

# Processing of Large-size InSAR Images: Parallel Implementation of Inverse Vortex Phase Field Algorithm

Andrey V. Sosnovsky and Victor G. Kobernichenko

Ural Federal University, Yekaterinburg, Mira st., 19, Russia,  
sav83@e1.ru

**Abstract.** A parallel implementation of the inverse vortex phase field algorithm is proposed for unwrapping the large-size interferograms obtained by the space synthesized aperture radars. It is shown that the parallelization of calculations in solving the unwrapping problem by this method is possible due to the fact that the phase discontinuity compensation operations are linear. So, the computations can be carried out in any sequence and they can be arbitrarily distributed among the individual computing devices. The method of interferogram splitting into blocks of arbitrary size (not exceeding the memory size of computing devices) is also proposed, and it allows solving the problem of unwrapping for the large-size interferograms. The results of processing the ALOS PALSAR interferograms with the size of  $14 \times 3$  thousands elements on a quad-core computing device are presented.

**Keywords:** InSAR phase unwrapping, parallel algorithms, remote sensing data processing

## 1 Introduction

Digital elevation models (DEM) and displacement maps are widely used in various scientific and technical areas, such as cartography, geodesy, geology, environmental monitoring of mining areas, monitoring the transport communications, *etc.* [1–5]. The radar remote sensing is performed by interferometric SAR techniques (InSAR and DInSAR) that allows one to obtain both types of elevation data with the semi-automatic data processing, which makes it very attractive for use in these tasks. However, the phase unwrapping stage of interferometric processing converts a relative phase defined on the interval  $[-\pi, \pi]$  into an absolute phase. The latter is approximately proportionally related to the surface topography (for InSAR) or the relief displacement (DInSAR). So, the unwrapping stage is the obvious “bottleneck” of the whole radar interferometry. Existing algorithms for its solving are generally based on application of the optimization techniques (Minimal Cost Flow, Integer optimization), searching for the optimal integration path (Goldstein residue cut), on solution of large systems of equations (least square method), *etc.* Such techniques have low computational efficiency

(typically quadratic complexity) and are difficult for the parallel execution. Also, the most part of the existing methods is aimed to building the absolute phase field, which is congruent to the relative phase field. But it is actually useless in terms of the side-looking radar geometry, which causes an irreversible damage to the local data parts.

Phase discontinuity is an element of the relative (wrapped) interferometric phase, which leads to dependence of the absolute (unwrapped) phase on the integration contour shape, and, so, the absolute phase cannot be restored uniquely. It is possible to identify the following elements of the discontinuity: two discontinuity points (so-called residues), where the cumulative sum on elementary path (4 adjacent elements) of the phase gradient is not equal to zero, and a discontinuity line, which virtually connects these points. It is possible to allocate all discontinuity points by calculating the residue function for the whole interferogram, but not the discontinuity line. This fact makes the solution of unwrapping problem ambiguous. As it was shown earlier [11], an inverse vortex phase field algorithm for phase unwrapping allows one to obtain a stable continuous solution of the problem, and it may be suitable for the parallel and block implementation for the large-size data processing.

## 2 Inverse vortex phase field method for phase unwrapping

Inverse vortex phase field method (IVPF) uses an artificial phase discontinuity (phase singularity) of the opposite sign (pseudo-discontinuity) to compensate the interferogram phase discontinuity. The most widespread type of the phase discontinuity is the layover, which can be described as an angle of rational function 1 in a complex domain

$$\dot{I}(z_{m,n}) = \exp \left\{ j \arg \left[ \frac{z_{m,n} - z_{01}}{z_{m,n} - z_{p1}} \right] \right\}, \quad (1)$$

where  $z_{01}$  and  $z_{p1}$  are the coordinates of the discontinuity points on the interferogram,  $z_{m,n} = m + jn$  is the complex coordinate variable.

Such discontinuity may be compensated by two artificial phase vortices  $\dot{C}_0(z_{m,n})$  and  $\dot{C}_p(z_{m,n})$  with centers at the points  $z_{01}$  and  $z_{p1}$  that have inverse directions

$$\begin{aligned} \dot{C}_0(z_{m,n}) &= \exp \left\{ j \cdot \arg \left[ \frac{1}{z_{m,n} - z_0} \right] \right\}, \\ \dot{C}_p(z_{m,n}) &= \exp \{ j \cdot \arg [z_{m,n} - z_p] \}, \end{aligned} \quad (2)$$

where  $\dot{C}_0(z_{m,n})$  and  $\dot{C}_p(z_{m,n})$  are the inverse phase vortices.

A new interferogram  $\dot{I}_c(z_{m,n})$  is formed after the compensation of all phase discontinuities

$$\dot{I}_c(z_{m,n}) = \dot{I}(z_{m,n})\dot{C}_0(z_{m,n})\dot{C}_p(z_{m,n}). \quad (3)$$

As it was determined earlier by experiments [11], there is no necessity to identify pairs of discontinuity points, and all such points may be processed separately in any order. The SAR interferograms usually contain the multiple chaotic

located discontinuities. So, it is essential, at first, to localize them by the residues function, and then form the product of the complex images of the inverse vortices for each point

$$\dot{P}(z_{m,n}) = \dot{C}_{01}(z_{m,n})\dot{C}_{02}(z_{m,n}) \cdot \dots \cdot \dot{C}_{0M} \cdot \dot{C}_{p1}(z_{m,n})\dot{C}_{p2}(z_{m,n}) \cdot \dots \cdot \dot{C}_{pN}, \quad (4)$$

where  $\dot{P}(z_{m,n})$  the the inverse vortex phase field. A dot product of the complex interferogram and inverse vortex phase field

$$\dot{I}_c(z_{m,n}) = \dot{I}(z_{m,n})\dot{P}(z_{m,n}) \quad (5)$$

forms the corrected interferogram  $\dot{I}_c(z_{m,n})$ , where the phase ambiguity is not obligatory fully resolved because the inverse vortices may lead to occurrence of new phase discontinuities or to moving it into a new position. The procedure should be iteratively repeated until all ambiguities to be fully resolved.

For elementary discontinuities, such correction is unwanted because the application of the inverse vortex will here lead to additional distortion of the unwrapped phase. On the other hand, a simple zeroing of such discontinuities instead of inverse vortex correction does not produce an unwrapping error and allows one to increase the computational speed.

So, the algorithm for the phase unwrapping will include the following steps:

1. generation of interferogram residues function —  $R_{m,n}$ ;

If  $R_{m,n} = 0$  for all  $(m, n)$ , then go to step 7;

2. detection of elementary discontinuities  $\{W_e(m, n)\}$  according to the criterion of the 8-neighbourhood of two opposite-signed residues, and their correction by phase zeroing;

3. regeneration of interferogram residues function  $R_{m,n}^c$ ;

If  $R_{m,n}^c = 0$  for all  $(m, n)$ , then go to step 7;

4. generation the inverse vortex phase field for remaining residues points  $\{z_{01}, z_{02}, \dots, z_{0M}\}$  and  $\{z_{p1}, z_{p2}, \dots, z_{pN}\}$

$$\dot{P}(z_{m,n}) = \exp \left( j \cdot \arg \frac{(z_{m,n} - z_{p1}) \cdot (z_{m,n} - z_{p2}) \cdot \dots \cdot (z_{m,n} - z_{pn})}{(z_{m,n} - z_{01}) \cdot (z_{m,n} - z_{02}) \cdot \dots \cdot (z_{m,n} - z_{0n})} \right); \quad (6)$$

5. interferogram correction by the inverse vortex phase field

$$\dot{I}(z_{m,n}) \rightarrow \dot{I}(z_{m,n}) \cdot \dot{P}(z_{m,n}); \quad (7)$$

6. regeneration of the interferogram residues function —  $R_{m,n}^{c''}$ ;

If  $R_{m,n}^{c''} \neq 0$  for all  $(m, n)$ , then go to step 4, else go to step 7;

7. simple phase unwrapping by the linear path.

### 3 Improving the computational efficiency for the inverse vortex phase field method

Steps 4–5 of the algorithm have the most computational complexity, but their performance may be improved by the following measures.

1. Multiplication of the complex functions  $\dot{I}(z_{m,n})$  and  $\dot{P}(z_{m,n})$  may be unambiguously replaced by summation of their arguments, and the arguments can be evaluated once before the first iteration.

2. The inversed vortex  $P^1(z_{m,n})$  for a single point can be computed once for the field of  $2M \times 2N$  size and saved into the processing device memory, and on step 4, the vortex fragment of  $M \times N$  size with the centre at the point corresponding to the zero  $z_{0i}$  or the pole  $z_{pi}$  can be simply read from the memory.

3. Despite the algorithm is global, the inverse vortex phase field is constructed by independent repeated complex multiplications (or additions in the relative phase domain) of the deterministic function. So, it can be easily implemented by the a parallel execution of computing devices (within one iteration).

Computations of residues function on step 1–3 and 6 are local. Therefore, it can be distributed by different computational devices. Computations for steps 4–5 must be made on the interferogram of the original size, but discontinuities may be passed in any order by different computational devices, and the partial results may be summarized.

But the following problem remains. Usually, raw SAR interferograms (without data cropping and multilooking) have sizes of thousands elements by each side, or more than 100 million (sometimes, more than billion) pixels in the complex single-precision format (more 1–5 GB or more). During the algorithm execution, these data may be reproduced for several times, requiring big RAM volumes. So, to make the IVPF applicable to the large-size interferograms, the following implementation strategy (the block-IVPF algorithm) may be offered.

1. Splitting the whole interferogram by the blocks  $B_i$  of relatively small size (for example,  $1000 \times 1000$  elements), that are easily processed by a single processing unit. The right-sided and bottom blocks may have the distinctive size.

2. Localization of the discontinuity points in the  $B_j - th$  block.

3. Processing the  $B_i$  block:

- generation of the inversed vortex of the  $B_j$  block (double-sized respectively to the block size) in the borders of  $B_i$  block by the suitable part of (2), Fig. 1a;
- application of the  $B_j$  inverse vortex of a  $j_m$  point by vortex border shifting (Fig. 1b);

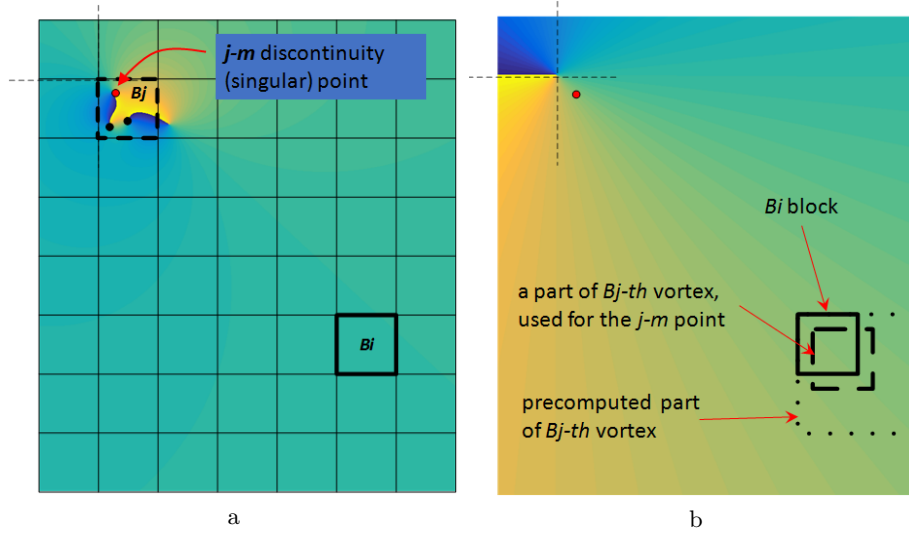
– repeat the step until all points in  $B_j$  block the are processed;

4. Repeat step 3 until all  $i - j$  pairs are processed.

5. Block merging.

6. Localisation of the discontinuity points. If any discontinuity points remain, then return to the step 1, else, stop the algorithm.

Execution of step 3 may be implemented on the parallel computing devices, because, as it mentioned earlier, all operations in the algorithm are commutative and may be executed independently on each other. For the parallel execution, it is required to transmit the block data (or group of blocks data) and all discontinuity points coordinates to the elementary device and to transmit the processed block back to the central device for merging.



**Fig. 1.** Block processing for the  $i - j$  pair; a) Interferogram splitting into blocks ( $B$ ); b) Inverse vortex for the  $B_j$ -th block and the border for the  $B_i$ -th block

## 4 Experimental results

Experimental imagery data was received by the ALOS PALSAR radar and are represented by an interferometric pair of SAR images for the polygon contained surfaces of different types (fields, forests, urban, and mining areas). Data samples were obtained in dual polarization mode, and the HH-polarization was used. The phase field has an ambiguity height of 17.2 m (at the near stage); the spatial sampling interval was  $15.0 \times 3.1$  m. Two data subsets were used for the unwrapping experiments

1. scene with the size of  $3000 \times 3000$  samples;
2. scene with the size of  $14000 \times 3000$  samples (Fig. 2a).

The block-IVPF algorithms was implemented in the MATLAB environment. It includes the main program and two sub-programs, which have 65 code strings in total, only. The codes were executed on a system with the Intel Core-i5 processor with 2.67 GHz clock frequency, the RAM size was 12 GB.

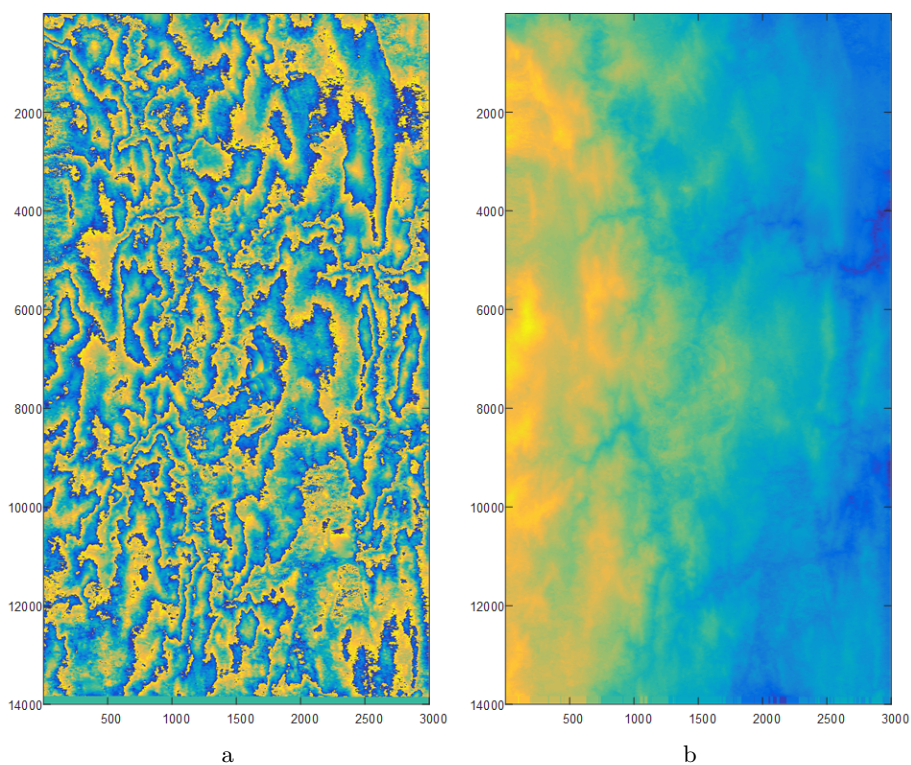
The first scene was processed by the both initial and block versions of the IVPF, the interferogram was splitted into 16 blocks in the block version. The results turned out to be equal, the computational time was 420 sec for the initial version and 730 sec for the block version. The last was slower, because only single computing device was used. Both algorithms converged in 6 iterations. The accuracy of the unwrapping result was earlier proved (for the initial version) in [9, 10]. The second scene (Fig. 2b) was processed by the block-version algorithm only, because the initial version, as well as the Minimum cost flow and Region growing algorithms (in SARscape implementation) fail causing “Out of memory”

error; the computational time was about 6 hours, the algorithm also converged in 6 iterations.

## 5 Conclusion

The block implementation for the Inverse vortex phase field algorithm for the InSAR data processing was proposed. Two sets of the ALOS PALSAR data were processed, and it was shown that the algorithm gives the same results, as the initial algorithm version. The algorithm allows one to process large data sets, which are unsupported by the initial algorithm and some other processing algorithms. The algorithm has very compact code and is suitable for execution on parallel computing systems. It is because the algorithm uses the commutative computational operations, and the data blocks may be randomly distributed between the computational devices.

**Acknowledgments.** The work was supported by the Act 211 Government of the Russian Federation, contract A 02.A03.21.0006.



**Fig. 2.** An ALOS PALSAR interferogram unwrapping results by the IVPF algorithm; a) original filtered interferogram; b) IVPF-processed absolute phase

## References

1. Elizavetin, I. V., Ksenofontov, E. A.: Resultaty experimentalnogo issledovaniya vozmozhnosti pretsizionnogo izmereniya reliefa Zemli interferentsionnym metodom po dannym kosmicheskogo RSA [The results of experimental research of precious Earth relief measurement by interferometric method with space-based SAR]. *Issledovaniya Zemli iz kosmosa*. 1, 75–90 (1996)
2. Joughin, I. R. , Li, F. K., Madsen, S. N., Rodrigues, E., Goldstein, R. M. Synthetic Aperture Radar Interferometry. *IEEE Proc.*, 88(3), 333–382. (2000).
3. Hanssen, R. F.: *Radar Interferometry. Data Interpretation and Error Analysis*. Dordrecht. Kluwer (2001)
4. Dorosinskiy, L. G. Radar signals class recognition algorithm synthesis. *CRIMICO'2014 proceedings*, 24(1), 1137–1138 (2014)
5. Dorosinskiy, L. G. Synthesis and analysis of radar signal classification algorithms. *International Journal of Pure and Applied Mathematics*. 109(3), 681–689. (2016)
6. Aoki, T.; Sotomaru, T.; Ozawa, T.; Komiyama, T.; Miyamoto, Y.; Takeda, M. Two-dimensional phase unwrapping by direct elimination of rotational vector fields from phase gradients obtained by heterodyne techniques. *Opt. Rev.* 5, 374–379 (1998)
7. Costantini, M. A novel phase unwrapping method based on network programming, *IEEE Trans. Geosci. Remote Sensing*, 36, 813–821 (1998)
8. Heshmat, S., Tomioka, S., Nishiyama, S. Performance Evaluation of Phase Unwrapping Algorithms for Noisy Phase Measurements. *International Journal of Optomechatronics*, 8 (4), 260–274 (2014)
9. Sosnovsky, A., Kobernichenko, V. A technique for evaluation of InSAR processing stages efficiency. *CRIMICO'2017 proceedings*, 26(2), 2716–2722. (2016)
10. Sosnovsky, A., Kobernichenko, V. An accuracy estimation of digital elevation models obtained by interferometric synthetic aperture radars. *XXII international conference Radiolocatsiya, navigatsiya, svyaz' (RLNC'2016)*, Voronezh, Russia, 1074-1081. NPF SAKVOEE (2016)
11. A. V. Sosnovsky, V. G. Kobernichenko. An InSAR phase unwrapping algorithm with the phase discontinuity compensation. *Proceedings of the 2nd International Workshop on Radio Electronics & Information Technologies*. Yekaterinburg, Russia, November 15, 2017. *CEUR-WS*, 2005. 127–136. (2017).