An Application of Continuous Weights to Phase Unwrapping

Non-member Tao LU (The University of Tokushima)
Member Shunichiro OE (The University of Tokushima)
Non-member Hiroshi MURAI (Shikoku University)

Two-dimensional phase unwrapping is a key step in extraction of digital elevation models (DEMs) from interferometric synthetic aperture radar (IFSAR) data. Least-squares (LS) algorithm is one of major approaches to phase unwrapping, and weighted LS algorithm offers great potential if weights are appropriately obtained. However, the determination of weights is still a critical problem. As noise increases, binary weights becomes more difficult to be determined, because of high possibility of coexisting of noise pixels and non-noise pixels in same histogram bin. Hence instead of determining binary weights, this paper presents an effective approach to determining continuous weights by incorporating weighted LS algorithm into a minimization method. The continuous weights are estimated by minimizing a norm of the differences between real values and calculated ones. The proposed approach is illustrated through a simulative example in which the input data are corrupted by both uniformly distributed noise and scattered noise. By comparing errors of phase unwrapping by binary weights and those by continuous weights, the latter turns out to be superior to the former.

Keywords: continuous weights, phase unwrapping, IFSAR, NLSSQP method, optimization

1. Introduction

Synthetic aperture radar interferometry is a technique for the generation of high-resolution DEMs. Phase unwrapping is a key processing step in extracting DEMs from IFSAR data\(^1\)\(^2\)\(^3\). The main processing steps of IFSAR are shown in Fig.1. An IFSAR image (i.e., phase difference image) can be formed by multiplying the complex reflectivity at each point of one image (i.e., master SLC image) by its conjugate value in the second image (i.e., slave SLC image). The information of ground elevation is involved in this phase difference image. However, the phase difference is only in modulo \(2\pi\) (wrapped form). To estimate the elevation at each point of the image, the correct integer number of \(2\pi\) phase cycles must be added to each wrapped phase value. This processing in which \(2\pi\) ambiguity is solved is called “phase unwrapping”. Phase unwrapping algorithms can be organized into two main classes\(^4\): path-following algorithms\(^5\)\(^6\) and minimum-norm algorithms\(^7\)\(^8\)\(^9\)\(^10\). The path-following algorithms use localized operations that follow paths through the wrapped path; while the minimum-norm algorithms take a more global approach that seeks to minimize some measure of the difference between the gradients of the wrapped phase and those of the unwrapped phase. Both algorithms can work well if the images do not contain noises.

When certain phase values are corrupted due to noise, the corrupted phase values should be properly weighted in order that they do not affect the unwrapping results. Ghiglia\(^8\) assumed that if we have some additional information that allows us to define a corresponding weighting array, we could prevent the noise phase value from having any influence on the results by assigning a weight of zero to this inconsistent data. In his study, the weighting array is defined in the case of uniformly distributed noise. Pritt\(^9\) introduced an algorithm of defining the initial phase weights not only for uniformly distributed noise but also for scattered noise. In the algorithm, the weighting array was defined based on the standard deviations of phase partial derivatives and the threshold was determined as the minimum location of re-mapped histogram bins.

The weighting array mentioned above is binary weights. The challenge of defining a good binary weighting array is to define the threshold so that total number of zero-weighted pixels is small but the corrupted phases can be picked out. Otherwise, if the threshold is too low, not enough pixels will be mask out, which causes many low-quality phase values to corrupt the unwrapped solution; on the other hand, if the threshold is too high, then too many pixels will be mask out, so that regions will be isolated from one another. In binary weight determination, the threshold, which divides pixels into noise pixels and non-noise pixels, is decided by standard deviation values. The pixels of same standard deviation values may properly be included in same histogram bin; for example, the noise pixels and the non-noise ones, which have same standard deviation values, may coexist in the same bin. Thus, if one bin is determined as zero, some non-noise pixels will simultaneously be regarded as noise pixels, vice versa. The stronger the noise is, the higher the possibility of coexisting of noise pixels and non-noise pixels in the same bin is. Therefore, it is not always easy to simply decide one bin to be zero or unity as noise increases, and it is
boundaries by means of the boundary conditions:

\[
\Delta_{0,j}^x = - \Delta_{i,j}^x, \quad \Delta_{M+1,j}^x = - \Delta_{M,j}^x \\
\Delta_{0,j}^y = - \Delta_{i,j}^y, \quad \Delta_{N+1,j}^y = - \Delta_{N,j}^y
\]  

The differences between the partial derivatives of the solution and the partial derivatives described by Eq.(1) and Eq.(2) must be minimized as follows

\[
\sum_{ij} w_{ij}^x (\phi_{ij} - \phi_{i-1,j} - \Delta_{ij}^x)^2 + \sum_{ij} w_{ij}^y (\phi_{ij} - \phi_{i,j-1} - \Delta_{ij}^y)^2 \Rightarrow 0
\]  

where \( w_{ij}^x \) and \( w_{ij}^y \) are weighting arrays expressed by

\[
w_{ij}^x = \min(w_{ij}^x, w_{i-1,j}^x) \\
w_{ij}^y = \min(w_{ij}^y, w_{i,j-1}^y)
\]

The least squares solution to this problem is defined by the equation

\[
w_{i+1,j}^x (\phi_{i+1,j} - \phi_{ij}) - w_{ij}^x (\phi_{ij} - \phi_{i-1,j}) + w_{i+1,j}^y (\phi_{i,j+1} - \phi_{ij}) - w_{ij}^y (\phi_{ij} - \phi_{i,j-1}) = \hat{\rho}_{ij}
\]

where \( \hat{\rho}_{ij} \) is defined by

\[
\hat{\rho}_{ij} = w_{i+1,j}^x \Delta_{i+1,j}^x - w_{ij}^x \Delta_{ij}^x + w_{i,j+1}^y \Delta_{i,j+1}^y - w_{ij}^y \Delta_{ij}^y
\]

The classical Gauss-Seidel relaxation method solves Eq.(8) by iterating on the following equation

\[
\phi_{ij} = (w_{i+1,j}^x \phi_{i+1,j} + w_{i,j+1}^y \phi_{i,j+1} + w_{ij}^y \phi_{i,j-1} - \hat{\rho}_{ij})/v_{ij}
\]

where \( v_{ij} \) is defined by

\[
v_{ij} = w_{i+1,j} + w_{i,j+1} + w_{ij}
\]

2.2 NLSSQP Method

NLSSQP method is one of the optimization methods in which Sequential Quadratic Programming(SQP) is employed to solve Non-linear Least Square (NLS) problems with constrained conditions. The weight values are estimated by minimizing a norm of the difference between real values and calculated ones. The relationship between real values and the calculated ones is

\[
\phi^0 = \phi(w, \psi) + r(w)
\]

where
$g^0$: real value vector; $\phi(w, \psi)$: calculated value vector; $w$: weight vector; $\psi$: known input data vector; $r(w)$: error vector.

The most comprehensive function to be minimized in the weight value estimation procedure is given as:

$$
\min \{ f(w) \} \Rightarrow 0 \quad \cdots \cdots \cdots \cdots \cdots \quad (13)
$$

where

$$
f(w) = \frac{1}{2} \sum_{i=1}^{M} \sum_{j=1}^{M} (r_{ij}(w))^2 \quad \cdots \cdots \cdots \cdots \cdots \quad (14)
$$

$$
r_{ij} = \phi_{ij}^0 - \phi_{ij} \quad \cdots \cdots \cdots \cdots \cdots \quad (15)
$$

Eqs. (13) and (14) are commonly called Least Square problem. Due to the complexity of the relationship of weights and unwrapped phase values, the relation between $r(w)$ and $w$ is usually nonlinear, therefore the problem described above is a NLS problem. In order to solve such a problem, this paper employs the NLSSQP method\(^{(13)}\). The detailed procedure is described in the following.

First let’s see the following nonlinear least square problem:

Constrained conditions:

$$
g(w) < 0, \quad g(w) = (g_1(w), \ldots, g_m(w))^T \quad \cdots \cdots \cdots \cdots \cdots \quad (16)
$$

$$
h(w) = 0, \quad h(w) = (h_1(w), \ldots, h_m(w))^T \quad \cdots \cdots \cdots \cdots \cdots \quad (17)
$$

and the objective function is

$$
\min f(w) = \min \left\{ \frac{1}{2} \sum_{i=1}^{M} \sum_{j=1}^{N} (r_{ij}(w))^2 \right\} \quad \cdots \cdots \cdots \cdots \cdots \quad (18)
$$

Denoting the Lagrangian multiplier vectors of constraints $g(w) < 0$ and $h(w) = 0$ by $\lambda$ and $\mu$, the Lagrangian function is defined by:

$$
L(w, \lambda, \mu) = \frac{1}{2} r(w)^T r(w) + \lambda^T g(w) + \mu^T h(w) \quad \cdots \cdots \cdots \cdots \cdots \quad (19)
$$

The Hesse matrix corresponding to $w$ is given as:

$$
\nabla_{xx} L = J(w)^T J(w) + \sum_{j=1}^{M \times N} r_j(w) \nabla^2 r_j(w) + \sum_{i=1}^{m} \lambda_i \nabla^2 g_i(w) + \sum_{j=1}^{l} \mu_j \nabla^2 h_j(w) \quad \cdots \cdots \cdots \cdots \cdots \quad (20)
$$

where $J(w)$ stands for the Jacobi matrix of $r(w)$. The $B$ matrix in SQP method can be expressed as:

$$
B_k = J(w_k)^T J(w_k) + A_k + C_k \quad \cdots \cdots \cdots \cdots \cdots \quad (21)
$$

Then the algorithms of NLSSQP method can be described briefly as follows:

Step 0: To set up initial $w_0$, the matrix of $A_0$, $C_0$ and parameter $\delta > 0$, $\omega \in (0, 0.5)$, $\tau \in (0, 1)$, $k = 0$.

Step 1: When the $w_k$, $A_k$, $C_k$ are known, we can solve the following QP problem with respect to $d$.

$$
\minimize \frac{1}{2} d^T (J(w_k)^T J(w_k) + A_k + C_k) d + r(w_k)^T J(w_k) d \quad \cdots \cdots \cdots \cdots \cdots \quad (22)
$$

subject to

$$
g(w_k) + \nabla g(w_k) d < 0 \quad \cdots \cdots \cdots \cdots \cdots \quad (23)
$$

$$
h(w_k) + \nabla h(w_k) d = 0 \quad \cdots \cdots \cdots \cdots \cdots \quad (24)
$$

The solution of this problem is $d_k$ by which we can determine the searching direction. Correspondingly, Lagrangian multiplier vectors of constraints $g(w) < 0$ and $h(w) = 0$ become $\lambda_{k+1}$ and $\mu_{k+1}$, respectively.

Step 2: If $(w_k, \lambda_{k+1}, \mu_{k+1})$ satisfy convergent conditions, then stop; otherwise, go to step 3 to judge the convergence.

Step 3: To determine $\alpha_k$ in the following procedure (Line Search):

Step 3.1: Let $\nu_{k+1} = 1$ and $j = 1$.

Step 3.2: Regarding the following line search evaluation function:

$$
\theta_k(w) = f(w) + \delta \max(w, 0, g_1(w), \ldots, g_m(w), h_1(w), \ldots, h_m(w)) \quad \cdots \cdots \cdots \cdots \cdots \quad (25)
$$

If the next inequality is satisfied,

$$
\theta_k(w_k + \nu_{k} d_k < \theta_k(w_k) - \omega \nu_{k} d_k^T (J(w_k)^T J(w_k) + A_k + C_k) d_k \quad \cdots \cdots \cdots \cdots \cdots \quad (26)
$$

then let $\alpha_k = \nu_{k}$, and go to Step 4; otherwise, go to Step 3.3.

Step 3.3: let $\nu_{k+1} = \tau \nu_{k}$, $j = j + 1$, then go to Step 3.2.

Step 4: Let $w_{k+1} = w_k + \alpha_k d_k$.

Step 5: Renew the matrices of $A_k$ and $C_k$ to produce $A_{k+1}$ and $C_{k+1}$.

Step 6: Let $k = k + 1$, go to Step 1.

The computation procedure is briefly summarized in Fig.2.

3. Results and Discussions

3.1 The Results of Unweighted Phase Unwrapping

In order to illustrate the effectiveness of the present approach, we begin with simulated phase data of a 256 × 256 pixels without noise as shown in Fig.3(1). These phase data increase linearly from the upper left corner
Fig. 2. The flow chart of the proposed approach.

Fig. 3. (1) wrapped phase input without noise; (2) unwrapped phase solution by unweighted LS; (3) error map.

to the lower right corner. The unwrapped solution by unweighted LS method is given in Fig.3(2). We could not judge from Fig.3(2) whether the unwrapping is correct or not, because the range of the unwrapped solution is so large. Hence, error map, which shows the difference between unwrapped phase solution and real phase data, is used. Unlike path-following algorithms, least-squares algorithms do not produce an unwrapped surface that is congruent to the wrapped phase, so it is necessary to make surface congruent by congruence operation before making error map. Fig.3(3) is the error map, where black indicates no difference. Because the original input is noise free and totally consistent, the unwrapped phase solution and real phase data are always agreement and phase unwrapping by unweighted LS method is perfect.

Next, the noise free data depicted in Fig.3(1) are corrupted with both uniformly distributed noise and scattered noise (Fig.4(1)); correspondingly the unwrapped solution by unweighted LS and the error map are shown in Fig.4(2)-(3). Because of noise, the differences (gray level from grey to white indicates difference from small to large) exist in most parts of the error map of Fig.4(3). Therefore we can conclude that the unwighted phase unwrapping is not effective for noise data, and that the weighted phase unwrapping is necessary to improve such unwrapped solution.

### 3.2 The Results of Binary Weight Phase Unwrapping

Before we apply binary weight phase unwrapping, we first explain the algorithm of Pritt for automatic determinining threshold. The algorithm can be concisely described in the following three steps:

- **Step 1**: The phase standard deviation values are calculated by Eq.(27)

  \[
  Z_{mn} = \sqrt{\sum (\Delta x_{ij} - \bar{\Delta}_{mn})^2} + \sqrt{\sum (\Delta y_{ij} - \bar{\Delta}_{mn})^2} / k \]

  \[
  \cdots \cdots \cdots (27)
  \]

  where \( \Delta x_{ij} \) and \( \Delta y_{ij} \) are the partial derivatives of the phase defined as Eq.(1) and Eq.(2); \( \bar{\Delta}_{ij} \) are the averages of these partial derivatives in the \( k \times k \) windows.

- **Step 2**: The histogram of the phase deviation values is obtained, in which the values are scaled to range 0-1 and placed into ten histogram bins. Then the values are re-mapped again into ten histogram bins in which the first and last bins each contain about 5 percent of re-mapped values. At last, the location of the minimum is selected as threshold.

- **Step 3**: The values larger than the threshold are defined to be zero, and the remaining values are defined as unity.

Fig.5 shows the re-mapped distribution histogram and the automatically selected threshold by the algorithm of Pritt. Fig.6(1) shows binary-valued weight data. In the figure, the black pixels are defined to be zero and the remaining pixels are defined to be unity, in order that the influence of noise phase data can be prevented by assigning a weight of zero to the inconsistent. From error map of Fig.6(3), it is shown that the differences still exist in the image, especially at the left up corner and in the location of uniformly distributed noise. But if Fig.6(3) is compared with Fig.4(3), it can be found that binary weight phase unwrapping is better than unweighted phase unwrapping. It is apparent, however, that the binary weight phase unwrapping can not work well. It is for this reason that the present paper attempts to propose a continuous weight phase unwrapping.

### 3.3 The Results of Continuous Weight Phase Unwrapping

#### 3.3.1 Definition of Initial Weights

Theoretically, each pixel needs a weight, but it is impossible for us to calculate the weight of every pixel because too
Fig. 5. Histogram of binary weight distribution

Fig. 6. (1) binary-valued weight data; (2) unwrapped phase solution; (3) error map.

Fig. 7. Histogram of continuous weight distribution

Fig. 8. (1) continuous-valued weight data; (2) unwrapped phase solution; (3) error map.

many weights will cost much computation time. Hence after Step 1 and Step 2 as described in section 3.2, we need to proceed the following Step 3.

Step 3: The initial weights of the pixels are defined in ten parts (W1, W2, ..., W10) according to ten histogram bins. For example the initial weights of pixels whose phase deviation values are between 0.0 – 0.1 are defined as W1.

3.3.2 Constrained Conditions In order to make the NLSSQP method more effective, the following constrained conditions of weights are defined:

\[
g_j(W) = W_i - 1 < 0 \quad \cdots \quad (28)
\]

\[
g_j(W) = -W_i < 0 \quad \cdots \quad (29)
\]

\[
g(W_i) - g(W_{i-1}) > 0 (i = 2, \ldots, 9) \quad \cdots \quad (30)
\]

Without the constrained conditions, the computation time will be considerably long; and sometimes, we will obtain irrational results such as negative values or the ones larger than unity.

3.3.3 Results Fig.7 shows the selected continuous weights by the present approach. The first and the last bins are defined as 0.0 and 1.0 which are the same as binary weights, the others are defined between 0.0 to 1.0. Fig.8(1) shows continuous-valued weight data. Fig.8(2) and Fig.8(3) show the results of unwrapped data and error map.

Comparing the unwrapped solution and error map by continuous weights with those by binary weights, we can see that unwrapped solution and error map by continuous weights(Fig.8(2)-(3)) are better than those by binary weights(Fig.6(2)-(3)). Not only the errors are reduced at the left up corner but also the errors within the location of uniformly distributed noise are greatly reduced.

In order to estimate the approach quantitatively, we use two Mean Square Errors:

\[
e_1 = \frac{1}{MN} \sum_{i=1}^{M-1} \sum_{j=0}^{N-1} w_{ij}^x (\phi_{ij} - \phi_{i-1,j} - \Delta \phi_{ij})^2
\]

\[
+ \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=1}^{N-1} w_{ij}^y (\phi_{ij} - \phi_{i,j-1} - \Delta \phi_{ij})^2 \quad (31)
\]

\[
e_2 = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} (\phi_{ij} - \phi_{ij}^0)^2 \quad \cdots \quad (32)
\]

where \(e_1\) is a convergent error in phase unwrapping, and \(e_2\) is an absolute error in NLSSQP method. Table 1 lists weights, \(e_1\) and \(e_2\) of unweighted phase unwrapping, binary weight phase unwrapping and continuous weight phase unwrapping, respectively. As can be seen, unweighted phase unwrapping does not work well for noise input data, while weighted phase unwrapping is effective. Comparing two kinds of weighted phase unwrapping, not only the \(e_1\) of continuous weight phase unwrapping is about 10 times smaller than that of binary weight phase unwrapping, but also the \(e_2\) of the former is about 30 times smaller than that of the latter.

The advantage of continuous weight phase unwrapping is more evident on the estimation of \(e_2\), which conforms to the aim of phase unwrapping to obtaining unwrapped data closest to real data. So it can be said that contin-
uous weight unwrapping is obviously superior to binary phase unwrapping on this aspect.

4. Conclusions

The unwrapped phase unwrapping can work well only in the case of noise free. When phase values are corrupted due to noise (e.g., uniformly distributed noise and scattered noise), the weighted phase unwrapping becomes desirable, in which the weight values are usually treated as binary ones. However, the binary weights do not always work as noise increases. In order to improve the weighted phase unwrapping, this paper has developed an approach to searching for the continuous weights by incorporating weighted phase unwrapping into an optimal method. The continuous weights are evaluated by minimizing a norm of the difference between real values and calculated ones. The results of the illustrative example indicated that the unwrapping solutions obtained by continuous weights were better than those by binary weights.

This approach is not perfect yet, though it has the potential of determining continuous weights. However, the advantage of this method is that the phase unwrapping algorithm is only one of the subroutines of the proposed approach, so other more effective phase unwrapping method can be incorporated into it.

(Manuscript received April 27, 2000, revised November 24, 2000)

References