Cluster Correction for Cluster Analysis-Based Multibaseline InSAR Phase Unwrapping

Zhihui Yuan, *Member, IEEE*, Tianjiao Chen, Hanwen Yu, *Senior Member, IEEE*, Wei Peng, Lifu Chen, *Member, IEEE*, and Xuemin Xing, *Member, IEEE*

***Abstract*—In recent years, multibaseline phase unwrapping (MBPU) has been widely studied, but it is still suffering from the problem of low noise robustness. In view of this, two cluster correction methods for the cluster analysis-based MBPU algorithms are presented to improve the accuracy and effectiveness of the algorithm. In the first method, a suitable box centered on each pixel is selected and the cluster with the highest frequency in the box is taken as the corrected cluster to which the central pixel belongs; in the second method, all pixels are divided into core pixels and noncore pixels according to the predefined density, and then the cluster of the noncore pixels is corrected to the one that appears most frequently in the selected box centered on the noncore pixels. Experiments on both simulated and real datasets confirm the effectiveness of the proposed cluster correction methods.**

***Index Terms*—Cluster analysis (CA), cluster correction, interferometric synthetic aperture radar (InSAR), multibaseline (MB), phase unwrapping (PU).**

I. Introduction

INTERFEROMETRIC synthetic aperture radar (InSAR) can obtain terrain and deformation information of ground targets according to the absolute interferometric phases of SAR images corresponding to the same scene [1], [2]. However, an InSAR system can only get the principal value of the absolute phases, i.e., wrapped phases, which is between $-\pi$ and π or 0 and $2\ \pi$. Therefore, the absolute phases must be reconstructed by phase unwrapping (PU) [3], [4]. To solve this problem, the single-baseline phase unwrapping (SBPU) technique is first proposed. For this traditional technique, there are two unknowns in one equation, which will lead to many solutions. Therefore, to ensure the uniqueness of the solution, SBPU assumes that the phase difference between adjacent pixels is less than π, which is called the Itoh condition or phase continuity assumption [4], [5]. In order to overcome the constraint of the Itoh condition in SBPU, the multibaseline phase unwrapping (MBPU) technique, whose theoretical basis is Chinese remainder theorem (CRT), is proposed [6].

In the past few decades, MBPU has been extensively studied. Three MBPU methods using multiple interferograms with different baselines or frequencies were first proposed in [6]. Then, the maximum likelihood (ML) estimation-based method was proposed in [7], [8]. Subsequently, a new strategy based on maximum posterior probability (MAP) estimation was proposed in [9], [10]. In [11], an extended Kalman filter (EKF) and the ML criterion were combined to improve the performance of the ML-based MBPU method. In [12], the nonlocal means filtering technique was applied to the MAP-based MBPU method. In [13], a two-stage programming approach (TSPA) was proposed, which transplanted the framework of the SBPU to the MBPU. Subsequently, two refined TSPA methods were presented. In [14], the performance of the first stage of TSPA was improved through assuming that the local terrain height surface can be approximated by a local plane to combine more neighborhood information. In [15], the second stage of TSPA was improved by using the unscented Kalman filter. In [16], the L^p-norm model was used to compare the digital elevation model (DEM) reconstruction accuracy between the classical SBPU methods and their corresponding TSPA-based MBPU methods, and it was found that the latter is better. In [17], [18], the baseline design criteria and requirements for MBPU were outlined. In [19], the artificial intelligence (AI)-based SBPU methods and MBPU methods in InSAR were reviewed comprehensively.

In addition, more and more researchers focus on the cluster analysis (CA)-based MBPU methods. In [20], the fast CA-based MBPU technique was first proposed. In this technique, the pixels were clustered according to the intercept information of the MB interferograms, and then MBPU is performed cluster by cluster. In [21], the closed-form robust CRT was used to solve the ambiguity vectors of each pixel, then the clusters were conducted according to the ambiguity vectors, and some measures were designed to improve the performance. In [22], a CA-based noise-robust MBPU algorithm (abbreviated as CANOPUS) was proposed, in which the recognizable mathematical pattern was extended from the intercept dimension to the azimuth, range, and intercept dimensions. In [23], a linear combination strategy

Manuscript received 29 August 2022; revised 22 September 2022; accepted 23 September 2022. Date of publication 6 October 2022; date of current version 12 October 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 42271456, Grant 42074033, Grant 61701047, and Grant 41701536, in part by the Hunan Provincial Natural Science Foundation of China under Grant 2019JJ50639 and Grant 2017JJ3322, and in part by the Scientific Research Fund of Hunan Provincial Education Department under Grant 20B038 and Grant 18A148. *(Corresponding author: Hanwen Yu).*

Zhihui Yuan, Tianjiao Chen, and Lifu Chen are with the School of Electrical and Information Engineering, Changsha University of Science and Technology, Changsha 410114, China (e-mail: yuanzhihui@csust.edu.cn; douxie@stu.csust.edu.cn; lifu_chen@csust.edu.cn).

Hanwen Yu is with the School of Resources and Environment, University of Electronic Science and Technology of China, Chengdu 611731, China (e-mail: yuhanwenxd@gmail.com).

Wei Peng and Xuemin Xing are with the School of Traffic and Transportation Engineering, Changsha University of Science and Technology, Changsha 410114, China (e-mail: pengwei@csust.edu.cn; xuemin.xing@csust.edu.cn).

Digital Object Identifier 10.1109/JSTARS.2022.3211535

was adopted to increase the ambiguity heights and reduce the influence of noise for the CA-based MBPU method. In [24], a closed-form robust CA-based MBPU and filtering algorithm was proposed, which further improves the efficiency and accuracy of the PU results. In [25], the block-matching and 3-D filtering (BM3D) technique was combined with the CA-based MBPU method to further improve the PU accuracy. In [26], a deep convolutional neural network (CANet)-based MBPU method was proposed.

However, the noise robustness of the CA-based MBPU methods still needs to be further improved: 1) It is sensitive to phase noise, and a small phase noise may cause a large unwrapping error, which makes the noise robustness of the method poor [17]; 2) two clusters that are close to each other may merge into one cluster for the CA-based MBPU algorithm, resulting in wrong clustering results [22]; and 3) for the CANOPUS algorithm, many noise patterns and some small-scale clusters will appear in the case of serious noise [22]. Therefore, it is necessary to take some measures to correct those wrong clusters.

Inspired by the concepts of density and core patterns introduced in [22], we propose two cluster correction methods for the CA-based MBPU algorithms to improve the noise robustness. In the first method, the cluster that appears the most times in the selected box is taken as the corrected cluster of the center pixel. In the second method, all the pixels are first divided into core pixels and noncore- pixels according to the density which is predefined, and then the cluster of the noncore pixels is corrected to the one that appears the most times in the selected box.

The main contributions of this paper are threefold.

a) Two cluster correction methods are proposed, one is the pixel-by-pixel cluster correction (PPCC) method, and the other is the noncore pixel cluster correction (NPCC) method. Both methods can effectively reduce the interference of noise and improve the clustering accuracy and effectiveness of the CA-based MBPU algorithms.
b) Two new definitions of density are proposed for the NPCC method. Compared with the density definition in the CANOPUS algorithm, the boundary between adjacent clusters is more obvious under these two definitions.
c) The selection of box size and density threshold for the cluster correction methods is discussed. By selecting the box size and determining the density threshold appropriately, cluster correction can be carried out smoothly in different terrain environments while ensuring the execution efficiency.

The structure of the article is as follows. In Section II, the basic principle and limitations of the CA-based MBPU are reviewed. Then, Section III introduces the main principles of the cluster correction methods. Section IV verifies the effectiveness of the proposed cluster correction methods by experiments. Finally, Section V concludes this article.

II. Basic Principle and Limitations of the CA-Based MBPU Methods

The measured wrapped phase of the target can be determined by [20].

$$\varphi(s) = \psi(s) - 2k(s)\pi \tag{1}$$

where $\psi(s)$ is the flattened absolute interferometric phase of the *s*th pixel, $k(s)$ is the unknown ambiguity number, and $\varphi(s)$ is the wrapped phase. To obtain the correct unwrapped phase $\psi(s)$, we should find the correct ambiguity number $k(s)$ first. Since there are two unknowns in an equation, it is impossible to directly solve the two unknowns [$\psi(s)$ and $k(s)$] from (1). There are two methods to solve (1): The first one is SBPU, but it must obey the phase continuity assumption. The second one is MBPU. For brevity, only two baselines are considered in the following discussion. Assuming that the wavelength of the InSAR system is λ, and the look angle of the master channel is θ, then the terrain height can be reconstructed by [20]

$$h(s) = \frac{\lambda \cdot r(s) \cdot \sin\theta}{4\pi \cdot B_i} \cdot \psi_{\mathrm{i}}(s) \tag{2}$$

where $r(s)$ is the slant range of the *s*th pixel for the master channel, $\psi_i(s)$ denotes the flattened absolute interferometric phase of the *s*th pixel in the *i*th interferogram, $h(s)$ is the terrain height of the *s*th pixel, and B_i represents the perpendicular baseline of the *i*th interferogram. In combination with (1) and (2), the following formula can be obtained:

$$k_2 = \frac{B_2}{B_1} \cdot k_1(s) + \frac{\mathrm{B}_2 \cdot \varphi_1(s) - \mathrm{B}_1 \cdot \varphi_2(s)}{2\pi \cdot B_1}. \tag{3}$$

It can be seen from (3) that when the baselines have been determined, the slope of the line is a constant, and the intercept is jointly determined by B_1, B_2, $\varphi_1(s)$, and $\varphi_2(s)$. Defines the ambiguity vector $[k_1(s), k_2(s)]$ as the ambiguity numbers corresponding to the *s*th pixel of the two channels. Therefore, the pixels within the same cluster must have the same ambiguity vector, and the corresponding lines have the same slope and pass through the same integer point$[k_1(s), k_2(s)]$ on the k_1–k_2 lane. It follows that the lines within the same cluster overlap each other, i.e., their intercepts are the same. This is the basic principle of the CA-based MBPU methods, i.e., all the pixels corresponding to the lines with the same intercept belong to the same cluster. In addition, for the CANOPUS algorithm, it is required not only that the pixels in the same cluster have similar intercepts, but also that they are density-connected in the 3-D space composed of row (azimuth), line (range), and intercept dimensions.

To summarize, the main processing procedure of the CA-based MBPU methods can be shown in Fig. 1, and the main steps are as follows.

Step 1: The intercept map is obtained according to the inputted interferograms by using (3).

Step 2: All the pixels are clustered into different groups according to the CA algorithm or the CANOPUS algorithm.

Step 3: The intercepts of each cluster's centerline are found by using the histogram method.

Step 4: The ambiguity vectors of each group are obtained according to the intercept of each cluster's centerline.

Step 5: The unwrapped phases are calculated according to the ambiguity vectors of each group.

It should be noted that the clustering result in Fig. 1 is shown with different numbers to distinguish different clusters for the convenience of description, which is also adopted in [20], [22]. Therefore, we use the term "cluster number" to describe the

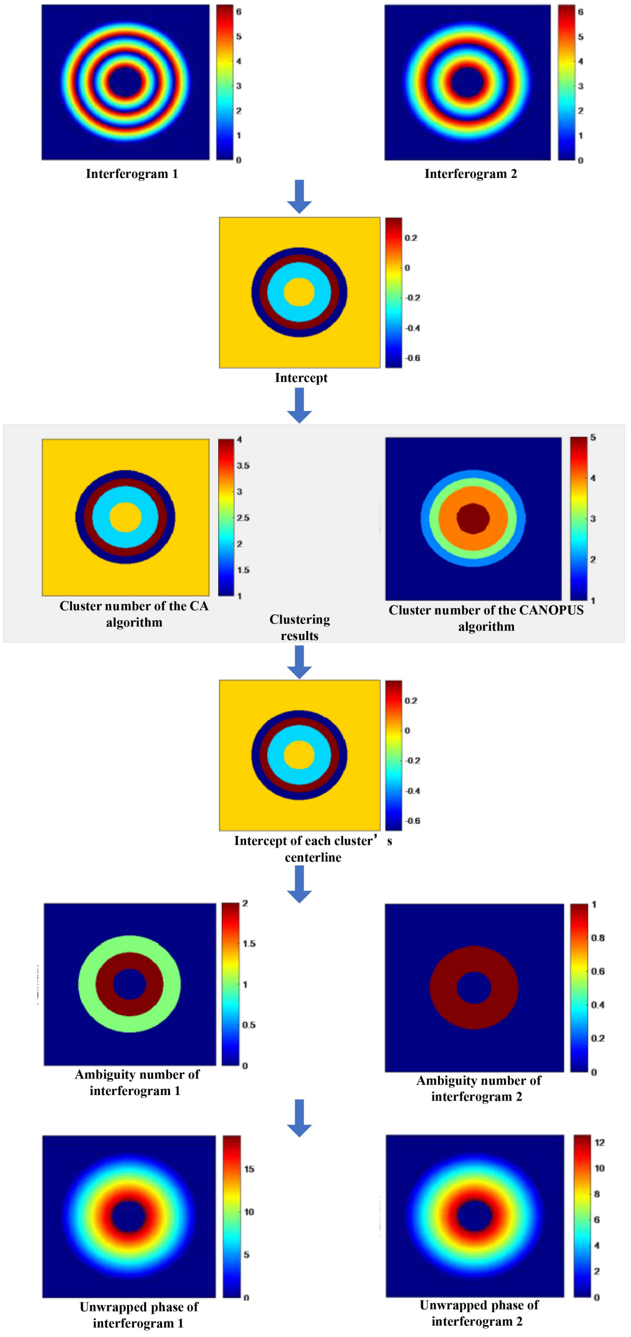

Interferogram 1
Interferogram 2
Intercept
Cluster number of the CA algorithm
Clustering results
Cluster number of the CANOPUS algorithm
Intercept of each cluster' s centerline
Ambiguity number of interferogram 1
Ambiguity number of interferogram 2
Unwrapped phase of interferogram 1
Unwrapped phase of interferogram 2

Fig. 1. Main processing procedure of the CA-based MBPU.

cluster to which a pixel belongs, in this article. But there are some subtle differences in the meaning of the term in these two clustering algorithms. The CA algorithm only distinguishes different clusters by the intercept information, so the cluster number is only related to the intercept value, and it is specified that each cluster is numbered in the order of the intercept value from small to large. For example, if there are L clusters in total, then they can be numbered as $1, 2, \ldots, L$, according to the intercept value. However, the CANOPUS algorithm distinguishes different clusters by row, line, and intercept information, and there are cases where the intercepts are similar but the row and line positions are quite different, so each cluster is numbered according to the order in which it is identified. Moreover, there are noise patterns in the CANOPUS algorithm that do not belong to any cluster and are numbered as -1. Therefore, the greater the cluster number difference in the CA algorithm, the greater the difference in their intercept values, and the greater the cluster number difference in the CANOPUS algorithm, the greater the difference in their 3-D spatial locations.

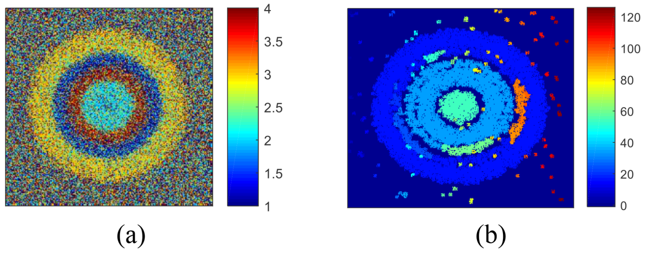


Fig. 2. Clustering results in the presence of noise. (a) The clustering result of the CA algorithm with noise (unit: rad). (b) The clustering result of the CANOPUS algorithm with noise (unit: rad).

However, the CA-based MBPU methods still have some shortcomings. For example, Fig. 2(a) is the clustering result of the CA algorithm in the presence of noise, and Fig. 2(b) is the clustering result of the CANOPUS algorithm under the same condition. It is clear from Fig. 2 that there will be some wrong clustering results when noise is present in the interferograms, which need to be corrected. Therefore, two cluster correction methods are proposed in the following section to improve the accuracy of the clustering results.

III. Cluster Correction for the Clustering Results

According to the analysis in the last paragraph of the previous section, there will be some wrong clustering results when only the CA or CANOPUS algorithm is used. To further improve the PU accuracy, cluster correction is needed after the initial clustering results are obtained. In fact, we only add a cluster correction step to the original CA or CANOPUS algorithm, but the other steps do not change. In other words, when the corrected clustering results are obtained, we still use the histogram method to find the intercepts of each cluster's centerline, and then use the search method introduced in [20] or the closed-form solving formula given by [24] to find out the ambiguity vectors corresponding to those intercepts, and finally get the unwrapped phase according to the ambiguity vectors.

A. *Pixel-by-Pixel Cluster Correction (PPCC) Method*

For the pixel-by-pixel cluster correction (PPCC) method, the cluster numbers of all pixels are considered unreliable and need to be corrected. A suitable box is selected for each pixel, and the cluster number with the highest frequency in the box is taken as the corrected cluster number of the pixel in the center of the box.

The specific steps are as follows: First is selecting a pixel and designing an appropriate box centered on the pixel, then counting the occurrence times of each cluster number in each box, looking for the cluster number with the most occurrence times in the current box, and then modifying the cluster number of current center pixel to the one with the most occurrence times in the box. This process is repeated until all the pixels are corrected. The pseudocode of the PPCC method is given below.

```
Step 1: WHILE $M \neq \emptyset$ ($M$ is defined as the set of pixels to be corrected).
Step 2: Select a pixel $i$ in $M$ and design an appropriate box centered on this pixel.
Step 3: Count the number of occurrences of each cluster number in the box.
Step 4: Correct the cluster number of pixel i to the cluster number that appears most frequently in the box, and update pixel i to the next pixel.
Step 5: END WHILE.
```

B. Noncore Pixel Cluster Correction (NPCC) Method

For the NPCC method, only the cluster numbers of the noncore pixels are considered unreliable and need to be corrected. The concepts of density, core pattern, and noncore pattern were introduced into the CANOPUS algorithm by [22]. For better understanding, "pattern" is replaced by "pixel" in this article. Intuitively, density is a parameter that describes whether a pixel is a core pixel or a noncore pixel. When the density is greater than the certain threshold (i.e., the *MinPts* in [22]), the pixel is regarded as a core pixel and its corresponding cluster number is considered to be reliable; otherwise, it is a noncore pixel, and its corresponding cluster number must be corrected. In view of the definition of density in [22] has no clear physical meaning, two new definitions are proposed in this article. The first definition is the number of pixels in the selected box with the same cluster number of the center pixel. The second one is the number of pixels in the selected box whose intercept difference with the center pixel is less than a fixed value. Assuming that the simplest integer ratio of B_2/B_1 is Γ_1/Γ_2, then the minimum intercept difference between any two clusters is $1/\Gamma_2$ according to Fig. 1(b) of [24]. Thus, this fixed value can be taken as $1/(2\Gamma_2)$. For example, if the first definition is used and the box size is $W \times W$, and there are n pixels with the same cluster number as the center pixel, then the density is n. After the density is defined, the pixel whose density is greater than the threshold value will be regarded as a core pixel; otherwise, it will be regarded as a noncore pixel and should be corrected. For convenience, the cluster correction method using the first density definition is called the NPCC1 method, and the other one is named as the NPCC2 method.

The pseudocode of the NPCC method is given below.

```
Step 1: WHILE $M \neq \emptyset$ ($M$ is defined as the set of pixels to be corrected).
Step 2: Select a pixel $i$ in $M$ and select an appropriate box centered on this pixel.
Step 3: Select a density definition to calculate the density n corresponding to pixel $i$.
Step 4: The density threshold $n_T$ is set according to the density distribution in the box.
Step 5: If the density of pixel $i$ is greater than $n_T$, it is denoted as a core pixel; otherwise, it is denoted as a noncore pixel.
Step 6: According to the ML criterion, the cluster number of the noncore pixel is changed to the cluster number that appears the most times in the box. And pixel i is updated to the next pixel.
Step 7: END WHILE.
```

C. Box Size and Minimum Points Determination Method

Before performing the two cluster correction methods, some parameters need to be determined first since different parameter values may lead to different results. For PPCC, we just need to determine the box size ($W_1 \times W_1$). But for NPCC, the box size ($W_2 \times W_2$) and the corresponding certain threshold *MinPts* (minimum number of points) are both need to be determined. A large box size can improve the execution efficiency, but it cannot distinguish some small clusters. A small box size can distinguish small clusters, but it will take more time.

Generally, the selection of box size for both PPCC and NPCC is related to the resolution of the SAR image, the complexity of terrain, the severity of phase noise, and so on. If the SAR image resolution of the interesting area is high or the surface changes drastically or the noise is not significant, a small box can be selected; otherwise, a large box should be selected. Of course, it is necessary to determine the optimal size through multiple experiments with several values.

In addition, the box size can be the same or different for each investigated pixel according to the actual situation. If the surface changes slowly or the noise is not significant, or the data set is small, a constant box size and minimum points can be employed. This is a simplified method, which can save the execution time of the algorithm. For the experiments in Section IV, all the data sets are small, so the boxes with the same size and MinPts are selected for the whole interferogram.

In fact, the selection of the box size is similar to the determination of the parameter radius ε in Section III-B of [22], so the radius determination method introduced in that paper can also be used exactly, i.e., regard the radius which leads to the first peak of the density derivative curve with respect to the radius as the final radius ε, and then the box size will be set to $[2\varepsilon] \times [2\varepsilon]$, where $[\cdot]$ represents the rounding operation. It should be noted, however, that the box size used for cluster correction does not need to be the same as the one used for density calculation.

The determination of the density certain threshold *MinPts* is more complicated. It depends not only on the noise level and terrain variation, but also on the size of the box. If a smaller density threshold *MinPts* is selected, the number of noise pattern can be reduced, but merging may occur between adjacent clusters. If a larger density threshold *MinPts* is selected, the higher clustering accuracy can be obtained, but the number of noise patterns will also increase. When the noise is serious and the terrain change is steep, there will be fewer pixels with the same cluster number, so a smaller *MinPts* should be chosen. Conversely, *MinPts* can be larger. According to the practical experiences in Section IV,

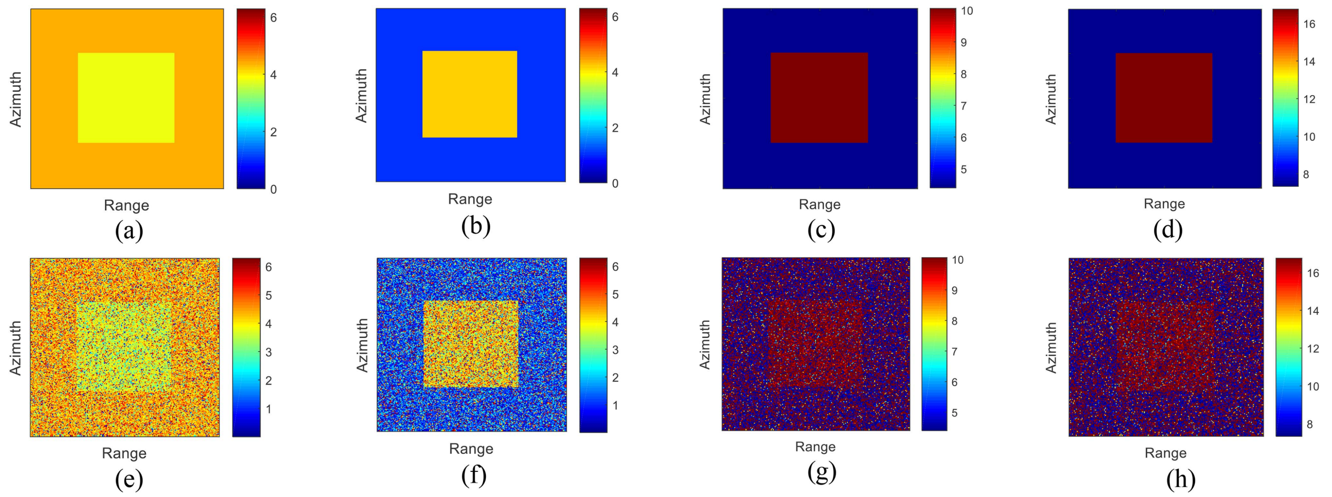


Fig. 3. Simulated MB InSAR data set for Experiment 1. (a) Noiseless interferogram with the short baseline (unit: rad). (b) Noiseless interferogram with the long baseline (unit: rad). (c) Noiseless unwrapped phase with the short baseline (unit: rad). (d) Noiseless unwrapped phase with the long baseline (unit: rad). (e) Noisy interferogram with the short baseline (unit: rad). (f) Noisy interferogram with the long baseline (unit: rad). (g) Noisy unwrapped phase with the shot baseline (unit: rad). (h) Noisy unwrapped phase with the long baseline (unit: rad).

MinPts should be greater than or equal to half of the total number of pixels in the box.

D. Time Complexity

Since this article only adds a cluster correction process to the original CA or CANOPUS algorithm, it only needs to analyze the time complexity of this process. The time complexity of the other processing steps is consistent with [20] and [22]. Because the PPCC method needs to correct all the pixels, its time complexity is obviously higher than the NPCC method, especially when there are fewer noncore pixels.

Suppose that the size of the interferogram is N, and the size of the selected box is W_1^2. Among all the steps of the PPCC method, it can be seen from the description of Section III-A that the time complexity of Step 3 and Step 4 are both $\mathrm{O}(W_1^2 N)$, and the time complexity of the other steps is negligible. Therefore, the time complexity of the PPCC method is $\mathrm{O}(2W_1^2 N)$.

For the NPCC method, suppose that the size of the selected box is W_2^2, and the total number of noncore pixels is C. Then, from the description of Section III-B that the time complexity of Step 3 is $\mathrm{O}(W_2^2 N)$, Step 6 is $\mathrm{O}(W_2^2 C)$, and the computation of the other steps is negligible. In general, the number of noncore pixels C is much smaller than the size N of the interferogram, so the time complexity of the NPCC method is $\mathrm{O}(W_2^2 N)$.

IV. Experimental Results and Analysis

In this section, the cluster correction methods proposed in Section III will be used on the clustering results of the two clustering algorithms to verify the effectiveness of the cluster correction methods.

The first experiment was performed on a simulated scenario with both discontinuous and smooth regions. The real heights of the scene are set to 35 and 80 m, respectively. The baselines are 500 and 300 m, and the coherence coefficients of the corresponding interferogram are 0.8 and 0.7, respectively. The true intercepts corresponding to this simulated scenario (i.e., the coresponding intercepts in the case of no noise) are 1 and 1/3, respectively, so there should be only two clusters when clustering with the CA algorithms in the case of no noise. The two simulated noiseless interferograms are shown in Fig. 3(a) and (b). The true noiseless unwrapped phases are shown in Fig. 3(c) and (d). The simulated noisy interferograms are shown in Fig. 3(e) and (f), respectively, and the noisy unwrapped phases are shown in Fig. 3(g) and (h). Comparing Fig. 3(a) and (b) with Fig. 3(e) and (f), we find that the interferogram has been seriously contaminated by noise. Comparing Fig. 3(c) and (d) with Fig. 3(g) and (h), we find that due to the presence of noise contamination, there are many serious errors in the unwrapped results obtained by direct using the CA or CANOPUS algorithm without cluster correction.

If the box size is set to be 9 × 9, we will get the density distribution maps with three different definitions shown in Fig. 4. The density distribution map defined by the CANOPUS algorithm is shown in Fig. 4(a). It can be seen from Fig. 4(a) that it does not show the boundaries between adjacent clusters well. Fig. 4(b) shows the density distribution map defined by the NPCC1 method, and Fig. 4(c) is the density distribution map defined by the NPCC2 method. It can be seen from Fig. 4(b) and (c) that the two density definitions show the boundaries between adjacent clusters well.

Then, the cluster correction methods are used to correct the clustering results of the CA algorithm and CANOPUS algorithm for verifying the effectiveness of the proposed methods. The cluster number distribution map generated by the CA algorithm is shown in Fig. 5(a), which shows that there are many wrong clusters. The cluster number distribution map after the application of PPCC is shown in Fig. 5(b). After NPCC1 is applied, the corresponding cluster number distribution map is shown in Fig. 5(c). After NPCC2 is applied, the corresponding cluster number distribution map is shown in Fig. 5(d). As can be seen

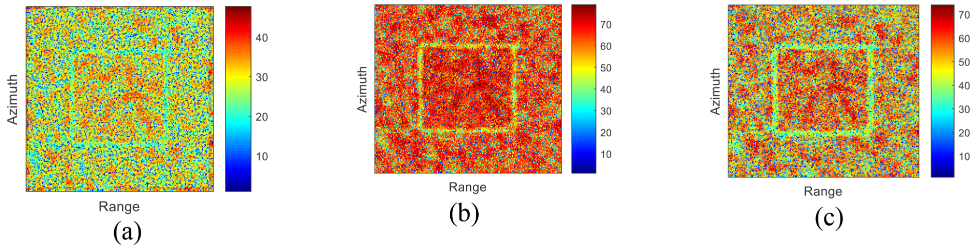


Fig. 4. Density distribution maps with three different definitions for Experiment 1. (a) Density distribution map defined by the CANOPUS algorithm. (b) Density distribution map defined by the NPCC1 method. (c) Density distribution map defined by the NPCC2 method.

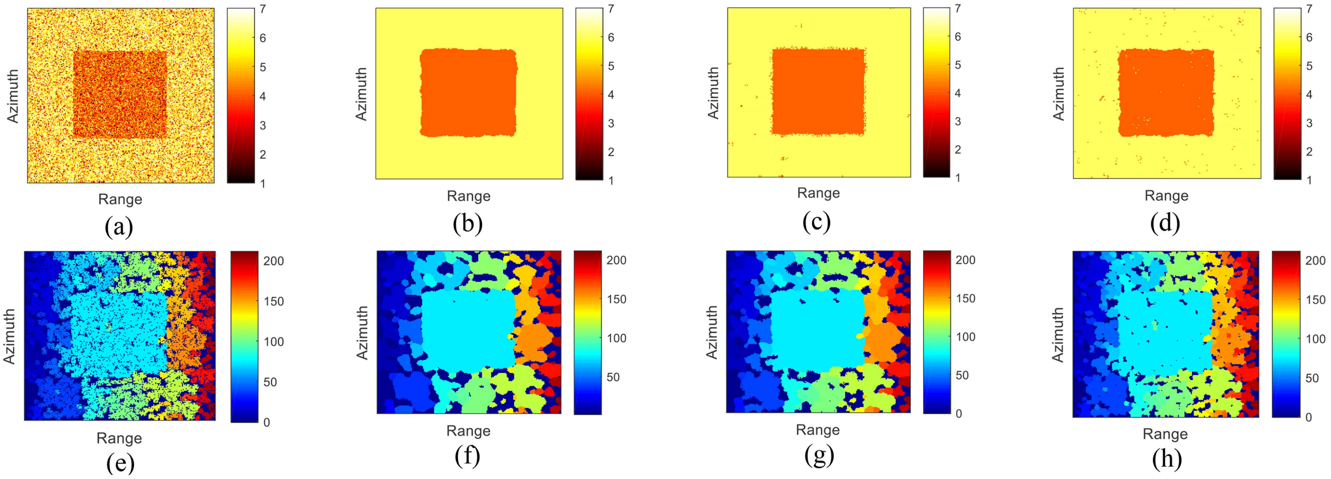


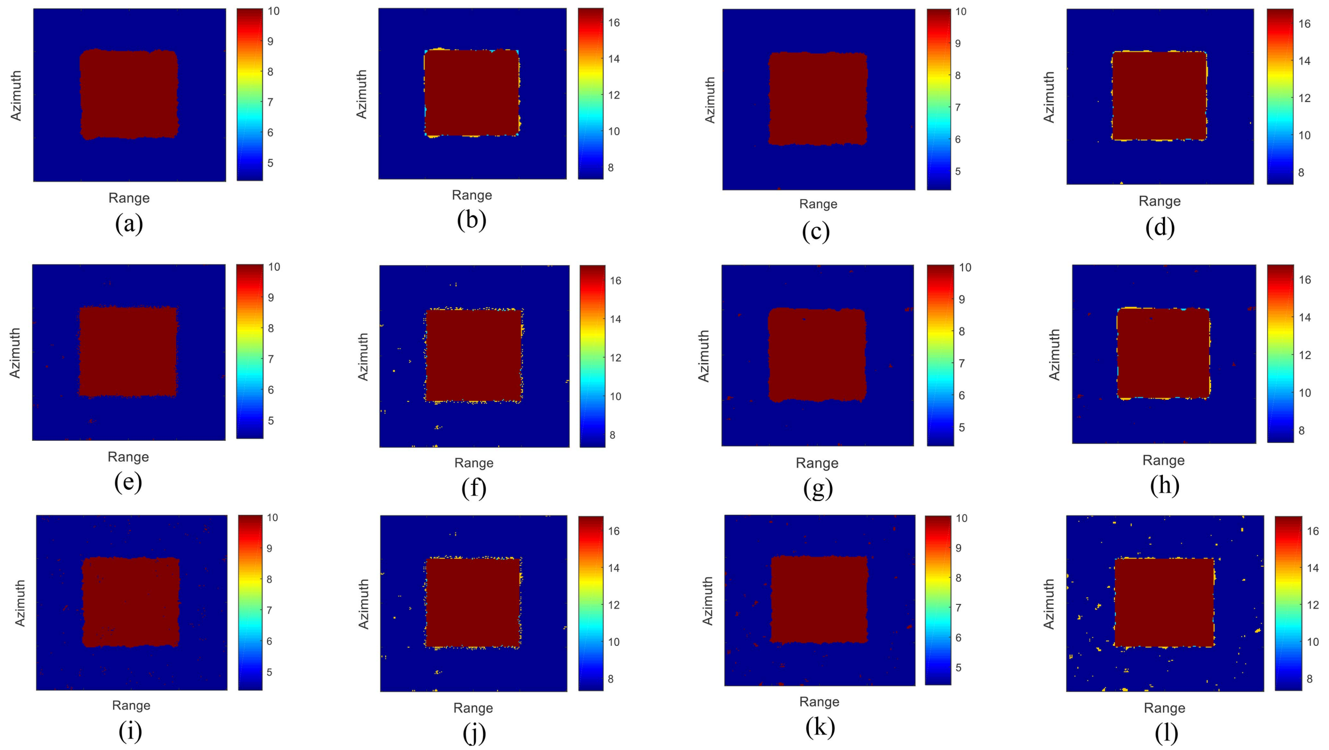
Fig. 5. Comparison of the clustering results before and after cluster correction for Experiment 1. (a) Cluster number distribution map corresponding to the CA algorithm. (b) Cluster number distribution map corresponding to the CA algorithm after cluster correction by the PPCC method. (c) Cluster number distribution map corresponding to the CA algorithm after cluster correction by the NPCC1 method. (d) Cluster number distribution map corresponding to the CA algorithm after cluster correction by the NPCC2 method. (e) Cluster number distribution map corresponding to the CANOPUS algorithm. (f) Cluster number distribution map corresponding to the CANOPUS algorithm after cluster correction by the PPCC method. (g) Cluster number distribution map corresponding to the CANOPUS algorithm after cluster correction by the NPCC1 method. (h) Cluster number distribution map corresponding to the CANOPUS algorithm after cluster correction by the NPCC2 method.

from the corrected results, most of the incorrect cluster numbers have been corrected.

Since different clusters in the CANOPUS algorithm may have the same ambiguity vector, the number of clusters in the clustering results of the algorithm is significantly more than that of the CA algorithm. The cluster number distribution map generated by the CANOPUS algorithm is shown in Fig. 5(e). There are many small clusters in the map, some of which even only have a few pixels, and these clusters are obviously meaningless. The cluster number distribution map corresponding to the application of PPCC is shown in Fig. 5(f). After NPCC1 is applied, the corresponding cluster number distribution map is shown in Fig. 5(g). After NPCC2 is applied, the corresponding cluster number distribution map is shown in Fig. 5(h). At this time, the number of small clusters decreased significantly. The cluster correction is like that of filtering, but it does not affect the resolution. Even if some clustering results are wrong, they only make the boundaries shift a bit, not make them disappear. For the CA algorithm, the execution time of the PPCC method is 0.726 s, the execution time of the NPCC1 method is 0.511 s, and the time of the NPCC2 method is 0.521 s. For the CANOPUS algorithm, the execution time of the PPCC method is 1.831 s, the execution time of the NPCC1 method is 0.964 s, and the time of the NPCC2 method is 0.972 s [CPU: Intel (R) Core (TM) i7-4510U 2.00 GHz, RAM: 8.00GB].

For the CA algorithm, the unwrapped phases with the short and long baseline after cluster correction by the PPCC method are shown in Fig. 6(a) and (b). After NPCC1 is applied, the corresponding unwrapped phases are shown in Fig. 6(c) and (d). After NPCC2 is applied, the corresponding unwrapped phases are shown in Fig. 6(e) and (f). For the CANOPUS algorithm, the unwrapped phases with the short and long baselines after cluster correction by the PPCC method are shown in Fig. 6(g) and (h). After NPCC1 is applied, the corresponding unwrapped phases are shown in Fig. 6(i) and (j). After NPCC2 is applied, the corresponding unwrapped phases are shown in Fig. 6(k) and (l).

From the preliminary results of this experiment, the two cluster correction methods proposed in this article can correct the wrong cluster numbers to improve the PU accuracy. The following is a quantitative analysis of the performance of the two cluster correction methods. For the CANOPUS algorithm, the correctness of a pixel's cluster cannot be judged directly according to the cluster number, so the correctness of the unwrapped results will be judged by the correctness of the corresponding



(a) (b) (c) (d) (e) (f) (g) (h) (i) (j) (k) (l)

Fig. 6. Comparison of the unwrapped results after cluster correction for Experiment 1. (a) Unwrapped phase with the short baseline for the CA algorithm after cluster correction by the PPCC method (unit: rad). (b) Unwrapped phase with the long baseline for the CA algorithm after cluster correction by the PPCC method (unit: rad). (c) Unwrapped phase with the short baseline for the CA algorithm after cluster correction by the NPCC1 method (unit: rad). (d) Unwrapped phase with the long baseline for the CA algorithm after cluster correction by the NPCC1 method (unit: rad). (e) Unwrapped phase with the short baseline for the CA algorithm after cluster correction by the NPCC2 method (unit: rad). (f) Unwrapped phase with the long baseline for the CA algorithm after cluster correction by the NPCC2 method (unit: rad). (g) Unwrapped phase with the short baseline for the CANOPUS algorithm after cluster correction by the PPCC method (unit: rad). (h) Unwrapped phase with the long baseline for the CANOPUS algorithm after cluster correction by the PPCC method (unit: rad). (i) Unwrapped phase with the short baseline for the CANOPUS algorithm after cluster correction by the NPCC1 method (unit: rad). (j) Unwrapped phase with the long baseline for the CANOPUS algorithm after cluster correction by the NPCC1 method (unit: rad). (k) Unwrapped phase with the short baseline for the CANOPUS algorithm after cluster correction by the NPCC2 method (unit: rad). (l) Unwrapped phase with the long baseline for the CANOPUS algorithm after cluster correction by the NPCC2 method (unit: rad).

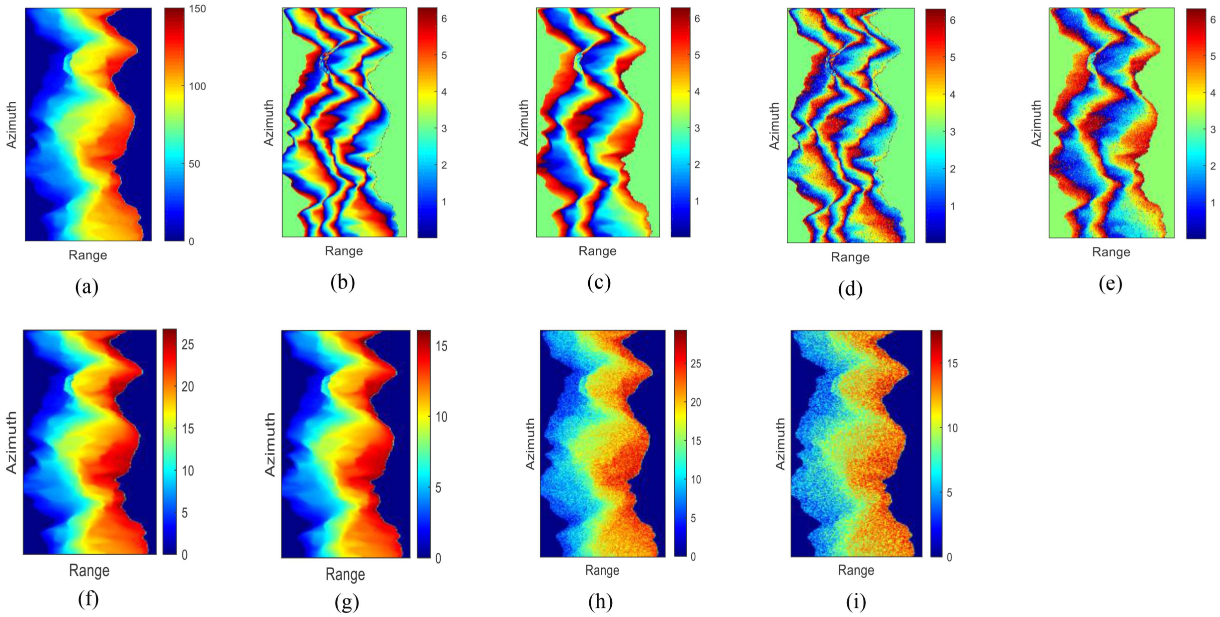
TABLE I
COMPARION OF PUSR FOR EXPERIMENT 1

	PUSR of the CA algorithm		PUSR of the CANOPUS algorithm	
	Interferogram 1	Interferogram 2	Interferogram 1	Interferogram 2
Before correction	76.19%	78.55%	79.84%	79.26%
PPCC	99.09%	98.78%	93.11%	92.77%
NPCC1	95.12%	96.31%	90.12%	89.22%
NPCC2	95.99%	96.44%	88.44%	90.11%

ambiguity vector of each pixel. The PU success rate (PUSR) is defined as the percentage of pixels whose ambiguity number is correctly restored to the total pixels in an interferogram. Table I shows the PUSR with the CA algorithm and the CANOPUS algorithm before and after cluster correction in Experiment 1. According to the results, the PUSR has been greatly improved after cluster correction.

In the second experiment, the cluster correction methods are tested on a more realistic height profile. Fig. 7(a) is the reference DEM corresponding to a mountainous area of Isolation Peak in Colorado, USA. And the height ambiguities are 32.1 and 53.5 m, respectively, and the coherence coefficients of the corresponding interferograms are both 0.7. Fig. 7(b) and (c) shows the noiseless interferograms with two different baselines, and Fig. 7(d) and (e) is the corresponding noisy interferograms. Fig. 7(f) and (g) shows the true unwrapped phase with two different baselines, and Fig. 7(h) and (i) is the noisy unwrapped phase obtained by the CA algorithm.

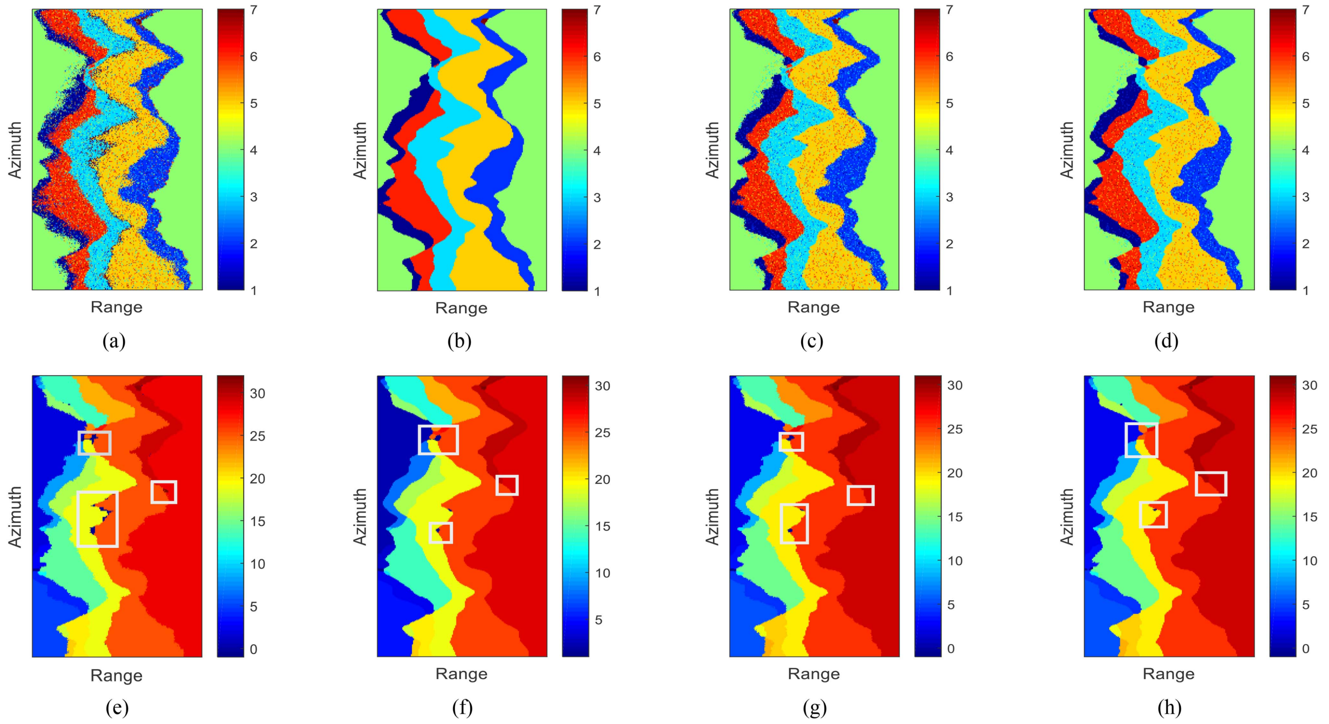
Fig. 8(a) is the cluster number distribution map of the CA algorithm. Fig. 8(b) is the cluster number distribution map after



(a) (b) (c) (d) (e)

(f) (g) (h) (i)

Fig. 7. Simulated MB InSAR data set for Experiment 2. (a) Reference DEM about Isolation Peak, Colorado, USA (unit: m). (b) Noiseless interferogram with the long baseline (unit: rad). (c) Noiseless interferogram with the short baseline (unit: rad). (d) Noisy interferogram with the long baseline (unit: rad). (e) Noisy interferogram with the short baseline (unit: rad). (f) Noiseless unwrapped phase with the long baseline (unit: rad). (g) Noiseless unwrapped phase with the short baseline (unit: rad). (h) Noisy unwrapped phase with the long baseline (unit: rad). (i) Noisy unwrapped phase with the short baseline (unit: rad).



(a) (b) (c) (d)

(e) (f) (g) (h)

Fig. 8. Comparison of the clustering results before and after cluster correction for Experiment 2. (a) Cluster number distribution map of the CA algorithm. (b) Cluster number distribution map of the CA algorithm corrected by the PPCC method. (c) Cluster number distribution map of the CA algorithm corrected by the NPCC1 method. (d) Cluster number distribution map of the CA algorithm corrected by the NPCC2 method. (e) Cluster number distribution map of the CANOPUS algorithm. (f) Cluster number distribution map of the CANOPUS algorithm corrected by the PPCC method. (g) Cluster number distribution map of the CANOPUS algorithm corrected by the NPCC1 method. (h) Cluster number distribution map of the CANOPUS algorithm corrected by the NPCC2 method.

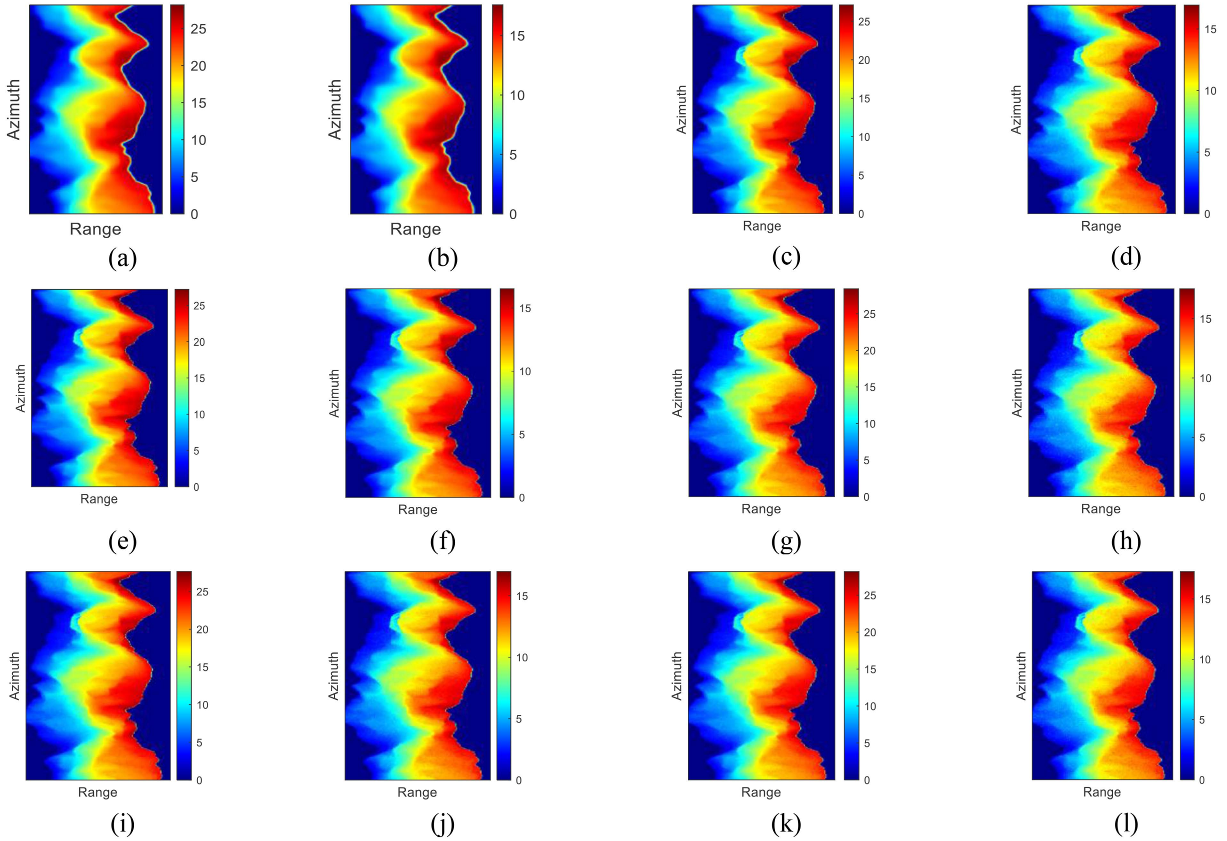


Fig. 9. Comparison of the unwrapped results after cluster correction for Experiment 2. (a) Unwrapped phase with the long baseline for the CA algorithm after cluster correction by the PPCC method (unit: rad). (b) Unwrapped phase with the short baseline for the CA algorithm after cluster correction by the PPCC method (unit: rad). (c) Unwrapped phase with the long baseline for the CA algorithm after cluster correction by the NPCC1 method (unit: rad). (d) Unwrapped phase with the short baseline for the CA algorithm after cluster correction by the NPCC1 method (unit: rad). (e) Unwrapped phase with the long baseline for the CA algorithm after cluster correction by the NPCC2 method (unit: rad). (f) Unwrapped phase with the short baseline for the CA algorithm after cluster correction by the NPCC2 method (unit: rad). (g) Unwrapped phase with the long baseline for the CANOPUS algorithm after cluster correction by the PPCC method (unit: rad). (h) Unwrapped phase with the short baseline for the CANOPUS algorithm after cluster correction by the PPCC method (unit: rad). (i) Unwrapped phase with the long baseline for the CANOPUS algorithm after cluster correction by the NPCC1 method (unit: rad). (j) Unwrapped phase with the short baseline for the CANOPUS algorithm after cluster correction by the NPCC1 method (unit: rad). (k) Unwrapped phase with the long baseline for the CANOPUS algorithm after cluster correction by the NPCC2 method (unit: rad). (l) Unwrapped phase with the short baseline for the CANOPUS algorithm after cluster correction by the NPCC2 method (unit: rad).

Fig. 8(a) is corrected by the PPCC method. Fig. 8(c) is the cluster number distribution map after Fig. 8(a) is corrected by the NPCC1 method. Fig. 8(d) is the cluster number distribution map after Fig. 8(a) is corrected by the NPCC2 method. Fig. 8(e) is the cluster number distribution map of the CANOPUS algorithm. Fig. 8(f) is the cluster number distribution map after Fig. 7(e) is corrected by the PPCC method. Fig. 8(g) is the cluster number distribution map after Fig. 8(e) is corrected by the NPCC1 method. Fig. 8(h) is the cluster number distribution map after Fig. 8(e) is corrected by the NPCC2 method. As can be seen from Fig. 8, most of the wrong cluster numbers have been corrected. For the CA algorithm, the execution time of the PPCC method is 2.327 s, the execution time of the NPCC1 method is 1.531 s, and the time of the NPCC2 method is 1.497 s. For the CANOPUS algorithm, the execution time of the PPCC method is 3.921 s, the execution time of the NPCC1 method is 2.161 s, and the execution time of the NPCC2 method is 1.927 s [CPU: Intel(R) Core(TM) i7-4510U 2.00 GHz, RAM: 8.00GB].

For the second experiment, Fig. 9(a) and (b) shows the unwrapped phases of the CA algorithm with the long and short baseline after cluster correction by the PPCC method. After NPCC1 is applied, the corresponding unwrapped phases are shown in Fig. 9(c) and (d). After NPCC2 is applied, the corresponding unwrapped phases are shown in Fig. 9(e) and (f). Fig. 9(g) and (h) shows the unwrapped phases of the CANOPUS algorithm with the long and short baselines after cluster correction by the PPCC method. After NPCC1 is applied, the corresponding unwrapped phases are shown in Fig. 9(i) and (j). After NPCC2 is applied, the corresponding unwrapped phases are shown in Fig. 9(k) and (l).

Table II shows the PUSR of the CA algorithm and the CANOPUS algorithm before and after cluster correction for Experiment 2. It can be seen from the table that after cluster correction, the PUSR is still greatly improved, indicating that the methods proposed in this article are also effective for real terrain.

TABLE II
COMPARION OF PUSR FOR EXPERIMENT 2

	PUSR of the CA algorithm		PUSR of the CANOPUS algorithm	
	Interferogram 1	Interferogram 2	Interferogram 1	Interferogram 2
Before correction	76.21%	75.33%	79.72%	79.79%
PPCC	99.01%	95.78%	93.81%	91.77%
NPCC1	98.12%	94.31%	90.42%	90.22%
NPCC2	97.99%	94.44%	90.77%	90.11%

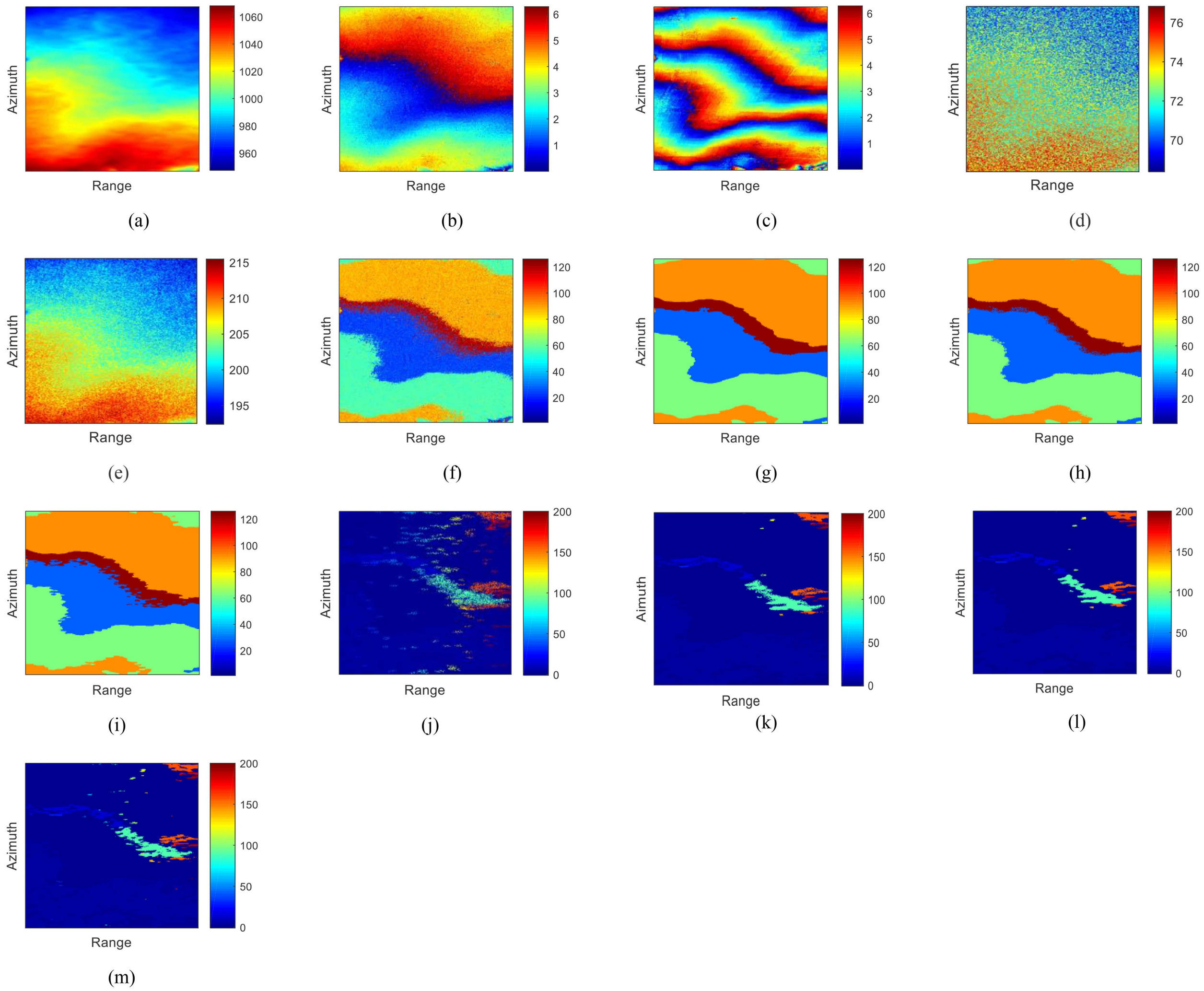


Fig. 10. TanDEM-X MB InSAR data set and clustering results of a mountainous area in Tongchuan, Shaanxi Province, China, 736×191 pixels. (a) Reference DEM (unit: m). (b) Noisy interferogram with the short baseline (unit: rad). (c) Noisy interferogram with the long baseline (unit: rad). (d) Noisy unwrapped phase with the short baseline (unit: rad). (e) Noisy unwrapped phase with the long baseline (unit: rad). (f) Cluster number distribution map of the CA algorithm. (g) Cluster number distribution map of the CA algorithm after cluster correction by the PPCC method. (h) Cluster number distribution map of the CA algorithm after cluster correction by the NPCC1 method. (i) Cluster number distribution map of the CA algorithm after cluster correction by the NPCC2 method. (j) Cluster number distribution map of the CANOPUS algorithm. (k) Cluster number distribution map of the CANOPUS algorithm after cluster correction by the PPCC method. (l) Cluster number distribution map of the CANOPUS algorithm after cluster correction by the NPCC1 method. (m) Cluster number distribution map of the CANOPUS algorithm after cluster correction by the NPCC2 method.

The third experiment is performed on a real and small-scale single-pass TanDEM-X InSAR data set (736 × 191 pixels). This is a mountainous area in Tongchuan City, Shaanxi Province, China. Fig. 10(a) is the reference DEM obtained by the Space Shuttle Radar Topographic Mapping mission. Fig. 10(b) and (c) shows the two interferograms with different baseline lengths, respectively. Fig. 10(d) and (e) shows the two unwrapped phases of the CA algorithm with two different baseline lengths,

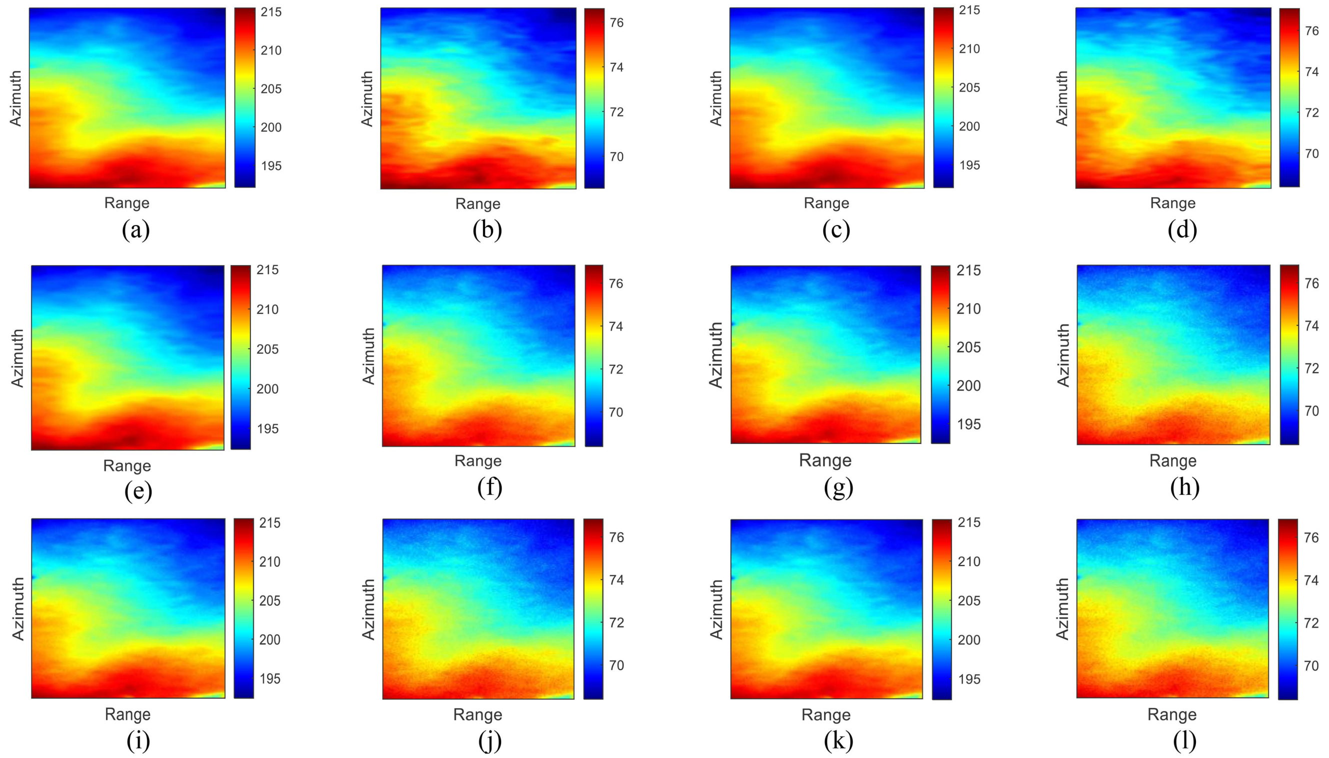


Fig. 11. Comparison of the unwrapped results after cluster correction for Experiment 3. (a) Unwrapped phase with the long baseline for the CA algorithm after cluster correction by the PPCC method (unit: rad). (b) Unwrapped phase with the short baseline for the CA algorithm after cluster correction by the PPCC method (unit: rad). (c) Unwrapped phase with the long baseline for the CA algorithm after cluster correction by the NPCC1 method (unit: rad). (d) Unwrapped phase with the short baseline for the CA algorithm after cluster correction by the NPCC1 method (unit: rad). (e) Unwrapped phase with the long baseline for the CA algorithm after cluster correction by the NPCC2 method (unit: rad). (f) Unwrapped phase with the short baseline for the CA algorithm after cluster correction by the NPCC2 method (unit: rad). (g) Unwrapped phase with the long baseline for the CANOPUS algorithm after cluster correction by the PPCC method (unit: rad). (h) Unwrapped phase with the short baseline for the CANOPUS algorithm after cluster correction by the PPCC method (unit: rad). (i) Unwrapped phase with the long baseline for the CANOPUS algorithm after cluster correction by the NPCC1 method (unit: rad). (j) Unwrapped phase with the short baseline for the CANOPUS algorithm after cluster correction by the NPCC1 method (unit: rad). (k) Unwrapped phase with the long baseline for the CANOPUS algorithm after cluster correction by the NPCC2 method (unit: rad). (l) Unwrapped phase with the short baseline for the CANOPUS algorithm after cluster correction by the NPCC2 method (unit: rad).

TABLE III
COMPARION OF PUSR FOR EXPERIMENT 3

	PUSR of the CA algorithm		PUSR of the CANOPUS algorithm	
	Interferogram 1	Interferogram 2	Interferogram 1	Interferogram 2
Before correction	73.11%	76.33%	78.72%	79.79%
PPCC	96.71%	97.78%	94.61%	95.77%
NPCC1	95.12%	94.81%	91.42%	93.22%
NPCC2	95.89%	94.94%	91.77%	90.11%

respectively. Fig. 10(f) is the cluster number distribution map of the CA algorithm. Fig. 10(g) is the cluster number distribution map of the CA algorithm after cluster correction by the PPCC method. Fig. 10(h) is the cluster number distribution map of the CA algorithm after cluster correction by the NPCC1 method. Fig. 10(i) is the cluster number distribution map of the CA algorithm after cluster correction by the NPCC2 method. Fig. 10(j) is the cluster number distribution map of the CANOPUS algorithm. Fig. 10(k) is the cluster number distribution map of the CANOPUS algorithm after cluster correction by the PPCC method. Fig. 10(l) is the cluster number distribution map of the CANOPUS algorithm after cluster correction by the NPCC1 method. Fig. 10(m) is the cluster number distribution map of the CANOPUS algorithm after cluster correction by the NPCC2 method. For the CA algorithm, the execution time of the PPCC method is 5.997 s, the execution time of the NPCC1 method is 3.618 s, and the execution time of the NPCC2 method is 3.566 s. For the CANOPUS algorithm, the execution time of the PPCC method is 8.745 s, the execution time of the NPCC1 method is 5.698 s, and the execution time of the NPCC2 method is 5.776 s [CPU: Intel(R) Core (TM) i7-4510U 2.00GHz, RAM: 8.00GB].

For the third experiment, Fig. 11(a) and (b) shows the unwrapped phases of the CA algorithm with the long and short baseline after cluster correction by the PPCC method. After NPCC1 is applied, the corresponding unwrapped phases are shown in Fig. 11(c) and (d). After NPCC2 is applied, the corresponding unwrapped phases are shown in Fig. 11(e) and (f). Fig. 11(g) and (h) shows the unwrapped phases of the CANOPUS algorithm with the long and short baseline after cluster correction by the PPCC method. After NPCC1 is applied, the corresponding unwrapped phases are shown in Fig. 11(i) and (j). After NPCC2 is applied, the corresponding unwrapped phases are shown in Fig. 11(k) and (l).

Table III shows the PUSR of the CA algorithm and the CANOPUS algorithm before and after cluster correction for Experiment 3. It can be seen from the table that after cluster correction, the PUSR is still greatly improved, indicating that the method proposed in this article is also effective for real MB InSAR data set.

According to the results of the above three experiments, it can be found that the two cluster correction methods, PPCC and NPCC, have their own advantages. The PPCC method is to correct all the pixels after clustering, which can also correct the core pixels with wrong cluster number and has a higher PUSR than the NPCC method. However, when the number of pixels is relatively large and the phase noise is relatively small, both methods can obtain good PUSR, but NPCC is more efficient than PPCC because it only needs to correct the core pixels.

V. Conclusion

MBPU is the key process of the MB InSAR technology, but due to the existence of phase noise, the PU results often contain errors. To improve the accuracy of the CA-based MBPU algorithms, two cluster correction methods are proposed in this article. One is the PPCC method, which takes the cluster number that appears the most times in the box as the final cluster number of the center pixel. The other method is the NPCC method, which divides all pixels into core pixels and noncore pixels according to the predefined density first, and then replaces the cluster number of the noncore pixels with the cluster number that appears most frequently in the selected box according to the ML criterion. The PPCC method is recommended when the interferogram is small or the phase noise is serious, so that the obtained PU results will have better PU accuracy. In the case of large interferogram scale and low noise, the NPCC method is recommended, which can improve the computational efficiency of the algorithm while ensuring the accuracy of the PU results. Experimental results verify the effectiveness of the proposed cluster correction methods.

References

[1] R. Bamler and P. Hartl, "Synthetic aperture radar interferometry," *Inverse Problems,* vol. 14, no. 4, pp. R1–R54, Aug. 1998.

[2] P. A. Rosen et al., "Synthetic aperture radar interferometry," *Proc. IEEE*, vol. 88, no. 3, pp. 333–382, Mar. 2000.

[3] H. Yu, Y. Lan, Z. Yuan, J. Xu, and H. Lee, "Phase unwrapping in InSAR: A review," *IEEE Geosci. Remote Sens. Mag.*, vol. 7, no. 1, pp. 40–58, Mar. 2019.

[4] D. C. Ghiglia and M. D. Pritt, *Two-Dimensional Phase Unwrapping: Theory, Algorithms, and Software*. New York, NY, USA: Wiley Interscience, 1998.

[5] K. Itoh, "Analysis of the phase unwrapping algorithm," *Appl. Opt.*, vol. 21, no. 14, pp. 2470–2470, 1982.

[6] W. Xu, E. C. Chang, L. K. Kwoh, H. Lim, W. Cheng, and A. Heng, "Phase unwrapping of SAR interferogram with multifrequency or multibasline," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 1994, pp. 730–732.

[7] V. Pascazio and G. Schirinzi, "Estimation of terrain elevation by multifrequency interferometric wide band SAR data," *IEEE Signal Process. Lett.*, vol. 8, no. 1, pp. 7–9, Jan. 2001.

[8] V. Pascazio and G. Schirinzi, "Multifrequency InSAR height reconstruction through maximum likelihood estimation of local planes parameters," *IEEE Trans. Image Process.*, vol. 11, no. 12, pp. 1478–1489, Dec. 2002.

[9] G. Ferraiuolo, V. Pascazio, and G. Schirinzi, "Maximum a posteriori estimation of height profiles in InSAR imaging," *IEEE Geosci. Remote Sens. Lett.*, vol. 1, no. 2, pp. 66–70, Apr. 2004.

[10] G. Ferraioli, A. Shabou, F. Tupin, and V. Pascazio, "Multichannel phase unwrapping with graph cuts," *IEEE Geosci. Remote Sens. Lett.*, vol. 6, no. 3, pp. 562–566, Jul. 2009.

[11] R. Ambrosino, F. Baselice, G. Ferraioli, and G. Schirinzi, "Extended Kalman filter for multichannel InSAR height reconstruction," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 10, pp. 5854–5863, Oct. 2017.

[12] G. Ferraioli, C.-A. Deledalle, L. Denis, and F. Tupin, "PARISAR: Patch based estimation and regularized inversion for multibaseline SAR interferometry," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 3, pp. 1626–1636, Mar. 2018.

[13] H. Yu and Y. Lan, "Robust two-dimensional phase unwrapping for multibaseline SAR interferograms: A two-stage programming approach," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 9, pp. 5217–5225, Sep. 2016.

[14] Y. Lan, H. Yu, and M. Xing, "Refined two-stage programming-based multibaseline phase unwrapping approach using local plane model," *Remote Sens.*, vol. 11, no. 5, 2019, Art. no. 491.

[15] Y. Gao., S. Zhang, T. Li, Q. Chen, X. Zhang, and S. Li, "Refined two-stage programming approach of phase unwrapping for multi-baseline SAR interferograms using the unscented kalman filter," *Remote Sens.*, vol. 11, no. 2, 2019, Art. no. 199.

[16] Y. Lan, H. Yu, Z. Yuan, and M. Xing, "Comparative study of DEM reconstruction accuracy between single-and multibaseline InSAR phase unwrapping," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art no. 5220411.

[17] H. Yu, H. Lee, N. Cao, and Y. Lan, "Optimal baseline design for multibaseline InSAR phase unwrapping," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 8, pp. 5738–5750, Aug. 2019.

[18] H. Yu, M. Xing, and Z. Yuan, "Baseline design for multibaseline InSAR system: A review," *IEEE J. Miniaturization for Air Space Syst.*, vol. 2, no. 1, pp. 17–24, Mar. 2021.

[19] L. Zhou, H. Yu, Y. Lan, and M. Xing, "Artificial intelligence in interferometric synthetic aperture radar phase unwrapping: A review," *IEEE Geosci. Remote Sens. Mag.*, vol. 9, no. 2, pp. 10–28, Jun. 2021.

[20] Z. Yuan, T. Chen, X. Xing, W. Peng, and L. Chen, "BM3D denoising for a cluster-analysis-based multibaseline InSAR phase-unwrapping method," *Remote Sens.*, vol. 49, no. 8, Jan. 2022, Art. no. 1836.

[21] Z. Yuan, Y. Deng, F. Li, R. Wang, G. Liu, and X. Han, "Multichannel InSAR DEM reconstruction through improved closed-form robust Chinese remainder theorem," *IEEE Geosci. Remote Sens. Lett.*, vol. 10, no. 6, pp. 1314–1318, Nov. 2013.

[22] H. Liu, M. Xing, and Z. Bao, "A cluster-analysis-based noise-robust phase-unwrapping algorithm for multibaseline interferograms," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 1, pp. 494–504, Jan. 2015.

[23] Z. Jiang, J. Wang, Q. Song, and Z. Zhou, "A refined cluster analysis-based multibaseline phase-unwrapping algorithm," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 9, pp. 1565–1569, Sep. 2017.

[24] Z. Yuan, Z. Lu, L. Chen, and X. Xing, "A closed-form robust cluster-analysis-based multibaseline InSAR phase unwrapping and filtering algorithm with optimal baseline combination analysis," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 6, pp. 4251–4262, Jun. 2020.

[25] Z. Yuan, T. Chen, X. Xing, W. Peng, and L. Chen, "BM3D denoising for a cluster-analysis-based multibaseline InSAR phase-unwrapping method," *Remote Sens.*, vol. 14, no. 8, 2022, Art. no. 1836.

[26] L. Zhou, H. Yu, Y. Lan, S. Gong, and M. Xing, "CANet: An unsupervised deep convolutional neural network for efficient cluster-analysis-based multibaseline InSAR phase unwrapping," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5212315.

Zhihui Yuan (Member, IEEE) was born in Hunan, China, in 1983. He received the B.S. degree in electronic information engineering from Xiangtan University, Xiangtan, China, in 2007 and the Ph.D. degree in communication and information systems from the Institute of Electronics, Chinese Academy of Sciences, Beijing, China, in 2013.

Since 2013, he has been a Lecturer with the School of Electrical and Information Engineering, Changsha University of Science and Technology, Changsha, China. From 2018 to 2019, he was a Visiting Scholar with the Roy M. Huffington Department of Earth Sciences, Southern Methodist University, working with Professor Zhong Lu on InSAR technique. He is the Principal Investigator of the scientific project “Research on Multichannel InSAR Robust DEM Inversion with High Precision for Complicated Terrain” supported by the National Natural Science Foundation of China. His research interests include signal processing and the application of synthetic aperture radar interferometry, phase unwrapping, high-resolution digital elevation model generation, and algorithm design.

Tianjiao Chen was born in Shandong, China, in 1996. He received the M.S. degree in electronic science and technology from the Changsha University of Science and Technology, Changsha, China, in 2022.

His research interests include the application of the interferometric synthetic aperture radar technique about multibaseline phase unwrapping.

Hanwen Yu (Senior Member, IEEE) received the B.S. and Ph.D. degrees from Xidian University, Xi’an, China, in 2007 and 2012, respectively.

He is currently a Full Professor with the School of Resources and Environment, University of Electronic Science and Technology of China, Chengdu, China and an Adjunct Full Professor with the Academy of Advanced Interdisciplinary Research, Xidian University, Xi’an, China and the Department of Engineering, University of Napoli, Parthenope, Italy. He was a Postdoctoral Fellow with the Department of Civil and Environmental Engineering and the National Center for Airborne Laser Mapping, University of Houston. He has authored more than 60 scientific papers and given scientific presentation about “Advanced techniques in InSAR Phase Unwrapping” invited by the IEEE GRSS Webinar in 2021. His research focuses on InSAR, and this work has led to new insights into the world-wide deformation monitoring and topographic mapping. He has reviewed more than 280 manuscripts for more than 20 different journals.

Dr. Yu has been involved in IEEE in general and IEEE Geoscience and Remote Sensing Society (IEEE GRSS) in particular. He is elected as 2019 IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING Best Reviewer. He is also elected as IEEE GRSS AdCom member beginning in 2023. He is the Editor-in-Chief of *IEEE Journal on Miniaturization for Air and Space Systems*, the Topical Associate Editor of IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING, and the Associate Editor of *IEEE Geoscience and Remote Sensing Magazine*. He is the recipient of several awards and honors from IEEE GRSS, including the 2021 Transactions Prize Paper Award. He is the Technical Program Committee Member and Session Chair of 2022 IGARSS and the Co-PI of the IEEE GRSS 50%-funding project “West China Initiative.”

Wei Peng received the Ph.D. degree in geodesy from Central South University, Changsha, China, in 2020.

He is currently a Lecturer with the Changsha University of Science and Technology, Changsha, China. His research interests include GPS and InSAR data processing and nontectonic deformation.

Lifu Chen (Member, IEEE) was born in February 1979. He received the B.S. degree in electronic information engineering from Beihang University, Beijing, China, in 2007, and the Ph.D. degree in communication and information systems from the Institute of Electronics, Chinese Academy of Sciences, Beijing, China, in 2011.

Since 2011, he has been a Lecturer with the School of Electrical and Information Engineering, Changsha University of Science and Technology, Changsha, China. He is the author of more than 40 articles. From August 2018 to August 2019, he was a Visiting Scholar with the School of Engineering, Newcastle University, U.K., working with Professor Zhenhong Li on AI in SAR/InSAR images. He is the Principal Investigator of the scientific project “Research on Real-time DEM reconstruction with High Precision using airborne InSAR system” supported by the National Natural Science Foundation of China. His research interests include SAR/InSAR signal processing, targets detection, and terrain classification for SAR/InSAR image with deep learning (landslide detection, classification for remote sensing images, extraction of water, shadow and layover from InSAR images, etc.).

Xuemin Xing (Member, IEEE) was born in Liaoning, China, in 1983. She received the B.S. degree in survey and mapping engineering and the M.S. degree in geomatics and surveying from Central South University, Changsha, China, in 2005 and 2008, respectively.

From 2016 to 2017, she was a Visiting Scholar with the Department of Environmental Sciences, Macquarie University, Sydney, NSW, Australia. She is currently an Associate Professor with the School of Traffic and Transportation Engineering, Changsha University of Science and Technology, Changsha, China, where she is also a key member of the Laboratory of Radar Remote Sensing Applications. She has authored more than 40 articles and has made 10 inventions. Her research interests include the application of time-series interferometric synthetic aperture radar technique in highway deformation monitoring, deriving spatial–temporal large-scale deformation induced by mining activities and modeling, and the integration of persistent scatterer, corner reflector, and small baseline subset technique.

Dr. Xing is a reviewer for three international journals and four national journals in China.