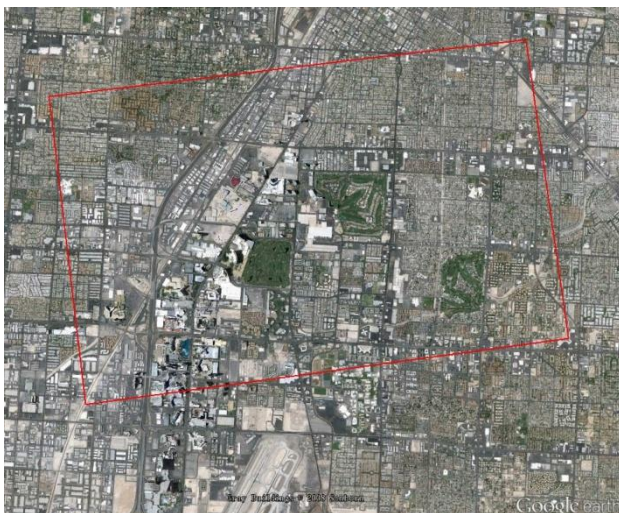


Figure 2. The area of test data in Las Vegas marked with a red rectangle

### 5.2 From 3D to 4D

For a 3D visualization application, the space coordinates are the indispensable elements to set up the virtual world. In addition to space coordinate, a PS-InSAR point cloud contains other information, for instance, velocity and coherence. Including the velocity or coherence information into the 3D visualization changes the problem to a 4D visualization (Weber *et al.*, 2009).

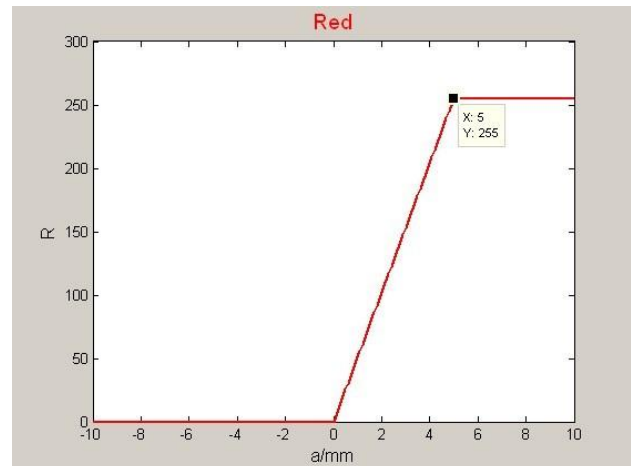


Figure 3. Red colour coding when the deformation velocity ranges from -10mm/y to 10mm/y

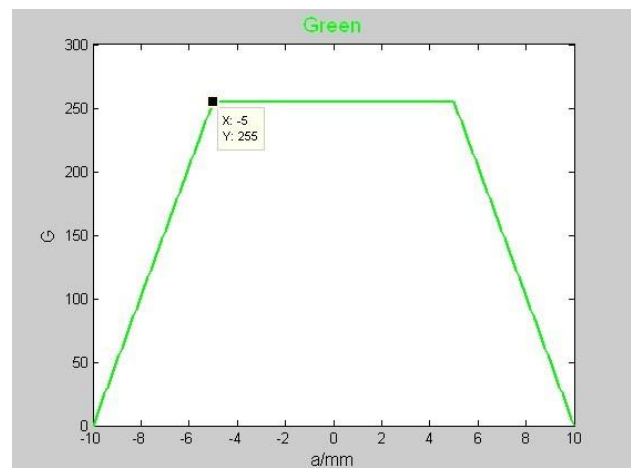


Figure 4. Green colour coding when the deformation velocity ranges from -10mm/y to 10mm/y

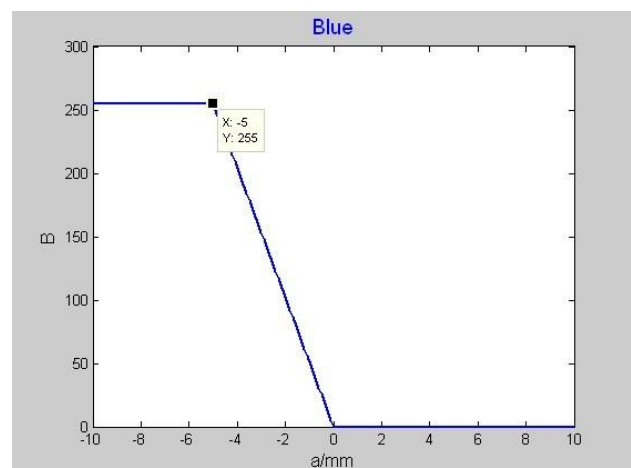


Figure 5. Blue colour coding when the deformation velocity ranges from -10mm/y to 10mm/y

A common method to visualize four-dimensional data is colour coding. In our case study, the velocity information is coded as colour information. The following functions are used to implement the colour coding:

$$Red(v) = \begin{cases} 0, v \leq 0 \\ 255 \frac{v}{0.5m}, 0 < v < 0.5m \\ 255, v \geq 0.5m \end{cases} \quad (1)$$

$$Green(v) = \begin{cases} 255 \frac{v+m}{0.5m}, v \leq -0.5m \\ 255, -0.5m < v < 0.5m \\ 255(1 - \frac{v-0.5m}{0.5m}), v \geq 0.5m \end{cases} \quad (2)$$

$$Blue(v) = \begin{cases} 255, v \leq -0.5m \\ 255(1 - \frac{v+0.5m}{0.5m}), -0.5m < v < 0 \\ 0, v \geq 0 \end{cases} \quad (3)$$

$$m = 10 \quad (4)$$

where  $v$  is the velocity of the PS points; in this example, we select 10mm/y as the threshold for the surface motion velocity. The sign of velocity values represents the direction of surface motion. In our case, a positive velocity value means movement toward the sensor and a negative velocity value means movements away from the sensor.

The three charts above (Figure 3-5) show how to implement colour coding if  $v_{max}=10\text{mm/y}$  and  $v_{min}=-10\text{mm/y}$ .

From above we can conclude that the stable points turn out to be green in the final composited image, because the RGB colour coding is (0, 255, 0) when velocity equal zero. Generally, uplifting points tend to be red and subsiding points tend to be blue.

### 5.3 Results

We pre-process the point clouds to allow fast visualization. As the position of a point is recorded in latitude, longitude, and height, we transform them Universal Transverse Mercator (UTM) Projection first.

After data pre-processing, we store the data, including space position and velocity, in an SQL database. The data is loaded to the WebGL buffers and associated to the shader attributes after the WebGL rendering context has been set up. In our application, the velocity information is visualized via the colour coding method. The velocity ranges between -60 mm/y to 60 mm/y, and the corresponding colour changes from blue to red gradually. We initialize the viewport and transformations in order to display the whole scene on the screen, as shown in Figure 6.

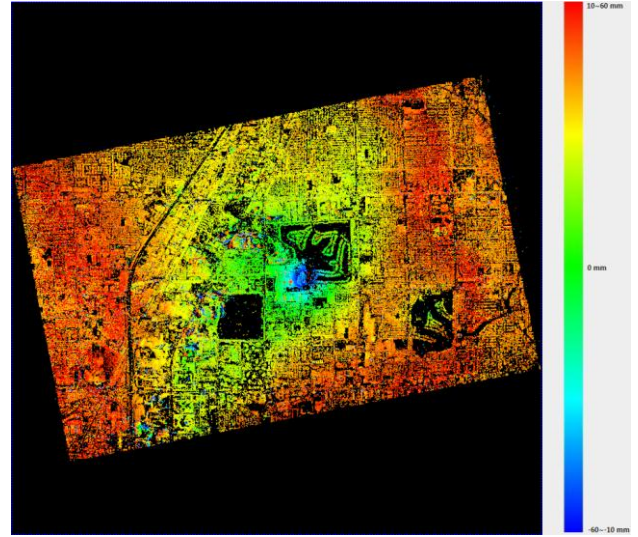


Figure 6. An overview of PS-InAR point clouds of Las Vegas ascending stack

Our web-based interactive visualization is not static, but dynamic. It allows zooming, shifting, and rotating to observe all details. In Figure 7, the Las Vegas Convention Center is shown. The building is a large structure consisting of several simple and regular shapes. The building deformation is shown in different colours.

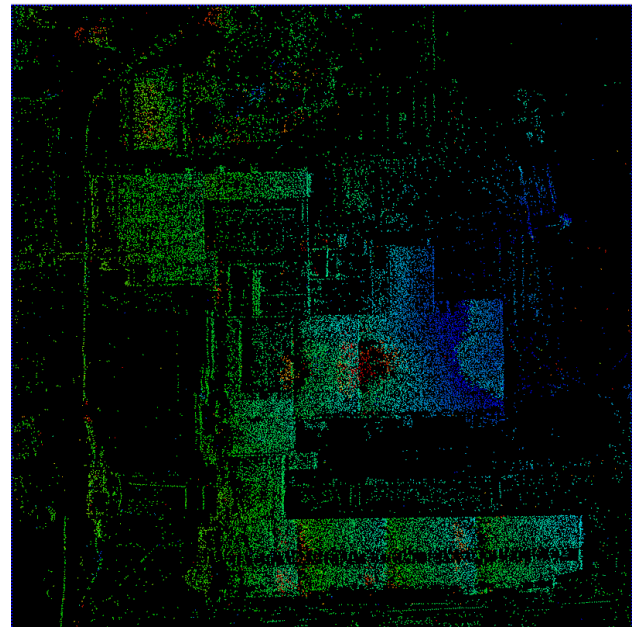
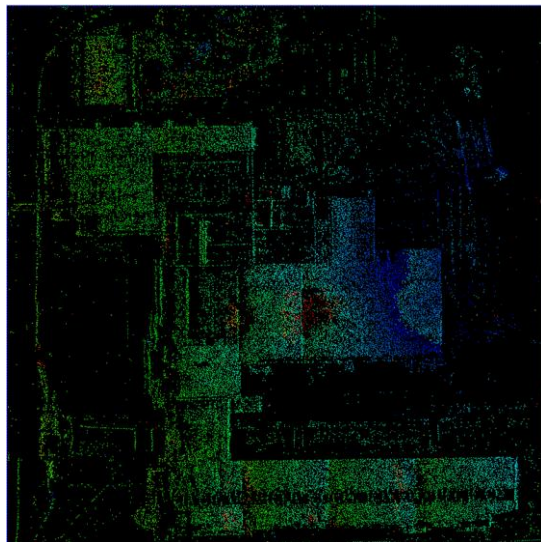


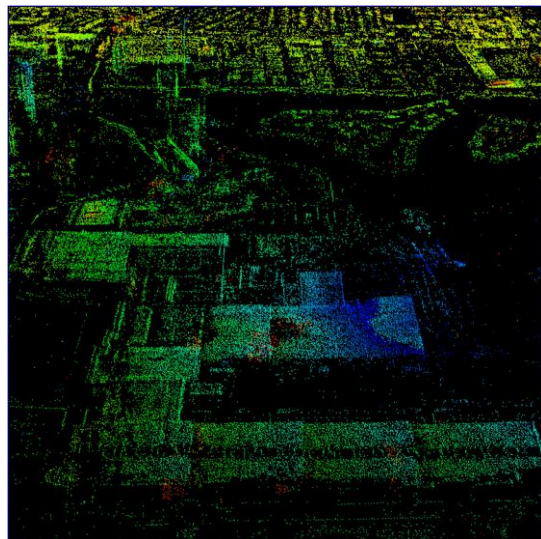
Figure 7. Zoom in onto the Las Vegas Convention Center.

In a static 2D visualization application, objects in the scene only can be seen in the specific view direction. Some objects cannot be seen if they are located behind other objects. Unlike static 2D visualization, our interactive visualization application allows observing the details in the scene from all directions. Figure 8 (a)-(c) presents the visualization results of Las Vegas Convention Center in different view directions.

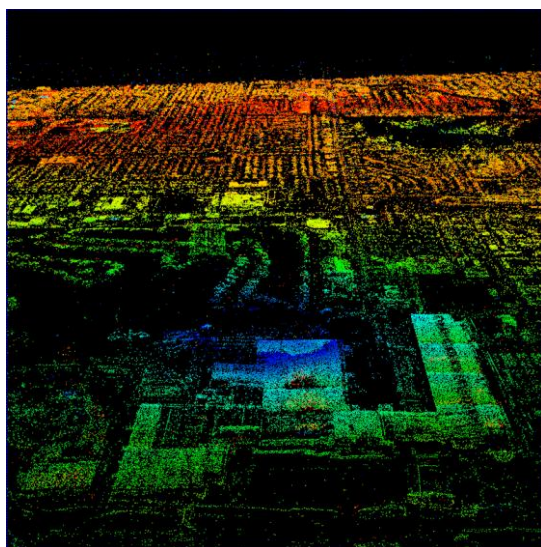




(a)



(b)



(c)

Figure 8. (a) View from the top of the building; (b) Squint from the front of the building; (c) View from the side of the building.

In our 4D visualization application, the velocity information is shown in different colours. In Figure 9, surface motions are clearly presented in the image. We can see blue points on the ground and several points on the top of buildings. The 4D visualization through colour coding helps us find the location where surface motion occurs and gives us a simple glance at intensity of surface deformations.

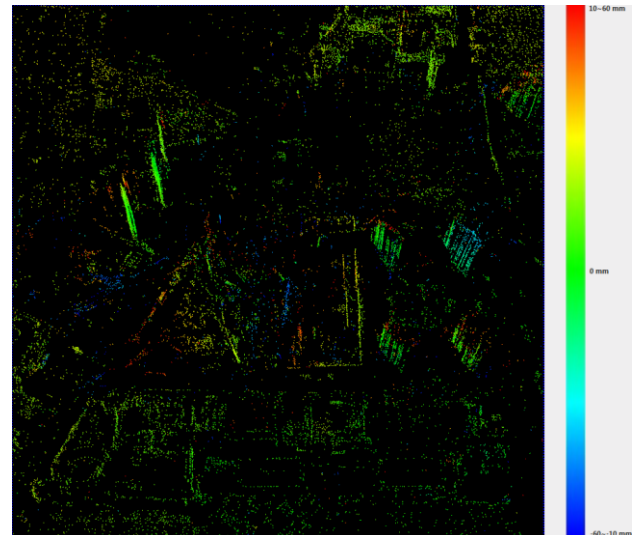


Figure 9. Surface motion velocity shown in different colours.

## 6. CONCLUSIONS

PS-InSAR is well-known remote sensing technique widely used in precise surface motion estimations on large areas. The interpretation of its result, the PS-InSAR point clouds, is usually difficult because it requires knowledge of both: radar remote sensing and geology. We introduced a web-based interactive visualization of PS-InSAR point clouds for education and training. It gives non-experts a deeper understanding of PS-InSAR point clouds and helps to better analyse the results.

Using HTML5 and WebGL, the web-based approaches ensure our visualization application can run free from plug-ins and is available on the Internet through modern browsers on many devices. Besides, the interactive visualization allows us to zoom, shift and rotate the data to observe the details from all directions. By doing so, users will get a rich three-dimensional experience, and it will help non-experts to understand the PS-InSAR results better.

## ACKNOWLEDGEMENTS

The authors would like to thank the German Aerospace Center (DLR) for providing the test datasets via the DLR AO LAN0793. This work is financially supported by the National Natural Science Foundation of China (Grant No. 41174120) and by the International Graduate School of Science and Engineering, Technische Universität München, Munich.

## REFERENCES

- Anyuru A., 2012. *Professional WebGL Programming: Developing 3D Graphics for the Web*. Wrox, USA.
- Bamler, R., Hartl, P., 1998. Synthetic aperture radar interferometry. *Inverse Problems*, 14, pp. 1-54.
- Birr, S., Mönch, J., Sommerfeld, D., Preim, B., 2012. A novel Real-Time Web3D Surgical Teaching Tool based on WebGL. *Bildverarbeitung für die Medizin*, Springer, Berlin Heidelberg, pp. 404-409.
- Cantor, D., Jones, B., 2012. *WebGL Beginner's Guide*. Packt Publishing Ltd, UK.
- Colesanti, C., Ferretti, A., Prati, C., Rocca, F., 2002. Seismic faults analysis in California by means of the permanent scatterers technique. In: *Proceedings of the Third International Symposium on Retrieval of Bio- and Geophysical Parameters from SAR Data for Land Applications*, Sheffield, UK, pp. 125-131.
- Colesanti, C., Ferretti, A., Novali, F., Prati, C., Rocca, F., 2003. SAR monitoring of progressive and seasonal ground deformation using the permanent scatterers technique. *IEEE Transactions on Geoscience and Remote Sensing*, 41(7), pp. 1685-1701.
- Congote, J., Kabongo, L., Moreno, A., 2011. Interactive visualization of volumetric data with webgl in real-time. In: *Proceedings of the 16th International Conference on 3D Web Technology*, ACM, New York, pp. 137-146.
- Freeman, A., 2012. *Pro ASP. NET MVC 4 (Fourth Edition)*. Apress, New York.
- Ferretti, A., Prati, C., Rocca, F., 2001. Permanent Scatterers in SAR Interferometry. *IEEE Transactions on Geoscience and Remote Sensing*, 39(1), pp. 8-20.
- Ferretti, A., Monti Guarnieri, A., Prati, C., Rocca, F., Massonnet, D., 2007. *INSAR Principles: Guidelines for SAR Interferometry Processing and Interpretation*. ESA Publications, The Netherlands, pp. A3.
- Ferretti, A., Savio, G., Barzaghi, R., Borghi, A., Musazzi, S., Novali, F., Prati, C., and Rocca, F., 2007. Submillimeter Accuracy of InSAR Time Series: Experimental Validation. *IEEE Transactions on Geoscience and Remote Sensing*, 45(5), pp. 1142-1153.
- Gabriel, A. K., Goldstein, R. M., Zebker, H. A., 1989. Mapping small elevation changes over large areas: Differential radar interferometry. *Journal of Geophysical Research: Solid Earth (1978-2012)*, 94, pp. 9183-9191.
- Galloway, J., Haack, P., Wilson, B., Allen, K. S., 2012. *Professional ASP. NET MVC 4*. Wiley Publishing, Inc.
- Gehlot, S., Verbree, E., Hanssen, R. F., 2006. Dissemination of PS-InSAR results for improved interpretation and analysis. In: *Proc. ESA FRINGE-05 workshop on Advances in SAR Interferometry from ENVISAT and ERS missions*, Frascati, Italy.
- Henning, M., Gaspers, D., Mertsching, B., 2013. Interactive WebGL-based 3D visualization for situated mathematics teaching. In: *Proc. 2013 International Conference on Information Technology Based Higher Education and Training*, Antalya, Turkey.
- Khronos Group, 2013. WebGL Specification. <http://www.khronos.org/registry/webgl/specs/latest/1.0/> (09.Jan.2014)
- Khronos Group, 2014. WebGL. <http://www.khronos.org/webgl> (09.Jan.2014)
- Li, S. S., Balz, T., Jendryke, M., Liao, M. S., 2013. Web-based interactive visualization of PS-InSAR results from cross-heading orbits. In: *Proc. of the 35th International Symposium on Remote Sensing of Environment*, Beijing.
- Loesch, B., Christen, M., Nebiker, S., 2012. OpenWebGlobe-an open source SDK for creating large-scale virtual globes on a WebGL basis. In: *Proc. ISPRS Congress, The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XXII*, Melbourne, Australia.
- Parisi, T., 2012. *WebGL: Up and Running*. O'Reilly Media, Inc., USA.
- Perissin, D., Wang, Z., Wang, T., 2010. The SARPROZ InSAR tool for urban subsidence/manmade structure stability monitoring in China. In: *Proc. of ISRSE 2010*, Sidney, Australia.
- Weber, B., Mueller, P., Wonka, P., and Gross, M., 2009. Interactive Geometric Simulation of 4D cities. In: *Computer Graphics Forum (Proceedings of Eurographics)*, 28(2), pp. 481-492.
- Wegmuller, U., Wemer, C., Strozzi, T., 1998. SAR interferometric and differential interferometric processing chain. In: *Proc. of IGARSS 1998, Seattle*, 2, pp. 1106-1108.
- Yao, G. Q., M, J. Q., 2008. D-InSAR technique for land subsidence monitoring. *Earth Science Frontiers*, 15(4), pp. 239-243.
- Zebker, H., Villasenor, J., 1992. Decorrelation in interferometric radar echoes. *IEEE Transactions on Geoscience and Remote Sensing*, 30(5), pp. 950-959.