

# SEMANTIC INTERPRETATION OF INSAR POINT CLOUD

*Yuanyuan Wang<sup>(1)</sup>, Xiao Xiang Zhu<sup>(1,2)</sup>*

(1) Helmholtz Young Investigators Group “SiPEO”, Technische Universität München, Arcisstraße 21, 80333 Munich, Germany

(2) Remote Sensing Technology Institute (IMF), German Aerospace Center (DLR), Oberpfaffenhofen, 82234 Weßling, Germany

## 1. INTRODUCTION

This work presents a step towards a better interpretation of the scattering mechanism of different objects and their deformation histories in 3D InSAR point cloud. We trace individual SAR scatterer in high resolution optical images where their geometries, materials, and other properties can be better analyzed.

The proposed approach consists of the following steps:

1. Retrieve the 3D positions of the scatterers from SAR images, i.e. a tomographic SAR inversion.
2. Co-register the 3D point cloud of the SAR scatterers with a reference 3D airborne LiDAR surface model, due to the relative position of the SAR point cloud to the reference point.
3. Co-register high resolution optical image with the reference 3D model, so that each scatterer can be traced in the optical image.
4. Classify the optical image pixels based on its semantic meaning, e.g. geometry, material, and so on.
5. Analyzing 3-D objects in the point cloud based on their semantic class.

## 2. TOMOGRAPHIC SAR INVERSION

We make use of the D-TomoSAR software *Tomo-GENESIS* [1], [2] developed in DLR to process TerraSAR-X image stacks. For an input data stack, Tomo-GENESIS retrieves the following information:

- the number of scatterers inside each pixel,
- the scattering amplitude and phase of each scatterer,
- their 3D positions and motion parameters, e.g. linear deformation rate and amplitude of seasonal motion.

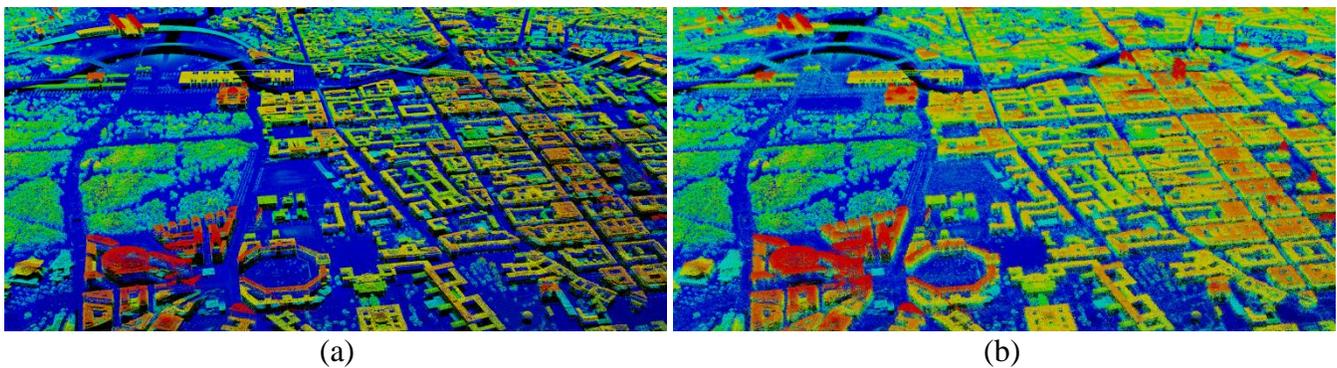
The scatterers’ 3D positions are geo-coded into Universal Transverse Mercator (UTM), so that the results from multiple data stacks with different viewing angles can be combined. For our test area Berlin, two image stacks – one ascending orbit, the other descending orbit – are processed. These two point clouds are fused together, using a feature-based matching algorithm which estimates and matches common building edges in the two point clouds [3].

## 3. COREGISTRATION OF TOMOSAR POINT CLOUDS AND REFERENCE MODEL

The reference model is a 3D point cloud from an airborne LiDAR sensor [4], which is represented, same as the TomoSAR point cloud, in the UTM coordinate system. And hence, the co-registration problem is the estimation of translation between two rigid point clouds, subject to a certain tolerance on rotation and scaling. However, our LiDAR point cloud is nadir-looking, in contrast to the side-looking geometry of SAR. In another word, façade point barely appears in LiDAR point cloud while it is prominent in TomoSAR point cloud. These unique modalities have driven our algorithm developed in the following way:

1. Edge extraction
  - The reference point cloud is rasterized into a 2D height image.
  - The point density of TomoSAR point cloud is estimated on the rasterized 2D grid.
  - The edges in the LiDAR height image and the TomoSAR point density image are extracted.
2. Initial alignment
  - Horizontally by cross-correlating the two edge images.
  - Vertically by cross-correlating the height histogram of the two point clouds.
3. Refined alignment
  - The façade points in both point clouds are removed.
  - The final solution is obtained using iterative closest point (ICP).

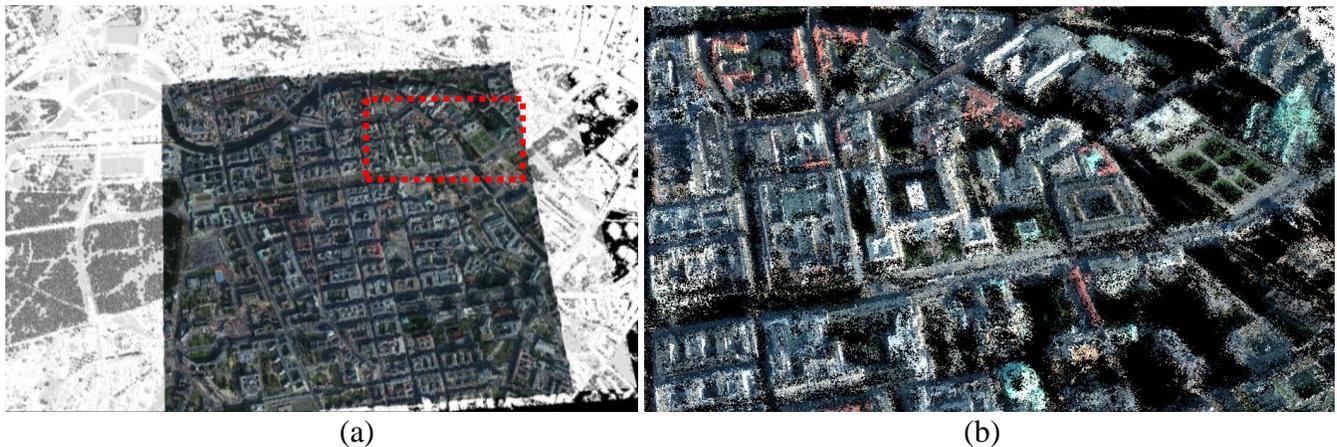
Figure 1 shows the reference model, and the fused one. Successful co-registration can be confirmed by seeing the correct location of the façade points in subfigure (b).



**Figure 1. (a) Close up of the reference LiDAR point cloud in downtown Berlin, and (b) the co-registered point cloud combining the TomoSAR and LiDAR point cloud.**

#### 4. COREGISTRATION OF OPTICAL IMAGE AND REFERENCE MODEL

The optical image was already co-registered with the reference model, hence the TomoSAR point cloud. See the following figure. Such textured point cloud enables the analysis of the SAR point cloud based on the features in optical image.



**Figure 2. (a) TomoSAR point cloud texture with RGB color from optical image, where the white background is the TomoSAR point cloud that is not covered by the optical image, and (b) close up of the dashed red rectangle in (a).**

## 5. SEMANTIC CLASSIFICATION OF OPTICAL IMAGE

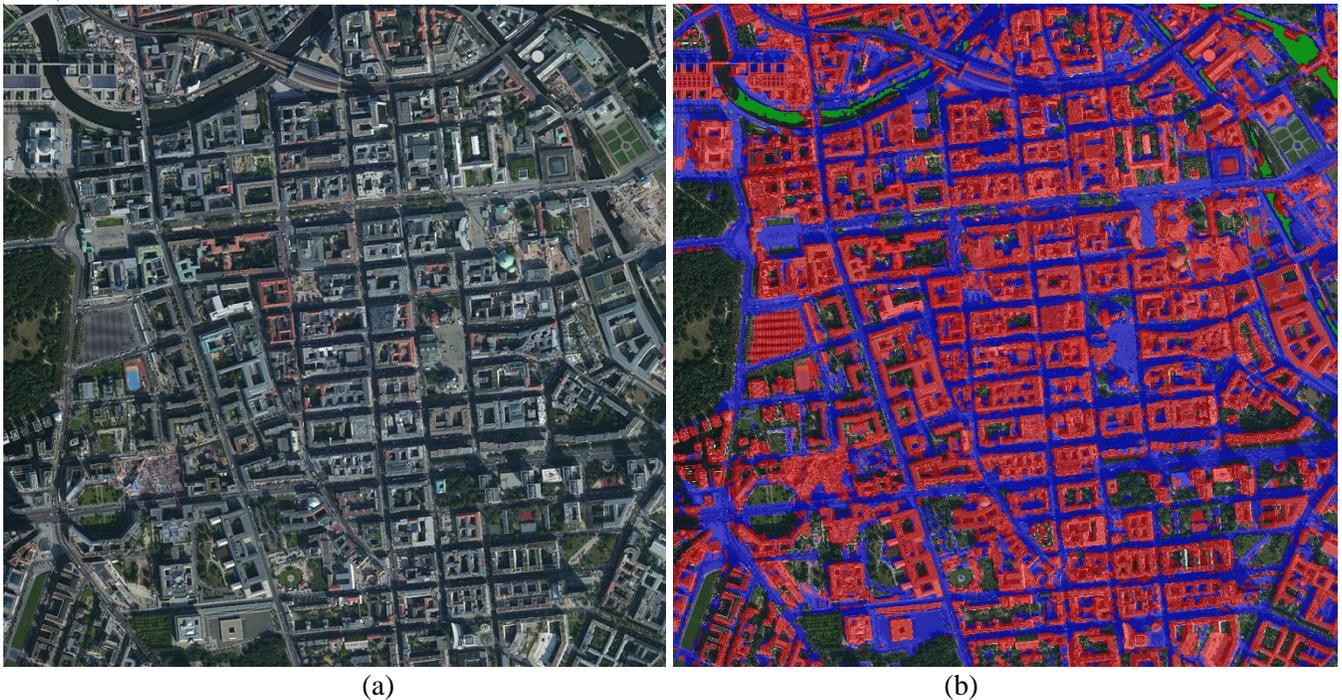
The classification is done patch-wised using a dictionary-based algorithm. The workflow is as follows.

1. Dense features are extracted using a sliding window on each patch. Collecting the feature vectors of all the patches should result in a large number of them.
2. The ensemble of the feature vectors should already give a dictionary  $\mathbf{D}$ . However, due to the sheer size of such dictionary, it is discretizing using  $k$ -mean clustering. The cluster center is extracted as the final dictionary atoms, yielding  $\mathbf{D} \in \mathbb{R}^{N \times k}$ , where  $N$  is the dimension of the feature space.
3. Each patch is then described using the occurrence of the dictionary atoms.
4. A support vector machine is trained as the classifier using samples manually picked from the patches.

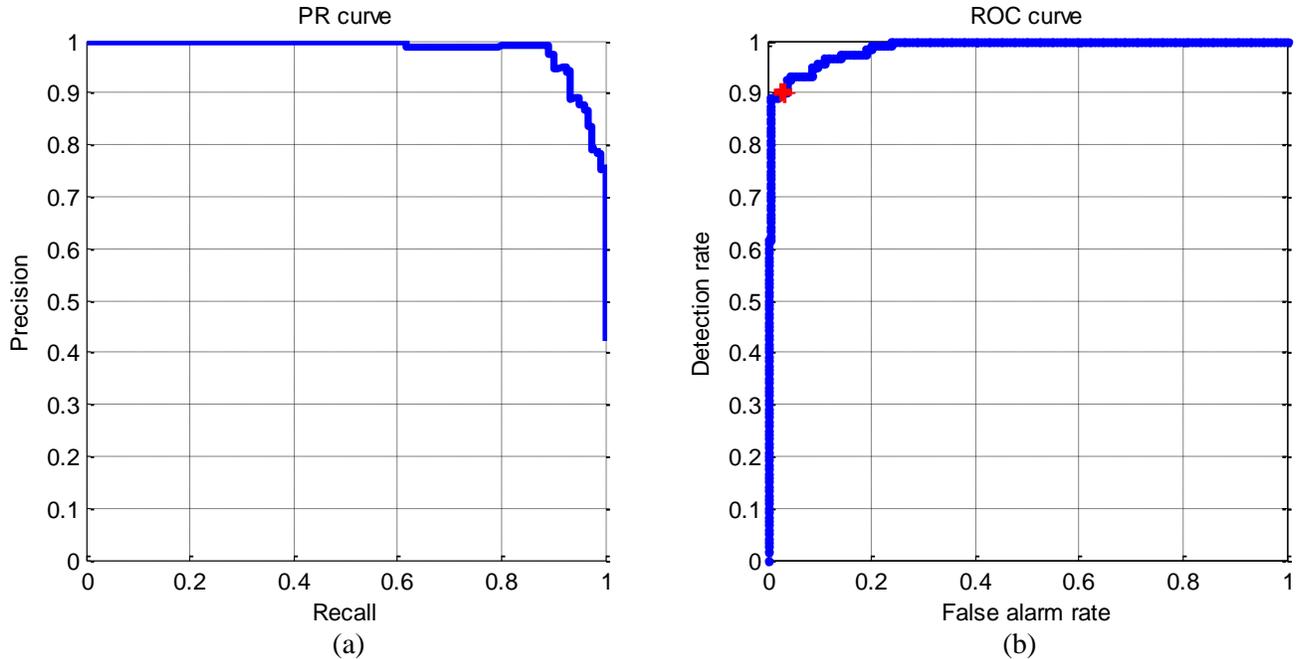
This kind of framework is well explained in [5] as Bag of Words.

The classification is done using a standard support vector machine (SVM) from an open source library VLFeat [6]. For our test image of  $5000 \times 5000$  pixels, it is tiled into patches of  $50 \times 50$  pixel, with 46 pixel overlap. That is to say, the classification of each patch is only assigned to the  $4 \times 4$  pixel in the center. Among all the patches, 570 are manually selected as training samples. Four classes are preliminarily defined: building, roads/rail, river, and vegetation, with each one has 240, 159, 39, and 132 training patches, respectively. The feature in our experiment is simply the vectorized RGB pixel values in a  $3 \times 3$  sliding window, which results in a feature space of 27 dimension. Figure 3 shows the classification result, where the left image is the optical image, and in the right image, classified building, road, river, and vegetation are marked as red, blue, green, and blank. Despite the extremely simple feature we used, the four classes are very well distinguished.

Since building is of our particularly interest, its classification performance is evaluated by classifying half of training samples using the SVM trained with the other half. The average precision of the current algorithm is 98%. The full precision and recall curve is plotted in Figure 4(a). The equivalent receiver operating characteristic curve is also shown in Figure 4(b). The red cross marks our decision threshold which gives a detection rate of 90%, and false alarm rate of 3%.



**Figure 3. (a) the test optical image, and (b) the classification of building, road, river, and vegetation, where they are colored in red, blue, green, and blank.**



**Figure 4. (a) precision and recall curve of the building classification with an average precision is 98%, and (b) the ROC curve of the classification. The red cross marks our decision point which gives a detection rate of 90%, and false alarm rate of 3%.**

## 7. REFERENCES

- [1] X. Zhu, *Very High Resolution Tomographic SAR Inversion for Urban Infrastructure Monitoring: A Sparse and Nonlinear Tour*, vol. 666. Deutsche Geod äische Kommission, 2011.
- [2] X. Zhu, Y. Wang, S. Gernhardt, and R. Bamler, "Tomo-GENESIS: DLR's Tomographic SAR Processing System," in *Urban Remote Sensing Event (JURSE), 2013 Joint*, 2013, pp. 159–162.
- [3] Y. Wang and X. X. Zhu, "Automatic Feature-based Geometric Fusion of Multi-view TomoSAR Point Clouds in Urban Area," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. PP, no. 99, 2014.
- [4] Data provided by "Land Berlin" and "Business Location Service", supported by "Europ äischer Fonds für Regionale Entwicklung".
- [5] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *Workshop on statistical learning in computer vision, ECCV, 2004*, vol. 1, pp. 1–2.
- [6] B. F. Andrea Vedaldi, "VLFeat: an open and portable library of computer vision algorithms.," in *Proceedings of the 18th International Conference on Multimedia 2010*, Firenze, Italy, 2010, pp. 1469–1472.