European Society of Computational Methods in Sciences and Engineering (ESCMSE)

Journal of Numerical Analysis, Industrial and Applied Mathematics (JNAIAM)
vol. 19, no. 1-2, 2026, pp. 39-47
ISSN 1790–8140

Parallel Algorithm for Solving the Phase Unwrapping Problem in InSAR Data Processing

E. N. Akimova[1,a,b)], **V. E. Misilov**[a,b)], **A. V. Sosnovsky**[b)]

[a)] Krasovskii Institute of Mathematics and Mechanics,
S. Kovalevskaya Street 16, Ekaterinburg 620077, Russia
[b)] Institute of Radioelectronics and Information Technology,
Ural Federal University, Mira Street 19, Ekaterinburg 620002, Russia

Received October, 2025; accepted in revised form October, 2025

Abstract: This work is devoted to the development of an efficient parallel algorithm for solving the problem of phase unwrapping in processing of interferometric radar data in the remote sensing of the Earth. The algorithm is based on the inverse vortex field method. A parallel algorithm is implemented for multicore central processors. Performance and efficiency of the code are studied in the experiments on processing the real data. A speedup of 2 times was achieved using a lower precision arithmetic. A total speedup of 19 times was achieved on a 16-core processor.

Keywords: remote sensing of the Earth; interferometric synthetic aperture radar systems; phase unwrapping; parallel algorithm; OpenMP

Mathematics Subject Classification: 65Y05; 78A55

PACS: 84.40.Xb; 02.60.-x

1 Introduction

Synthesized aperture radar systems are used for solving remote sensing of Earth applied problems. Interferometric processing (InSAR) of the synthesized aperture radar data is applicable for the Earth's relief information extraction [1, 2]. InSAR interferograms are widely used in the problems of creating and updating maps of the relief of the Earth's surface in geodesy, cartography, environmental monitoring, geological, hydrological and glaciological studies, and for monitoring transport communications. The phase unwrapping problem had been studied for last 40 years, and more than several dozens of different algorithms have been developed.

Modern radar systems have ultra-high spatial resolution and a wide band, which leads to the need to unwrap large interferograms consisting of tens millions of elements. In a strict formulation, the phase unwrapping problem is NP-difficult problem, which makes it difficult or impossible to perform interferometric processing of large-size interferograms (more than several thousand elements on each side). Various algorithms have been developed for phase unwrapping [3]. Some

[1]Corresponding author. E-mail: aen15@yandex.ru

methods utilize too many simplifications and approximations, leading to low accuracy. For example, the region growing algorithm have linear complexity but may leave large artifacts in the resulting image. The least squares method or Green's function method are easy for parallel implementations, but have low accuracy. More accurate methods have high computational complexity and may require several days to process large interferograms. The minimum cost flow is the most common algorithm nowadays. It provides the close to optimal solution but has at least quadratic computational complexity and is difficult for parallel implementation.

To obtain acompromise between accuracy and complexity, the inverse vortex field method was previously proposed [3, 4, 5]. It gives a satisfactory accuracy, quasi-linear complexity, and it is suitable for parallelization.

In this work, we have implemented the inverse vortex field method as a parallel multi-threaded code for multicore processors using OpenMP technology. We have also studied the viability of using the lower bit computations in implementation. We have compared the accuracy of results obtained with double precision (FP64) and single precision (FP32) and found no significant degradation in the accuracy of the unwrapped model interferogram.

2 Problem Statement

The main problem of interferometric radar remote sensing of the Earth from space data processing is phase unwrapping procedure. Phase unwrapping is a process of transforming a two-dimensional relative phase pattern, which takes values only in the interval $[-\pi, \pi)$, into an absolute phase. Phase unwrapping is still the most problematic stage of interferometric processing, since such a problem does not have an analytical solution and is solved approximately.

Normally, the interferogram phase field $\Delta\phi(x, y)$ is potential, and a rotor of its gradient is equal to zero:

$$\nabla \times (\nabla(\Delta\phi(x, y))) = 0, \tag{1}$$

where x, y are spatial coordinates. Phase unwrapping for such case implies integration of the phase gradient:

$$\Psi(x, y) = \iint_{(D)} (\nabla(\Delta\phi(x, y)))dxdy, \tag{2}$$

where D is the interferogram area. All imaging radars use a discrete coordinate system, so the phase interferogram is represented as a lattice function $\delta\phi_{m,n}$, where m is an along-track coordinate ("azimuth"), and n is a cross-track coordinate ("range"). In such situation, the algorithm of integration can be written as following:

$$\begin{aligned} &\delta\phi_{m,n} = \mathrm{W}\left\{\Delta\phi_{m,n} - \Delta\phi_{m-m_i,n-n_j}\right\}, \\ &\Psi_{m,n} = \delta\phi_{m,n} + \delta\phi_{m-m_i,n-m_j}, \end{aligned} \tag{3}$$

where W is the wrapping operator, $m_i = 1, n_j = 0$ for unwrapping in vertical direction (by columns), or $m_i = 0, n_j = 1$ for unwrapping in horizontal direction (by rows).

However, under the influence of some physical phenomena (phase noise, layover etc) lead to appearance of phase discontinuities (phase gaps, singular points) on phase interferogram. Phase noise appears on the water and forest areas (see Fig. 1) and the layover effect appears at the areas of the elevation gradients (see Fig. 2).

The phase discontinuity, as a rule, is a certain line of an arbitrary trajectory, at the ends of which the condition 2 is violated. To localize such points, a dicreete residue function $Q_{m,n}$ is calculated by the following algorythm:

$$\begin{aligned} Q_{m,n} = &\mathrm{W}\{\Delta\phi_{m,n} - \Delta\phi_{m-1,n}\} + \mathrm{W}\{\Delta\phi_{m-1,n} - \Delta\phi_{m-1,n-1}\} + \\ &+ \mathrm{W}\{\Delta\phi_{m-1,n-1} - \Delta\phi_{m,n-1}\} + \mathrm{W}\{\Delta\phi_{m-1,n} - \Delta\phi_{m,n}\}, \end{aligned} \tag{4}$$

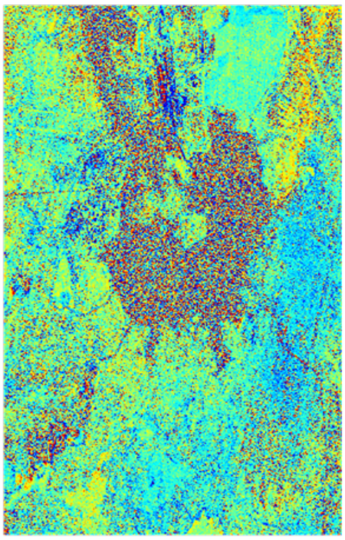

Figure 1: Example of noised phase

The sign $Q_{m,n}$ determines the direction of the phase field increase when bypassing the discontinuity along a closed contour: "+" corresponds to the counterclockwise field increase, "–" denotes ascending when walking clockwise. The discontinuity points occur on the interferogram, as a rule, in pairs, and when the line connecting is intersected in the process of phase unwrapping according to the rule (3), an error occurs, consisting in the appearance or omission of one interferogram fringe, and, so, the absolute phase in such a situation, in principle, cannot be unambiguously restored. Thus, the necessity of remote sensing information obtaining led to development of dozens of phase unwrapping algorithms. The algorithms were developed using the mathematical apparatus of vector field theory (Goldstein Residual Cut method and Green's function method), optimization theory (integer optimization method and minimum cost flow method, filtration theory (Kalman filtration method, nonlinear stochastic filtration method, etc.), solving large systems of linear equations (least squares method), genetic algorithms, neural networks, deep learning techniques, like wrap counting, phase regression, gradient information fusion, etc. [3].

All phase unwrapping alrorythms are applied to a discrete phase interferograms. Processing time and reach several days for big data, which makes it difficult to use the data for the operative solution of monitoring tasks. To unwrap the phase, algorithms have been developed, many of which have high computational complexity and require the use of approximations and simplifications that lead to a decrease in accuracy.

3 Inverse vortex field method for phase unwrapping

The algorithm, which resolves the phase discontinuities, uses the model of a unit vortex $J(z)$ that represents the inversed discontinuity [3]. After summing the interferogram $I(z)$ of size $(M_I \times N_I)$ and the combined inversed unit vortices for all discontinuity points $C(z)$, the unwrapped phase interferogram is obtained. Such an interferogram can be transformed to the absolute phase Ψ without unwrapping errors and artifacts. The process is iterative, because discontinuity points for discrete interferogram may migrate to neighbouring pixels after the inversed vortex application. The proof of the method's convergence is beyond the scope of this work, but all test and real unwrapping cases for remote sensing radar systems shown, that the construction process converges, and the number of migrated discontinuities decreases exponentially.

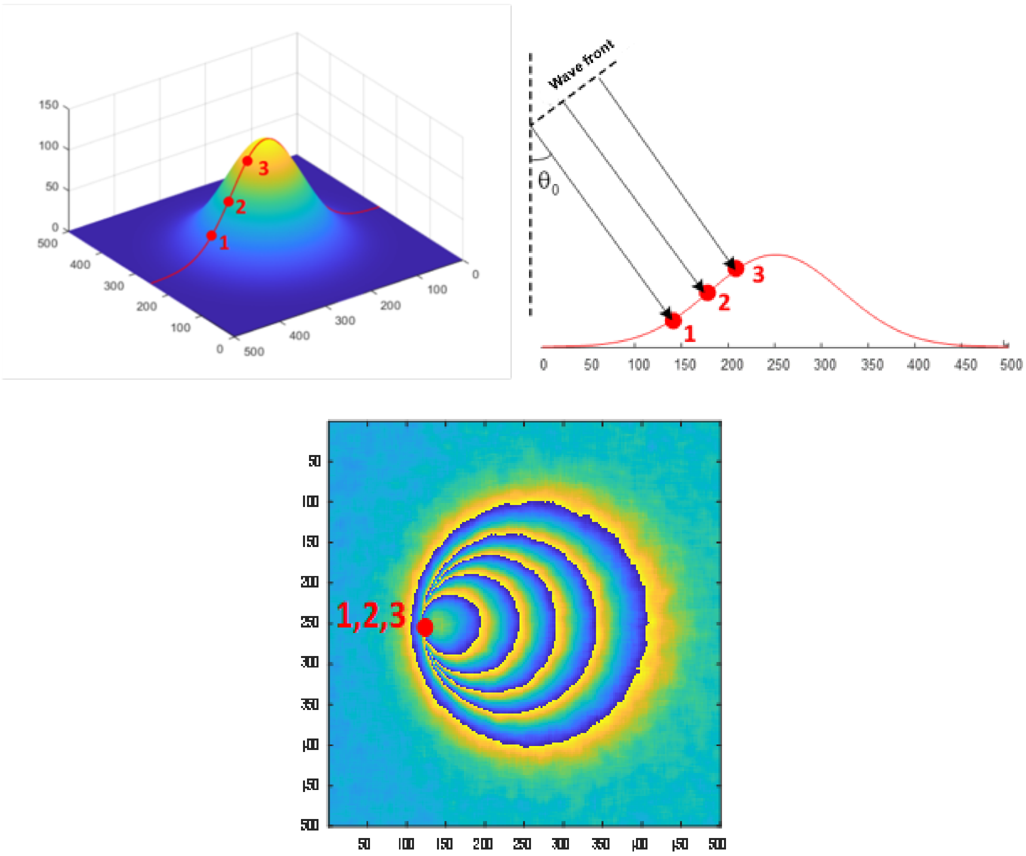

Figure 2: Example of layover effect

Algorithm 1 (serial) consists of the following steps.

0. Initiate the inverse vortex field $C(z)$ of size $(M_I \times N_I)$ with zero values.
1. Pre-compute the unit vortex $J(z) = \arg\{z\}$ for an interferogram of size $2M_I \times 2N_I$.
2. Calculate a discontinuity localization function for the interferogram $I(z) \cdot C(z)$ and calculate the number of discontinuity points. If no new points found, then go to step 6.
3. For each discontinuity point, select a window from the unit vortex field with the size of $M_I \times N_I$ centered at the point.
4. Combine all windows selected at step 3, multiplyed by the discontinuity factors, and add them to the inverse field $C(z)$.
5. Go to step 2.
6. Convert the field to the complex values: $C(z) \to \dot{C}(z) = \exp\{j \cdot C(z)\}$.
7. Perform the unwrapping of the continous phase field and finish.

4 Parallel implementation on CPU

Algorithm 2 (parallel multi-threaded), implemented on C++ language with OpenMP technology for multithreading is the following. Here, differences between it and Algorithm 1 highlighted in bold.

0. Initiate the inverse vortex field $C(z)$ of size $(M_I \times N_I)$ with zero values.
1. Pre-compute the unit vortex $J(z) = \arg\{z\}$ for an interferogram of size $2M_I \times 2N_I$.
2. Calculate a discontinuity localization function for the interferogram $I(z) \cdot C(z)$ and calculate the number of discontinuity points. If no new points found, then go to step 6. **This step is implemented as two nested loops iterating over all elements of the interferogram. The outer loop is distributed between the OpenMP threads.**
3. For each discontinuity point, **precompute the pointer to a window** in the precomputed unit vortex array centered at this point.
4. Combine all windows selected at step 3, multiplyed by the discontinuity factors, and add them to the inverse field $C(z)$. **This is performed as three nested loops. To efficiently utilize multiple threads, vectorization, caches, and take into account heterogeneous cores, we implement the following loop order:**
   (a) **The outer loop iterates over rows of elements, using `#pragma omp for schedule(dynamic,1)` construction. Thus, threads perform the processing one row at time;**
   (b) **Middle loop: iterating over the list of discontinuity points**;
   (c) **Inner loop: iterate over the elements of a row. This loop is vectorized using `#pragma omp simd` construction and also, tiled into blocks that fit into L1 cache. Experiments show that optimal block size is 1024 elements**.
5. Go to step 2.
6. Convert the field to the complex values: $C(z) \to \dot{C}(z) = \exp\{j \cdot C(z)\}$.
7. Perform the unwrapping of the continous phase field and finish.

5 Numerical Experiments

We have performed a series of numerical experiments to access the effect of arithmetic precision and multi-threading on the computing time and accuracy results. To perform these experiments, we have used a 16-core Intel i9-12900K processor.

The test data are represented by three data samples (interferograms): one real scene of TerraSAR-X data (“Grand Canyon” sample, see scene No. 1, Fig. 3-A) and two synthesized scenes with random smoothed Gaussian fields, recalculated into slant projection (see scenes No. 2 and No. 3, Figs. 3-B,C). The scene sizes were the following: 8146×3830 pixels for the data samples No. 1 and 4096×4096 for the data samples No. 2 and No. 3.

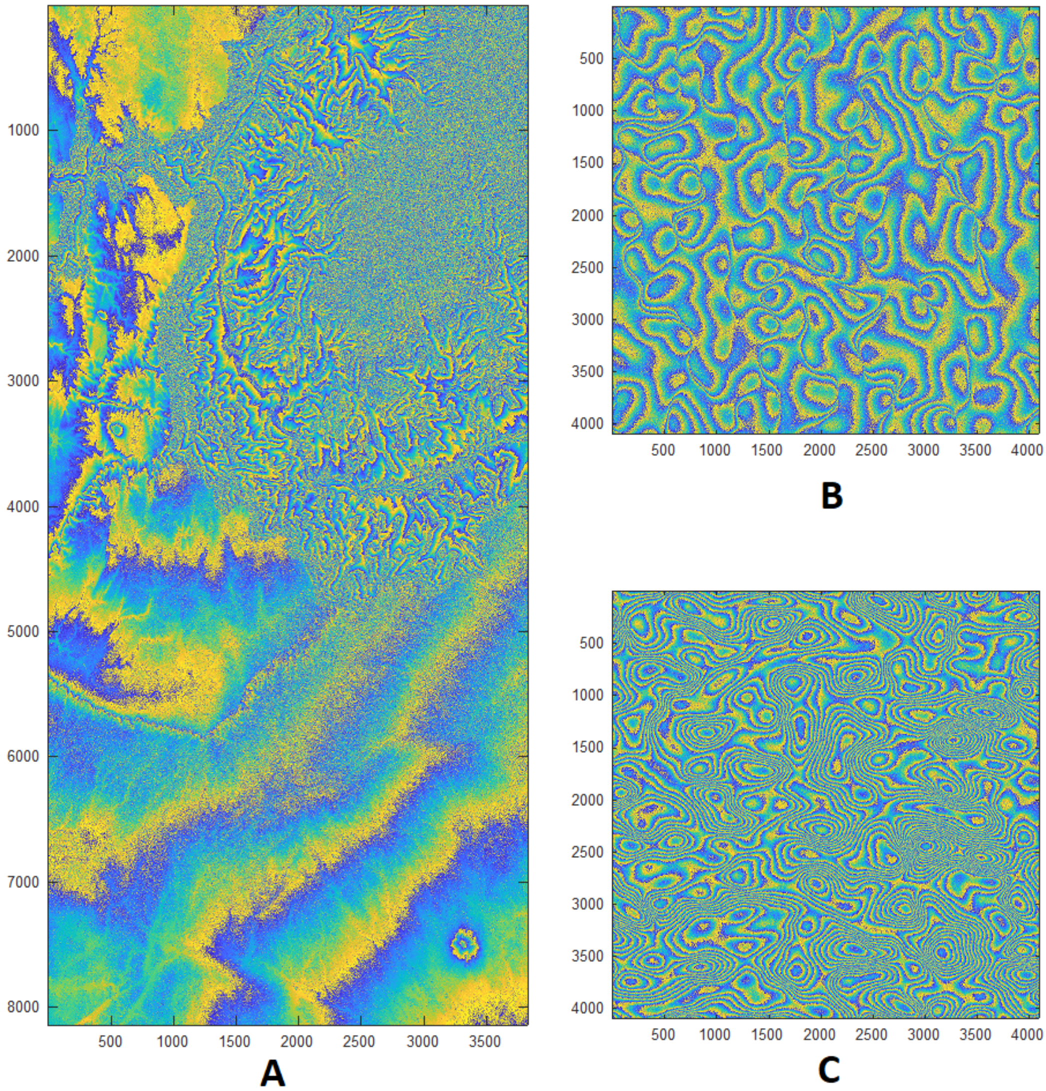

Figure 3: Test scenes: (**A**) scene No. 1 (TerraSAR-X, "Grand Canyon"); (**B**) scene No. 2 (smoothed Gaussian field in the slant projection); (**C**) scene No. 3 (similar to **B**).

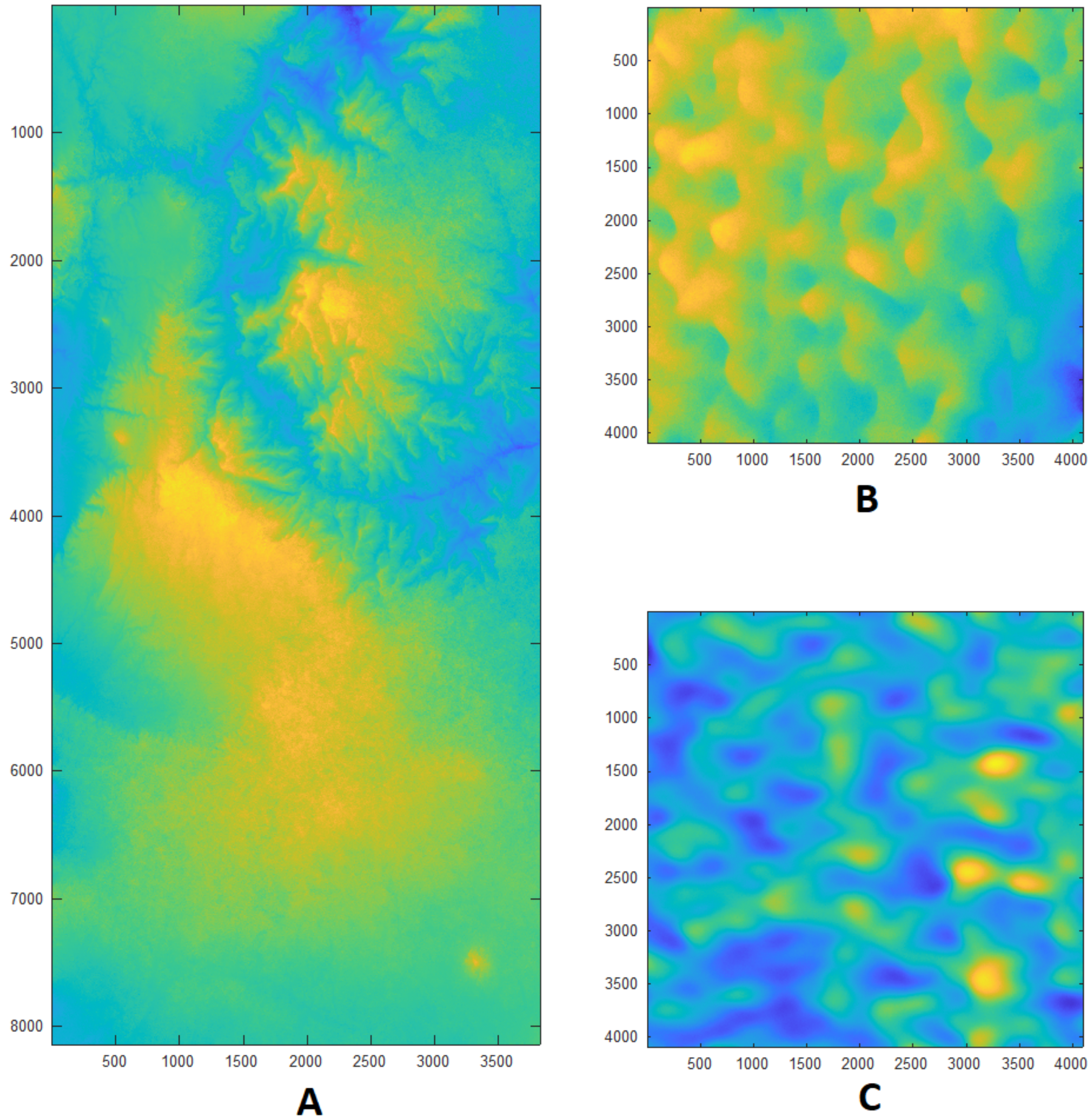

Figure 4: The unwrapping results: (**A**) scene No. 1; (**B**) scene No. 2; (**C**) scene No. 3

Figs. 4-A,B,C show the unwrapping results for the data samples No. 1, No. 2, and No. 3, respectively. The results for scene No. 1 are confirmed by visual estimation; the precision comparison was not performed for this scene.

To estimate the precision for the synthesized scenes No. 2 and No. 3, we used the absolute phase error σ_Ψ and the relative phase error $\widetilde{\sigma}_\Psi$:

$$\sigma_\Psi = \sqrt{\sum_{i=1}^{N} (\Psi_i - \Psi_{0i})^2/(N-1)}, \quad \widetilde{\sigma}_\psi = \sigma_\Psi/\sigma_{\Psi 0}.$$

Ψ_i are the absolute phases, $\sigma_{\Psi 0}$ is the reference absolute phase standard deviation.

The relative error of the synthesized scenes is 22.8% and 23.3%, respectively. These values may be reduced using additional post-processing procedures (residual interferogram flattening, low-pass filtration, scene mirroring [6]).

Table 1 presents the performance results for the developed code for various bit depths and number of threads. As expected, the difference in performance between single precision and double precision is nearly 2 times. For the considered samples, the relative error introduced by the lower precision arithmetic

$$\widetilde{\sigma}_{\text{prec}} = \|\Psi_{\text{FP64}} - \Psi_{\text{FP32}}\| \,/\, \|\Psi_{\text{FP64}}\|$$

is negligible ($\widetilde{\sigma}_{\text{prec}} < 10^{-6}$) in comparison with the total relative error of unwrapped scene. Thus, there is no accuracy degradation for the lower bit depth. Multithreaded code shows a speedup of 9 times in comparison with the single-threaded variant on a 16-core processor.

Table 1: Experimental results for the 3 different test scenes

Scene	Precision/Bit depth	Computing time, single-threaded (minutes)	Computing time, 16 threads (minutes)	Relative error
Real Data 8146×3830	single/FP32	85	10	—
Synthetic Data 1 4096×4096	single/FP32	7.8	0.86	22.8 %
	double/FP64	16.2	1.8	22.8 %
Synthetic Data 2 4096×4096	single/FP32	17.4	1.9	23.3 %
	double/FP64	36	4	23.3 %

6 Conclusion

In this work, we have developed an efficient parallel algorithm for solving the interferometric phase unwrapping problem in radar systems for remote sensing of the Earth. We have realized a parallel code for multicore processors. For our implementation, we have explored the parallel performance and viability of using the lower-precision computations. We have conducted a series of experiments with synthetic data that demonstrated that our code has good performance, 9-fold speedup on a 16-core processor, and usage of single precision instead of double provides a two-fold performance boost without compromising the accuracy of results. We have also performed experiments on large real data to access the parallel performance of our code. It took 10 minutes to unwrap an 8146×3830 interferogram using a 16-core processor in single precision.

Acknowledgment

Author A.V. Sosnovsky was financially supported by the project No. FEUZ-2023-0015 of the Ministry of the Higher Education and Science of Russian Federation. The Authors wish to thank the Reviewers for their careful reading of the manuscript and their fruitful comments and suggestions.

References

[1] Joughin I. R., Li F. K., Madsen S. N., Rodrigues E., Goldstein R. M., et al. Synthetic Aperture Radar Interferometry // IEEE Proc. 2000. Vol. 88. No. 3. P. 33–82.

[2] Hanssen R. F. Radar interferometry. Data interpretation and error analysis. Dordrecht; Boston: Kluwer Academic Publishers. 2002. 308 p.

[3] Martyshko P. S., Akimova E. N., Sosnovsky A. V., Kobernichenko V. G. An Algorithm for Solving the Problem of Phase Unwrapping in Remote Sensing Radars and Its Implementation on Multicore Processors // Mathematics. 2024, Vol. 12. No. 5. 727.

[4] 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. 1998. No. 5. P. 374–379.

[5] Tomioka S., Heshmat S., Miyamoto N., Nishiyama S. Phase unwrapping for noisy phase maps using rotational compensator with virtual singular points // Appl. Opt. 2010. No. 49. P. 4735–4745.

[6] Sosnovsky A. V. Investigation and modification of the inversed vortex phase field method for phase unwrapping. Ural Radio Engineering Journal. 2021;5(3):239–257. DOI: 10.15826/urej.2021.5.3.003.