Journal of
Applied Remote Sensing

RemoteSensing.SPIEDigitalLibrary.org

# Comparison of optimization algorithms for interferometric synthetic aperture radar phase unwrapping based on identical Markov random fields

Lifan Zhou
Dengfeng Chai
Yu Xia
Peifeng Ma

SPIE.

Lifan Zhou, Dengfeng Chai, Yu Xia, Peifeng Ma, "Comparison of optimization algorithms for interferometric synthetic aperture radar phase unwrapping based on identical Markov random fields," *J. Appl. Remote Sens.* **12**(2), 025016 (2018), doi: 10.1117/1.JRS.12.025016.

# Comparison of optimization algorithms for interferometric synthetic aperture radar phase unwrapping based on identical Markov random fields

**Lifan Zhou,[a,*] Dengfeng Chai,[b] Yu Xia,[a] and Peifeng Ma[c]**

[a]Changshu Institute of Technology, School of Computer Science and Engineering, Changshu, China

[b]Zhejiang University, College of Earth Science, Hangzhou, China

[c]Chinese University of Hong Kong, Institute of Space and Earth Information Science, Hong Kong, China

**Abstract.** Phase unwrapping (PU) is one of the key processes in measuring the elevation or deformation of the Earth's surface from its interferometric synthetic aperture radar (InSAR) data. PU problems may be formulated as maximum *a posteriori* estimation estimations of Markov random field (MRF). The key issue of this formulation is energy minimization. Iterated conditional mode (ICM), graph cuts (GC), loopy belief propagation (LBP), and sequential tree-reweighted message passing (TRW-S) have been proposed for the energy minimization. Unfortunately, they differ in the formulation of the MRF model for PU, which raises the question of how they compare against each other on the same MRF model for PU. We address this by investigating the four optimization algorithms and comparing them on an identical MRF model, which gives researchers some guidance as to which optimization method is best suited for solving the PU problem. Experiments using simulated and real-data illustrate that the GC algorithm is clearly the winner among the four algorithms in all cases. The ICM algorithm, although very rapid, performs much worse than the other three especially in the terrain with violent changes or discontinuities. The two message-passing algorithms—LBP and TRW-S—perform completely differently. The LBP algorithm performs surprisingly poorly on solving phase discontinuities issue, whereas the TRW-S algorithm does quite well (second only to the GC algorithm). [DOI: 10.1117/1.JRS.12.025016]

**Keywords:** interferometric synthetic aperture radar; Markov random fields; optimization algorithms; phase unwrapping.

Paper 171031 received Nov. 30, 2017; accepted for publication May 30, 2018; published online Jun. 19, 2018.

## 1 Introduction

Interferometric synthetic aperture radar (InSAR) is a powerful tool not only for the reconstruction of the topography,[1] but also for the Earth's surface deformation retrieval.[2,3] An InSAR system makes use of two or more SAR images covering the same scene to generate interferograms. The acquired interferometric phase is the principal value of the absolute phase, which corresponds to topography or deformation and is called the wrapped phase. Phase unwrapping (PU) is the process of resolving the absolute phase through the wrapped phase. PU is based on the assumption that the absolute value of phase difference between any two neighboring pixels is less than $\pi$, termed the Itoh condition.[4] The Itoh condition can be violated if the absolute phase surface is discontinuous or if the wrapped phase is noisy. The existence of low coherence regions and rapid-phase variations poses challenges to two dimensional (2-D) PU algorithms.

[*]Address all correspondence to: Lifan Zhou, E-mail: zhoulifan_rs@163.com

PU approaches comprise three main classes: path following, minimum norm, and Bayesian approaches. Path following algorithms apply line integration schemes over the wrapped phase image and rely on the assumption that the Itoh condition holds along the integration path. Wherever this condition fails, different integration paths may lead to different unwrapped phase values. Path following algorithms include branch cuts,[5,6] quality-guided,[7,8] mask cut,[4,9] and minimum discontinuity.[10,11] These approaches are sensitive to Itoh conditions that are often violated in 2-D cases. The second category of algorithms is minimum norm methods, which attempt to find a phase solution for which the $L^\rho$ norm of the difference between absolute phase differences and wrapped phase differences is minimized. In the case $\rho = 2$, we have a least-squares method.[12] A drawback of the $L^2$ norm-based criterion is that it tends to smooth discontinuities, unless they are provided as binary weights.[13] The minimum cost flow (MCF) algorithm, the $\rho = 1$ case of the $L^\rho$ norm method, was proposed by Costantini.[14] $L^1$ norm-based criterion performs better than $L^2$ norm in preserving discontinuities. Obviously, $L^0$ norm is the most desirable in practice as it tends to be more accurate than other $L^\rho$ criteria, but it was proved an NP-hard problem by Chen;[15] hence, only approximated $L^0$ norm solutions can be achieved.[4,16] In recent years, as the multitemporal/multibaseline InSAR attracted increasing interests, it is with significance to unwrap interferogram series to support these techniques, e.g., permanent scatterer SAR interferometry (PSInSAR),[17] small baseline subset (SBAS).[18] In 2004, Kampes and Hanssen[19] proposed a least-squares AMBiguity decorrelation adjustment algorithm, which is borrowed from GPS technique to solve the phase ambiguity problem in PSInSAR. In 2007, Hooper and Zebker[20] presented a technique for three-dimensional (3-D) PU in PSInSAR, which relies on first unwrapping in 1-D, then iteratively improving this solution in the other two dimensions. In 2011, Pepe et al.[21] developed an extended 3-D (2-D image plus 1-D time) MCF algorithm in SBAS technique. In 2012, Costantini et al.[22] proposed a general formulation for multidimensional PU, which exploits a highly redundant set of measurements of finite differences to perform PU with a reliable and accurate integration.

The Bayesian approach relies on a data observation mechanism model as well as a priori knowledge of the phase to be modeled.[23] One of the reasons why this framework is so popular is that it can be justified in terms of maximum *a posteriori* estimation (MAP) of a Markov random field (MRF),[24] whose major advantage lies in their ability to take contextual information into account.[25] Hence, PU problems are naturally represented in terms of energy minimization, where the energy minimization approaches are originally used, such as iterated conditional modes (ICM)[26] or simulated annealing (SA).[27] For instance, in 2004, Ferraiuolo et al.[28] proposed a solution of the MAP problem for PU achieved using ICM and showed the good performances of the method. In 2006, Ying et al.[29] proposed an MRF approach utilizing an efficient algorithm for parameter estimation using a series of dynamic programming connected by the ICM algorithm. In 2013, Chen et al.[30] proposed an integrated denoising and unwrapping of InSAR phase based on MRF using SA algorithm. However, the two well-known optimization approaches, i.e., ICM and SA, originally used are proved to be either ineffective or extremely inefficient.

Over the last few years, energy minimization approaches have had a renaissance, primarily due to powerful new optimization algorithms, such as graph cuts (GC),[31,32] loopy belief propagation (LBP),[33] and sequential tree-reweighted message passing (TRW-S).[34,35] These new optimization methods have proven to be very powerful for solving vision problems, such as stereomatching, photomontage, and image segmentation. In the realm of PU, the new optimization algorithms have been proposed in several literatures. For example, in 2001, Frey et al.[36] proposed a new representation for the 2-D PU problem and showed that LBP produces results that are superior to existing techniques. In 2007, Bioucas-Dias and Valadao[37] proposed a PU algorithm based on GC and validated that this approach, referred to as phase unwrapping max-flow/min-cut (PUMA), provides good results. In a similar way, in 2009, Ferraioli et al.[38] extended the GC unwrapping algorithm for multichannel SAR interferograms. This algorithm is meant to directly reconstruct target height $h$ from the interferometric phase series through a GC approach similar to the PUMA algorithm.

Furthermore, some attention has also been paid to the comparison of various optimization algorithms. Among them is Ref. 39, which conducted a comparative study on identical MRF models on a set of energy minimization benchmarks, such as stereo, image stitching, interactive segmentation, and denoising. What is more,[39] it provided a unifying software framework that facilitated a fair comparison of optimization techniques. Unfortunately, relatively little attention has been paid to the comparison of various optimization algorithms for PU. This raises the question of how they compare with each other in terms of the identical MRF model for PU. In this paper, we show a comparison of the ICM, GC, LBP, and TRW-S algorithms on the identical MRF model for PU, with the hope to give researchers some guidance as to which optimization method is best suited for solving the PU problem. We show that an MRF model can be used to solve a PU problem and examine the four optimization algorithms from different aspects using simulated data, which depends on their capability against different phase surfaces, noise levels, and clique potentials. Moreover, we also use real data to compare the solution quality and runtime of the four algorithms on real applications. This paper is organized as follows: Sec. 2 explains the formulation of the identical MRF for PU used in our tests. Section 3 describes the implementations of the four optimization algorithms on the same MRF model for PU in our tests. Section 4 provides simulated and real-data experiments of the four optimization algorithms in which some detailed analysis and comparison are provided. Finally, Sec. 5 concludes this paper.

## 2 Markov Random Field Model for Phase Unwrapping

The wrapped phase $\varphi$ and the unwrapped (absolute) phase $\phi$ are related by multiples of $2\pi$. Formally, we have

$$\phi = \varphi + 2k\pi, \tag{1}$$

where $k$ is the wrap count. The PU problem determines the unknown wrap count $k$ for each pixel. Dias and Leitão[40] suggested a general MRF model for PU, which is defined in terms of an energy function under a Bayesian perspective

$$E(k|\varphi) = \sum_{p\in \mathrm{P}, q\in \mathrm{N}_p} V(\Delta\phi_{pq}), \tag{2}$$

where $P$ is the set of pixels in a phase image and $N$ is the neighborhood system in the four-connected image grid graph. In the computation, the value of $k$ can be the binary variables of 1 or 0, which means that every pixel's label either increases by one (phase plus $2\pi$) or remains unchanged. We believe that the value of $k$ can also be negative in MRF-based PU algorithms, i.e., $k \in \{0, 1, -1\}$, which is similar to the traditional PU algorithms, e.g., path following algorithms, in which the PU result can be uniquely achieved by a simple integration process, i.e., addition of a $2\pi$ or subtraction of a $2\pi$. Under this condition, the energy minimization is achieved by multilabel optimization. The reason why a binary-label is chosen is that binary optimization is advantageous over multilabel optimization because the global minimum can only be reached for binary optimization, whereas there is no approach that can reach a global minimum for multilabel optimization. We also compared the performance of binary-label and multilabel optimization for PU problem, which illustrates that binary-label optimization performs much better than multilabel optimization when facing rapid-phase variations or phase discontinuities. However this issue is out of the scope of this paper. Moreover, we believe that it is not fair to pick only one MRF model to compare the four optimization algorithms for PU. It is worth mentioning that changes of the MRF model for PU depend on choosing different clique potentials $V(\cdot)$, a real-valued function, which defines how the phase of the neighboring pixels in the clique interacts.[29,37] Notice that the topography phase images usually contain two components: a spatially smooth component where the topography is comparatively flat and a nonsmooth component where the local topography varies drastically. Hence, a wide range of potential functions has been studied in the PU literature, which can effectively handle the type of mixed image characteristics, i.e., both smooth and nonsmooth features. The clique potential is usually defined by a generalized-$\rho$ Gaussian model, which is given as

$$V(\cdot) = (\Delta\phi_{pq})^{\rho}, \tag{3}$$

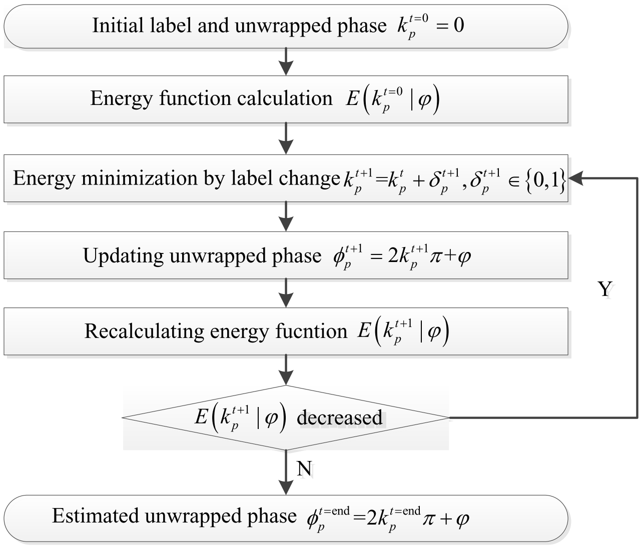

Initial label and unwrapped phase $k_p^{t=0}=0$
Energy function calculation $E\left(k_p^{t=0}\,|\,\varphi\right)$
Energy minimization by label change $k_p^{t+1}=k_p^t+\delta_p^{t+1},\delta_p^{t+1}\in\{0,1\}$
Updating unwrapped phase $\phi_p^{t+1}=2k_p^{t+1}\pi+\varphi$
Recalculating energy fucntion $E\left(k_p^{t+1}\,|\,\varphi\right)$
$E\left(k_p^{t+1}\,|\,\varphi\right)$ decreased
Y
N
Estimated unwrapped phase $\phi_p^{t=\text{end}}=2k_p^{t=\text{end}}\pi+\varphi$

**Fig. 1** Flowchart of the MRF-based PU approach.

where $\Delta\phi_{pq}$ measures the absolute phase difference of assigning labels $k_p$ and $k_q$ to two neighboring pixels given as

$$\Delta\phi_{pq}=[2\pi(k_p-k_q)+\Delta\varphi_{pq}], \tag{4}$$

$$\Delta\varphi_{pq}=\varphi_p-\varphi_q, \tag{5}$$

and $\rho$ is potential exponent, which decides the employed error criterion. When $\rho=2$, we have a quadratic model, which produces smooth images with very low probability of sharp transitions in intensity. When $\rho=1$, we have a linear model, which performs better than quadratic model in what discontinuity preserving is concerned. Such clique potentials, when $\rho\geq 1$, are named as convex clique potentials, which embrace the classical minimum $L^\rho$ norm PU problem. To allow for even sharper intensity transitions, nonconvex potential functions, when $0<\rho<1$, is employed. Hence, we will evaluate the performance of the four optimization algorithms using different clique potentials in Sec. 4.3.

This energy minimization for PU is achieved by a finite sequence of binary minimizations, which can be solved by different optimization algorithms, such as ICM, GC, LBP, and TRW-S algorithms, as is explained in Sec. 3. The overall process of MRF-based PU approach is shown in Fig. 1. Initially, the labels of all pixels are set to zero, i.e., $k_p^{t=0}=0$. At each iteration step, every pixel's label either be 1 (phase plus $2\pi$) or 0 (phase remains unchanged), i.e., $k_p^{t+1}=k_p^t+\delta_p^{t+1}$, in which the $t$ denotes iteration and $\delta_p^{t+1}\in\{0,1\}$. Every iteration aims to decrease the value of the energy function [Eq. (2)]. After each iteration, the unwrapped phase is updated, i.e., $\phi_p^{t+1}=2k_p^{t+1}\pi+\varphi$ and the energy function Eq. (2) is recalculated. When the energy ceases to decrease, the iteration is terminated, where the unwrapped phase is estimated, i.e., $\phi_p^{t=\text{end}}=2k_p^{t=\text{end}}\pi+\varphi$.

## 3 Optimization Algorithms for Phase Unwrapping

In this section, we describe the four optimization algorithms that we have implemented for MRF model to solve PU problem. Most of the algorithms were implemented by the software interface available at: http://vision.middlebury.edu/MRF/ for energy minimization; the exceptions is the GC algorithm, for which we refer to an approximate algorithm as a minor modification of standard GC algorithm to handle nonconvex clique potentials proposed in Ref. 37.

### 3.1 *Iterated Conditional Modes*

The ICM algorithm proposed by Ref. 26 is an iterative algorithm that modifies at each step the labels of each site by keeping the other fixed, but differently from SA and others stochastic algorithm, the new value is chosen in a deterministic way. The algorithm goes through all the pixels $p$, it starts with an estimate of the labeling $k_p = 0$, and calculates the energy function for each possible value of $k_p \in \{0, 1\}$

$$d_p(k_p) = \sum_{p \in \mathrm{P}, q \in \mathrm{N}_p} V(\Delta\phi_{pq}), \tag{6}$$

and then compares the label assigned to the pixel $p$ at the $t$'th iteration using the value that yields the largest decrease in the energy function. Hence, given the wrapped phase $\varphi$ and the labels, the algorithm sequentially updates each $k_p^t$ into $k_p^{t+1}$ by minimizing $E(k_p|\varphi)$ with respect to $k_p$. The process ends when the number of updates between two consecutive iterations becomes too small. This algorithm is very fast and is not computationally heavy as it just needs to evaluate the energy function at each iteration. By the way, the effectiveness of this algorithm is strictly related to the initialization $k_p^{t=0}$ provided to the algorithm. Unfortunately, the results of this algorithm are extremely sensitive to the initial estimate, and can often be trapped in the local minimum nearest to the initial solution due to the huge number of local minima (Algorithm 1).

### 3.2 *Graph Cuts*

We used the GC algorithm provided in Ref. 37. By mapping binary optimizations onto graph max-flows, the energy function [Eq. (2)] is rewritten as

$$E(k_p^t + \delta_p^{t+1}|\varphi) = \sum_{p \in \mathrm{P}, q \in \mathrm{N}_p} V[2\pi(\delta_p^{t+1} - \delta_q^{t+1}) + \Delta\phi_{pq}^t]. \tag{7}$$

**Algorithm 1** Iterated conditional mode.

```
Initialization $k^0 = 0$, possible_improvement = 1
while possible_improvement do
for all pixel $p$ do
  if $d_p(1) < d_p(0)$ then
      $k_p^{t+1} = k_p^t + 1$ (phase plus $2\pi$)
    else
      $k_p^{t+1} = k_p^t$ (remain unchanged)
  endif
endfor
 if $E(k^{t+1}|\varphi) < E(k^t|\varphi)$ then
   $\phi^{t+1} = \varphi + 2k^{t+1}\pi$
 else
   possible_improvement = 0
 endif
 endwhile
```

For the sake of simplicity, Eq. (7) is renamed as

$$E(k_p^t + \delta_p^{t+1}|\varphi) = \sum_{p\in \mathrm{P}, q\in \mathrm{N}_p} E_p(\delta_p^{t+1}, \delta_q^{t+1}). \tag{8}$$

The minimization of Eq. (8) can be achieved through a cut on a weighted graph $\sigma = \langle \nu, e \rangle$ ($\nu$ and $e$ denote the set of vertices and edges, respectively) with two terminals $s$ and $t$, if and only if it satisfy the regularity condition

$$E_p(0,0) + E_p(1,1) \le E_p(1,0) + E_p(0,1). \tag{9}$$

An $s - t$ cut is a set of edges such that the terminals are separated into two disjoint sets $s \in S$ (phase plus $2\pi$) and $t \in T$ (phase remains unchanged). The cost of the cut equals the sum of its edge weights between $S$ and $T$. According to Eq. (8), we have

$$\begin{aligned} E(0,0) &= V(\varphi_p - \varphi_q) \\ E(1,1) &= V(\varphi_p - \varphi_q) \\ E(0,1) &= V(\varphi_p - \varphi_q - 2\pi) \\ E(1,0) &= V(\varphi_p - \varphi_q + 2\pi). \end{aligned} \tag{10}$$

A directed edge $(p, q)$ is assigned the weight of $E(0,1) + E(1,0) - E(0,0) - E(1,1)$. Moreover, for vertex $p$, if $E(1,0) - E(0,0) > 0$, an edge $(s, p)$ is assigned the weight of, otherwise, an edge $(p, t)$ is assigned the weight of $E(0,0) - E(1,0)$. Similarly, for vertex $q$, if $E(1,1) - E(1,0) > 0$, an edge $(s, q)$ is assigned the weight of $E(1,1) - E(1,0)$, otherwise, an edge $(q, t)$ is assigned the weight of $E(1,0) - E(1,1)$. Elementary graphs are constructed and merged to obtain a main graph, shown in Fig. 2. Once energy is mapped on to the graph, energy is easily minimized using the min-cut/max-flow on the constructed graph. Among the available max-flow algorithms, we have used the augmenting path algorithms proposed in Ref. 41 for finding the min-cut/max-flow. For nonconvex potential, the GC algorithm of convex potential does not fit because it faces with the problem that the condition of regularity [Eq. (9)] does not hold for every pairwise clique interaction. In other words, it is not possible to map an energy function onto a graph, and energy cannot be minimized via standard GC algorithm. This problem can be resolved by applying majorize minimize (MM) concepts to energy function. That is wherever the pixel pair does not satisfy the regularity condition [Eq. (9)] then the edge weight between pixel pairs is set to zero to satisfy the regularity condition. The main computational cost of GC algorithm lies in computing the min-cut, which is done via max-flow. Thus, the computational complexity of GC algorithm for PU is $O(n^2 m)$ for a given $T$, where $n$ and $m$ are the number of vertices and edges, respectively, and $T$ is the number of iterations (Algorithm 2).

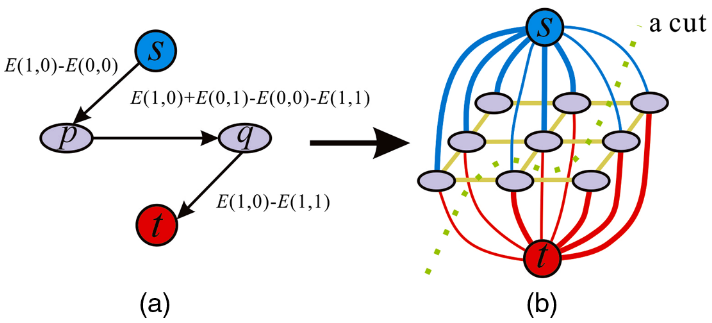


**Fig. 2** (a) Elementary graphs are constructed and (b) merged to obtain a main graph, where *s* and *t* represent terminals, and *p* and *q* represent the two neighboring pixels.

**Algorithm 2** Graph cuts.

```
Initialization $k^0 = 0$, possible_improvement = 1
while possible_improvement do
    Compute $E(0,0)$, $E(1,1)$, $E(0,1)$, and $E(1,0)$ for every pixel pair
    Find nonregular pixel pairs $[E(0,1) + E(1,0) - E(0,0) - E(1,0) < 0]$. If there is any,
    regularize it using the MM method (for instance, set the edge weight to zero).
    Construct elementary graphs and merge them to obtain the main graph.
    Compute the max-flow/min-cut $(S, T)$
    for all pixel $p$ do
        if pixel $p \in S$ then
            $k_p^{t+1} = k_p^t + 1$ (phase plus $2\pi$)
        else
            $k_p^{t+1} = k_p^t$ (remain unchanged)
        endif
    endfor
    if $E(k^{t+1}|\varphi) < E(k^t|\varphi)$ then
        $\phi^{t+1} = \varphi + 2k^{t+1}\pi$
    else
        possible_improvement = 0
    endif
endwhile
```

### 3.3 *Max-Product Loopy Belief Propagation*

The LBP algorithm provided by Ref. 42 is a message-passing algorithm that passes messages around the graph defined by a four-connected image grid. The method is iterative, with messages from all nodes being passed in parallel. Each message is a vector of labels $k \in \{0,1\}$. Let $m_{p\to q}^t$ be the message that node $p$ sends to a neighboring node $q$ at iteration $t$. It starts with all entries in $m_{p\to q}^t$ initialized to zero. At each iteration, new messages are computed as follows:

$$m_{p\to q}^t(k_p) = \min_{k_p}\left[V(\Delta\phi_{pq}) + \sum_{s\in N(p)\backslash q} m_{s\to p}^{t-1}(k_p)\right], \tag{11}$$

where $N(p) \setminus$ q denotes the neighbors of $p$ other than $q$. Messages are passed along rows and then along columns. When a row or column is processed, the algorithm starts at the first node and passes messages in one direction. Once the algorithm reaches the end of a row or column, messages are passed backward along the same row or column, and then a belief vector is computed for each node

$$b_p(k_p) = \sum_{q\in N(p)} m_{q\to p}^t(k_p). \tag{12}$$

Finally, the label $k_p$ that minimizes $b_p(k_p)$ individually at each node is selected. Iterations terminate when energy no longer decreases. The major advantage of the LBP algorithm lies in their ability to deal with nonconvex potential and implementation in a simple fashion. However,

**Algorithm 3** Loopy belief propagation.

```
Initialization $k^0 = 0$, $m^0_{p\to q} = 0$, possible_improvement = 1
while possible_improvement do
  Update the messages $m^t_{p\to q}$ and pass them along rows and then along columns. Once the algorithm reaches the end of a row or column, messages are passed backward along the same row or column.
  for all pixel $p$ do
    if $b_p(1) < b_p(0)$ then
      $k_p^{t+1} = k_p^t + 1$ (phase plus $2\pi$)
    else
      $k_p^{t+1} = k_p^t$ (remain unchanged)
    endif
  endfor
  if $E(k^{t+1}|\varphi) < E(k^t|\varphi)$ then
    $\phi^{t+1} = \varphi + 2k^{t+1}\pi$
  else
    possible_improvement = 0
  endif
endwhile
```

it is not guaranteed to converge and may go into an infinite loop switching between two labeling. In other words, it has a strong local minimum property that is somewhat analogous to that of the ICM algorithm. The implementation of the message-passing algorithm on the grid graph all run in $O(n)$ time for a given $T$, where $n$ is the number of pixels in the image and $T$ is the number of iterations (Algorithm 3).

## 3.4 *Sequential Tree-Reweighted Message Passing*

The TRW-S algorithm proposed by Ref. 35 is another message-passing similar to the LBP algorithm. The message update rule is identical to Eq. (11) of the standard LBP. However, the most striking difference from the LBP algorithm is its strategy to pass messages on grids. In the TRW-S implementation, nodes are processed in a scan-line order with a forward and backward pass. In the forward pass, each node sends messages to its right and bottom neighbors. In the backward pass, messages are sent to the left and upper neighbors. Another difference involves labels computed from the messages. In LBP, each pixel chooses independently the label with the lowest belief $b_p(k_p)$, whereas in TRW-S, the labeling $k_p \in \{0, 1\}$ is computed from that minimizes

$$c_p(k_p) = \sum_{q\in \mathrm{N}_{\text{left,up}}(p)} V(\Delta\phi_{pq}) + \sum_{q\in \mathrm{N}_{\text{right,bottom}}(p)} m^t_{q\to p}(k_p). \tag{13}$$

It is noted that there is no guarantee that the energy of TRW-S implementation might actually decrease with time. In practice, the energy sometimes starts to oscillate. To deal with this issue, one could keep track of the lowest energy to date and return that state when the algorithm is terminated. The computational complexity of TRW-S algorithm is same as the LBP algorithm (Algorithm 4).

**Algorithm 4** Tree-reweighted message passing.

```
Initialization $k^0 = 0$, $m^0_{p\to q} = 0$, possible_improvement = 1
while possible_improvement do
  Update the messages $m^t_{p\to q}$ and pass them in scan-line order with a forward and backward pass.
  In the forward pass, each node sends messages to its right and bottom neighbors.
  In the backward pass, messages are sent to the left and upper neighbors.
  for all pixel $p$ do
    if $c_p(1) < c_p(0)$ then
      $k^{t+1}_p = k^t_p + 1$ (phase plus $2\pi$)
    else
      $k^{t+1}_p = k^t_p$ (remain unchanged)
    endif
  endfor
  if $E(k^{t+1}|\varphi) < E(k^t|\varphi)$ then
      $\phi^{t+1} = \varphi + 2k^{t+1}\pi$
  else
      possible_improvement = 0
  endif
endwhile
```

## 4 Performance Analysis

In this section, the performance of the four optimization algorithms is compared with different aspects on some simulated data, which depends on their capability against different phase surfaces, noise levels, and clique potentials. In addition to that, we test the performance of each algorithm on real applications using real data and compare their accuracy. We implemented the algorithms in C or C++ and ran every experiment presented here on the same computer (2.5 GHz and 8 Gbyte RAM).

### 4.1 *Comparison of Different Phase Surfaces*

To analyze the accuracy of the four optimization algorithms, simulated data of different phase surfaces are used in the experiment. We define the simulated data based on two types of phase surfaces (one is continuous phase surface and other is discontinuous phase surface); hence, we have simulated four datasets introduced in Refs. 37 and 40 and the phase images shown in Fig. 3: first column—Gaussian surface, size 256 × 256; second column—Peaks surface generated by MATLAB’s Peaks function, size 256 × 256; third column—discontinuous Gaussian surface with a quarter set to zero, size 256 × 256; and fourth column—discontinuous Gaussian surface with a nonvertical and nonhorizontal aligned sector set to zero, size 256 × 256.

In the first experiment, we test the four algorithms on two kinds of continuous phase surface. Figures 3(b) and 3(c) show two wrapped phase images (256 × 256 pixels); they are generated from original absolute phase surfaces formed by Gaussian elevations with heights of 45 or 70 rad, which are shown in Figs. 3(a) and 3(b). The two wrapped images are simulated according to an InSAR observation statistics,[37] producing an interferometric pair with coherence coefficient 1.0 (noise-free). Regarding the first image, Fig. 4 shows the corresponding unwrapped surface by the four algorithms using convex potential $V(x) = x^2$, which produces smooth images with very low probability of sharp transitions in intensity.[40] It can be noted that the four algorithms all

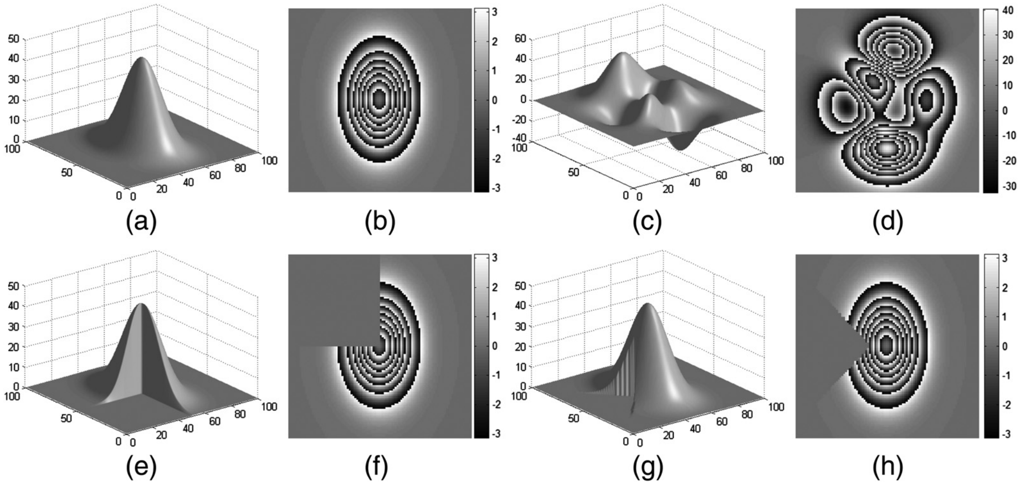


**Fig. 3** Test on the four simulated data with different phase surfaces. (a) Gaussian surface, size 256 × 256, (c) peaks surface, size 256 × 256, (e) discontinuous surface with a quarter set to zero, size 256 × 256, and (g) discontinuous surface with a nonvertical and nonhorizontal aligned sector set to zero, size 256 × 256. (b), (d), (f), and (h) The wrapped phase images corresponding to (a), (c), (e), and (g), respectively.

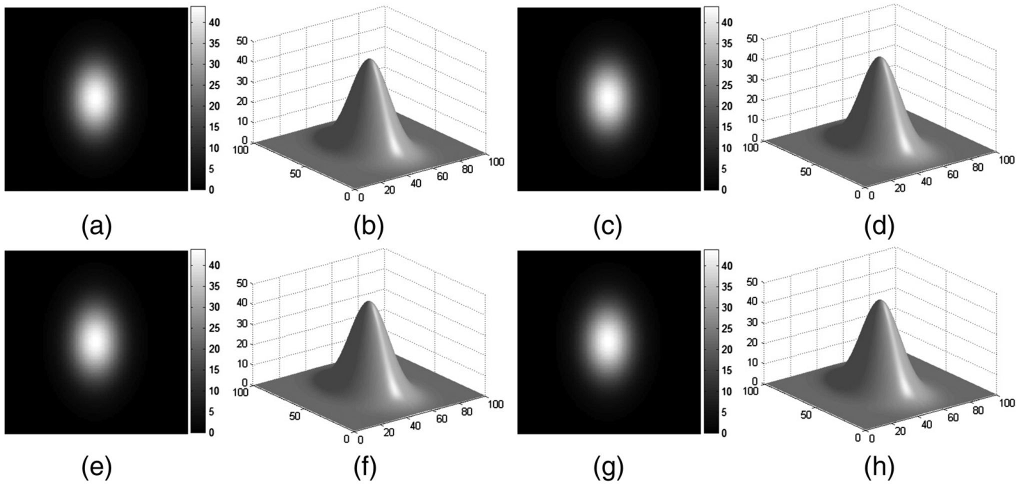


**Fig. 4** Unwrapped Gaussian phase surfaces generated by the (a) ICM algorithm, (c) GC algorithm, (e) LBP algorithm, and (g) TRW-S algorithm. (b), (d), (f), and (h) The mesh corresponding to (a), (c), (e), and (g), respectively.

successfully accomplish correct unwrapping. To quantitively analyze the accuracy of each algorithm, the root mean square (rms) errors are listed in Table 1. It can be noticed that the four algorithms have the same rms error value 0.0 (error free), meaning that the four algorithms can generate flawless results for noise-free inputs.

Regarding the second wrapped image, Fig. 3(d), although the coherence value is at the maximum (there is no noise), the phase fringes vary drastically and generate a lot of residues that degrade the accuracy of the unwrapping. Figure 5 shows the corresponding unwrapped surfaces obtained by the four algorithms using again convex potential $V(x) = x^2$. In Figs. 5(c), 5(e), and 5(g), we can see that the GC, LBP, and TRW-S algorithms all unwrap the phase image flawlessly. The exception is ICM algorithm, which unwraps several regions incorrectly as shown in Fig. 5(a). As indicated by the metrics in Table 1, the GC, LBP, and TRW-S algorithms have identical rms error value 0.0 (no error), whereas the ICM algorithm has higher rms error, which means that the ICM algorithms show a poor performance regarding the unwrapping of phase surface where the local topography varies drastically. This is because the ICM algorithm can

**Table 1** Rms errors (rad) of the four algorithms on the simulated data of the different phase surface types.

| Algorithms | Gaussian surface | Peaks surface | Discontinuous surface with a quarter set to zero | Discontinuous surface with both sectors set to zero |
|---|---|---|---|---|
| ICM | 0.00 | 4.11 | 8.71 | 5.76 |
| GC | 0.00 | 0.00 | 0.00 | 0.33 |
| LBP | 0.00 | 0.79 | 7.9 | 5.65 |
| TRW-S | 0.00 | 0.01 | 0.93 | 0.73 |

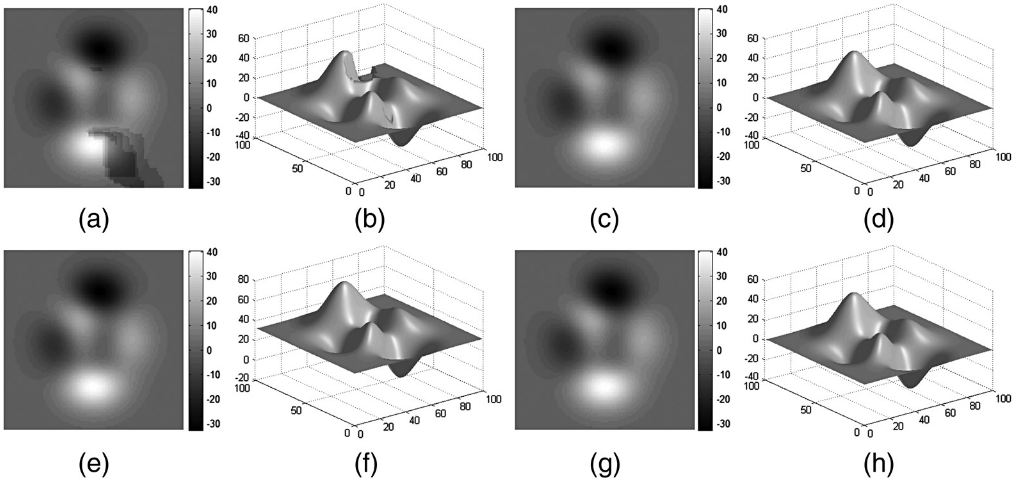


**Fig. 5** Unwrapped peaks phase surfaces generated by the (a) ICM algorithm, (c) GC algorithm, (e) LBP algorithm, and (g) TRW-S algorithm. (b), (d), (f), and (h) The mesh corresponding to (a), (c), (e), and (g), respectively.

often be trapped in the local minimum nearest to the initial solution due to the rapid-phase variations.

In the second experiment, we examine the four algorithms on Gaussian surface with discontinuities. Figures 3(e) and 3(g) are both analogous to Fig. 3(a), but now the original Gaussian surface is changed with a quarter set to zero and with both sectors set to zero, respectively. Therefore, this null quarter causes many discontinuities, which renders a very difficult PU problem. Notice that we do not utilize any discontinuity information from any quality map to the four optimization algorithms in this experiment. We employ nonconvex clique potential $V(x) = x^{0.5}$, which is desirable to allow discontinuity preservation.[37] Figures 6 and 7 show the two kinds of unwrapped surfaces produced by the four algorithms, respectively. It can be illustrated visually that the ICM and LBP algorithms both fail completely. On the contrary, the GC and TRW-S algorithms appear to unwrap the two discontinues image correctly. The same behavior can be noticed in Table 1, where the GC and TRW-S algorithms have lower error than the other two, which means that the GC and TRW-S algorithms suffer less from the discontinuity effects, whereas the ICM and LBP algorithms are both ineffective on solving discontinuities issue. The reason may be that the ICM and LBP algorithms are both easily trapped in the local minimum due to the phase discontinuities.

## 4.2 *Comparison of Energy and Time*

Considering that a logical way to compare optimization algorithms is in terms of energy and running time, we produce plots that record energy decrease versus time for every algorithm

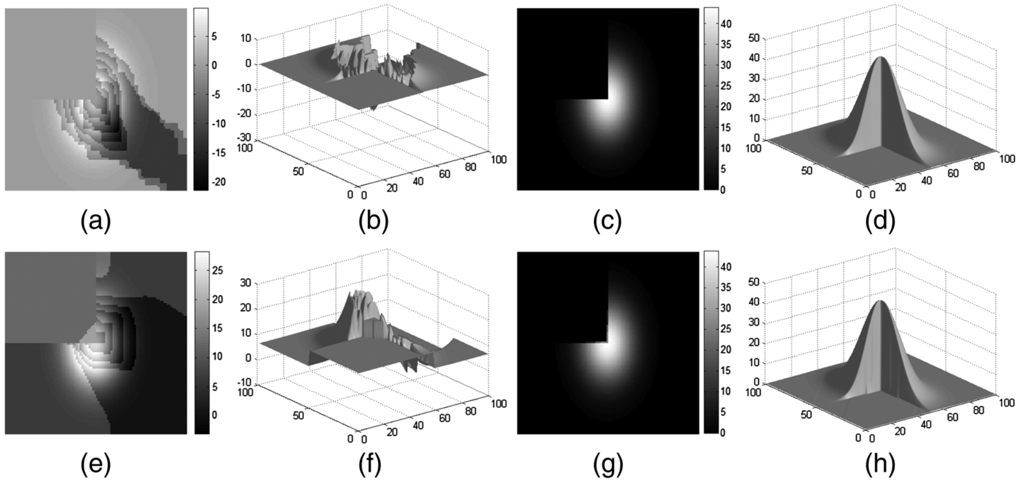


**Fig. 6** Unwrapped Gaussian phase surfaces with a quarter set to zero generated by the (a) ICM algorithm, (c) GC algorithm, (e) LBP algorithm, and (g) TRW-S algorithm. (b), (d), (f), and (h) The mesh corresponding to (a), (c), (e), and (g), respectively.

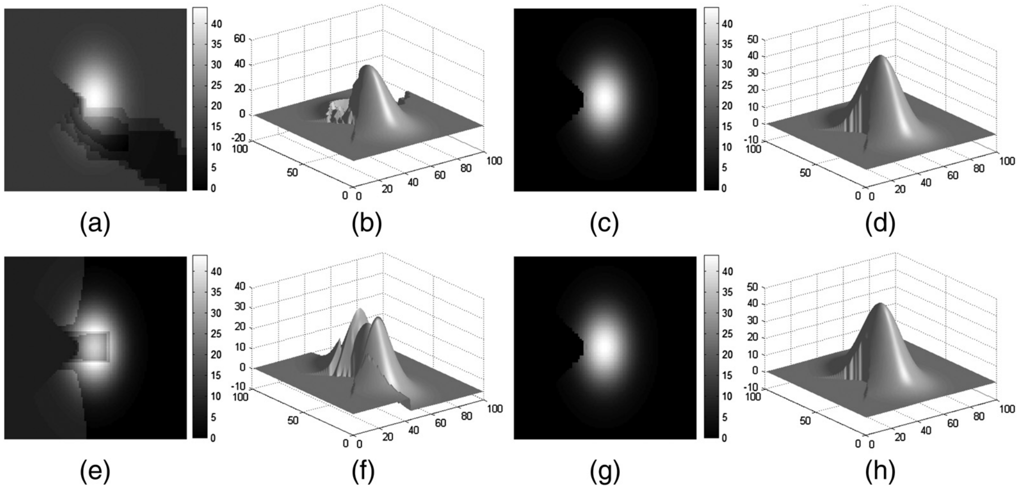
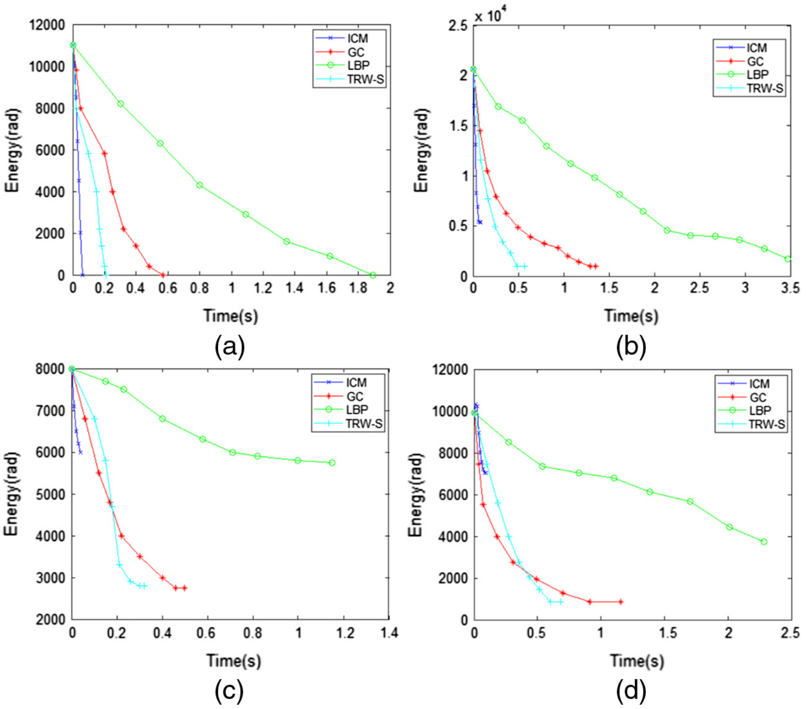


**Fig. 7** Unwrapped Gaussian phase surfaces with both sectors set to zero generated by the (a) ICM algorithm, (c) GC algorithm, (e) LBP algorithm, and (g) TRW-S algorithm. (b), (d), (f), and (h) are the mesh corresponding to (a), (c), (e), and (g), respectively.

tested. It is particularly clear that the best methods produce energies that are extremely close to the global minimum. The processes of global energy minimization for the four optimization algorithms tested on four data sets are shown in Fig. 8. The *x*-axis of the plot shows the running time, and the *y*-axis shows the energy of the different methods over time. To quantitively analyze the computational performance of each algorithm, the running time (s) is show in Table 2. Regarding the Gaussian surface, Fig. 8(a), it can be noticed that the four algorithms all reach the global minimum, which means that they can generate reasonable results. Among them, the ICM algorithm needs the least time to converge, followed by the TRW-S and GC algorithms, and the LBP algorithm is the slowest. Concerning the peaks surface, Fig. 8(b), it can be observed that the TRW-S algorithm reaches the global minimum quickly and close to its GC algorithm. The LBP algorithm is slower and performs slightly worse than the former two algorithms. On the other hand, the ICM algorithm is not guaranteed to converge, meaning that the algorithm fails completely. With respect to the two Gaussian surfaces with discontinuities, Figs. 8(c) and 8(d), we notice that although slower, the GC algorithm is guaranteed to compute the global minimum, and very close to its TRW-S algorithm, which comes extremely close but never actually attains



(a)

(b)

(c)

(d)

**Fig. 8** Energy decreasing for the unwrapping of image in (a) Gaussian surface, (b) Peaks surface, (c) discontinuous Gaussian surface with a quarter set to zero, and (d) discontinuous Gaussian surface with both sectors set to zero.

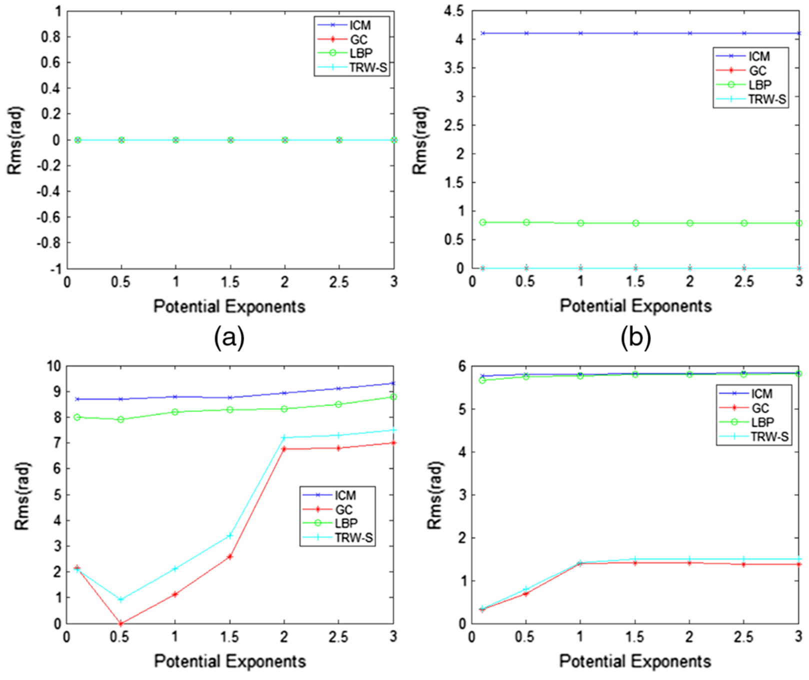
**Table 2** Running time (s) of the four algorithms on the simulated data of the different phase surface types.

| Algorithms | Gaussian surface | Peaks surface | Discontinuous surface with a quarter set to zero | Discontinuous surface with both sectors set to zero |
|---|---|---|---|---|
| ICM | 0.08 | 0.15 | 0.07 | 0.21 |
| GC | 0.58 | 1.38 | 0.49 | 1.18 |
| LBP | 1.88 | 3.5 | 1.18 | 2.31 |
| TRW-S | 0.21 | 0.58 | 0.32 | 0.68 |

the global minimum. On the contrary, the LBP algorithm does much worse than the other two solutions and the ICM does even worse. From Table 2, it is worth mentioning that the GC algorithm is more time consuming than the TRW-S algorithm, presumably due to its high computational cost of GC that lie in computing the minimum cut via max-flow algorithm.

## 4.3 *Comparison of Different Clique Potentials*

In this section, to evaluate what clique potentials should be taken in four algorithms on four datasets, we calculate the rms error against the clique potentials with different exponents (between 0.1 and 3). The relationship between the rms error and potential exponents is shown in Fig. 9. The behaviors of error curves in Figs. 9(a) and 9(b) where phase surface is continuous show that the rms errors of the four algorithms vary less against the variation of the potential exponents, which demonstrates that convex and nonconvex potentials both prove to be effective regarding continuous phase surfaces. Notice that convex potential has the advantage over nonconvex potential in the situation of continuous phase surface, both in



(a) (b)

(c) (d)

**Fig. 9** Relationship between the rms error (rad) of each algorithm and the potential exponents for the unwrapping of image in (a) Gaussian surface, (b) peaks surface, (c) discontinuous Gaussian surface with a quarter set to zero, and (d) discontinuous Gaussian surface with both sectors set to zero.

stability and computational efficiency. By contrast, it can be noticed in Figs. 9(c) and 9(d) where the phase surface is discontinuous, that GC and TRW-S algorithms with nonconvex potential (exponent is less than one) perform much better than that with convex potential (exponent is more than one), which means that the two algorithms with nonconvex potentials have more strong discontinuity preserving ability, whereas those with convex potentials tend to smooth phase discontinuities. On the contrary, the ICM and LBP algorithms are both useless regardless of whether employing convex or nonconvex potentials.

## 4.4 *Comparison of Different Coherence Coefficients*

To test the noise robustness of each algorithm deeply, we test the performances of the four algorithms against different coherence coefficients that are added with different coherence coefficients (1.0, 0.9, 0.8, 0.7, 0.6, and 0.5). Figure 10 is the relationship between the rms error (rad) of each algorithm and the coherence coefficient. Regarding the continuous phase surface, Figs. 10(a) and 10(b), it can be seen that the error curves of the three algorithms are similar in the majority of the coherence coefficient scale, as the exception is ICM algorithm, which suggests that the decorrelation effect on the former three algorithms is not evident in the continuous phase surface. Concerning the discontinuous phase surface, Figs. 10(c) and 10(d), the behavior of these curves show that the rms error of the GC and TRW-S algorithms varies less than that of the other two algorithms against the variation of the coherence coefficient, which demonstrates that the performance of the two algorithms suffers less with the degradation of the coherence coefficient. On the contrary, in the majority of the coherence coefficient scale, the ICM and LBP algorithms both generate higher rms errors, which mean that they suffer from poor noise robustness. The reason is that the ICM algorithm, as mentioned above, is often be trapped in the local minimum nearest to the initial solution due to the high phase noise, whereas the LBP algorithm is easy to enter into an infinite loop switching between two labelings, which implies that it has strong local minimum properties in the high-noise area.

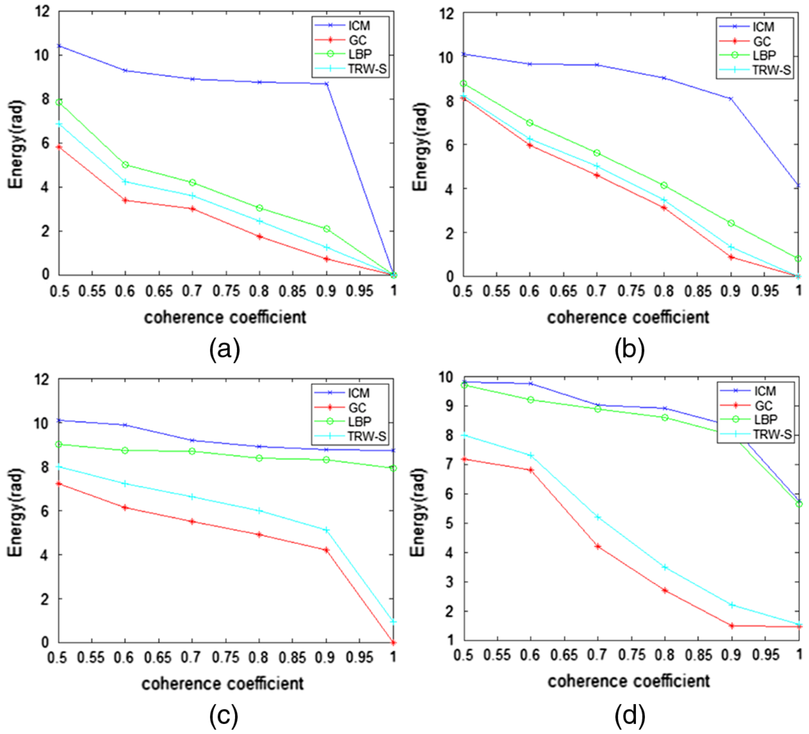


**Fig. 10** Relationship between the rms error (rad) of each algorithm and the coherence coefficient for the unwrapping of image in (a) Gaussian surface, (b) peaks surface, (c) discontinuous Gaussian surface with a quarter set to zero, and (d) discontinuous Gaussian surface with both sectors set to zero.

**Table 3** Major parameters of an interferometric pair of TerraSAR-X.

| Wavelength (m) | Resolution (m) | Polarization | Orbit | Incidence | Date |
|---|---|---|---|---|---|
| 0.031 | 3 | HH | Descending | 39.2 | March 10, 2008 |
|  |  |  |  |  | March 21, 2008 |

## 4.5 *Comparison of Real Data*

To test the performances of the four algorithms on real applications, real-data experiment is performed, which intends to reconstruct the real digital elevation model (DEM) using different algorithms and compare their accuracy. The SAR interferometric pair used in this experiment is acquired by two TerraSAR-X images covering the Grand Canyon, USA, and the data are downloaded from the Infoterra website.[43] The major parameters of the interferometric pair are listed in the Table 3. We use subsets of these two images to generate a flattened interferogram, Fig. 11(a), with a size of $709 \times 1136$ pixels and a coherence coefficient map, Fig. 11(b). Figure 11(c) shows the SAR intensity image of this data, where the topography varies drastically as it is located across the canyon, therefore, inducing many discontinuities and posing a very tough PU problem. For comparison, the ASTER G-DEM is simulated into the terrain phase, which is used as the reference shown in Fig. 11(d). Considering the original phase surface has many discontinuities, the nonconvex potential $V(x) = x^{0.5}$ is used, which has the discontinuity preserving ability.

Figure 12 shows the unwrapped results using the four algorithms. We observe that the four algorithms on real data have similar performance as those on simulated data. The ICM algorithm, Fig. 12(a), produces noticeable errors throughout the entire image. The LBP and TRW-S algorithms, Figs. 12(a) and 12(d), both remove most of the errors seen in the ICM result, yet some errors still appear in the bottom right corner due to an aliased wrapped image. The GC algorithm,

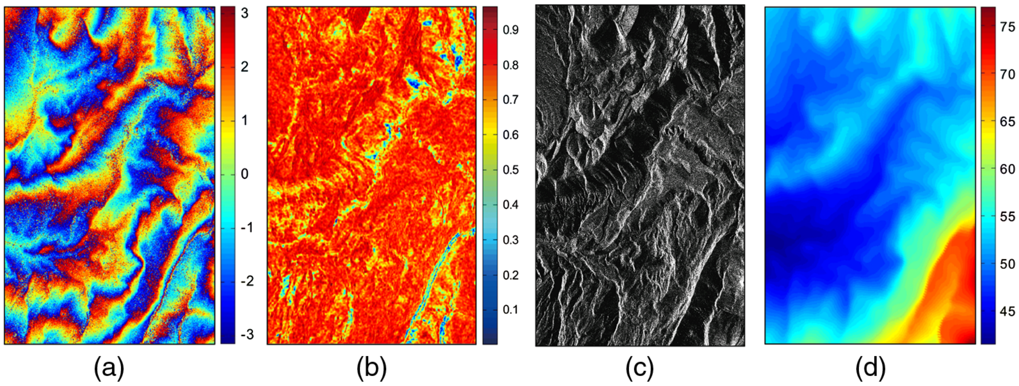


**Fig. 11** Test on the real data of two TerraSAR-X phase images with a size of 709 × 1136 pixels. (a) Flattened interferogram, (b) coherence coefficient map, (c) SAR intensity image, and (d) topography phase simulated by ASTER G DEM, which is used as the reference topography phase.

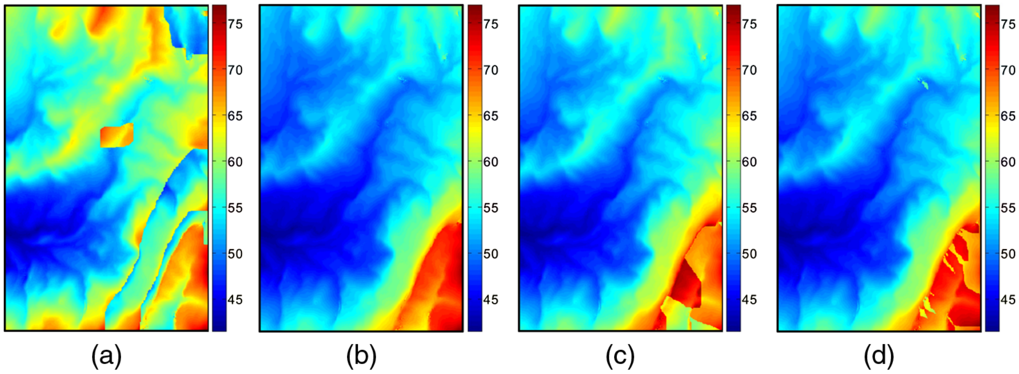


**Fig. 12** Unwrapped results on the real data using the (a) ICM algorithm, (b) GC algorithm, (c) LBP algorithm, and (d) TRW-S algorithm.

**Table 4** Running time and rms errors of the four algorithms on the real data.

| Algorithms | Running time (s) | Rms errors (rad) |
|---|---|---|
| ICM | 2 | 11.44 |
| GC | 177 | 1.4 |
| LBP | 447 | 3.47 |
| TRW-S | 149 | 2.3 |

Fig. 12(b), successfully alleviates the errors in the bottom right corner and generates seemingly preferable results similar to the reference topography phase, Fig. 11(d). The relative performance of the four algorithms can also be seen in the metrics in Table 4, where the GC algorithm has the lowest rms error, followed by the TRW-S and LBP algorithms, and the ICM algorithm generates the highest rms error. In addition, it can be observed that the ICM algorithm is faster than other three by a large margin and the GC algorithm is slightly slower than the TRW-S algorithm, whereas the LBP algorithm is very time-consuming.

## 5 Conclusions

We examined how the ICM, GC, LBP, and TRW-S algorithms compared with each other when applied to the same MRF model for PU, which gives researchers some guidance as to which optimization method is best suited for solving PU problem. First, we introduced an MRF model that can be used to solve a PU problem. Second, we described the four optimization algorithms that we have implemented for identical MRF model. Third, we gave some simulated-data experiments that depend on their capability against the variation of phase surfaces, noise levels, and clique potentials. In the end, we conducted real-data experiment in which some analysis and comparison are provided. According to the experimental results, we can, thus, conclude that:

1. We tested the performances of each algorithm against Gaussian, peaks and two discontinuous phase surfaces, respectively. We observed that the four algorithms all flawlessly unwrap the Gaussian surface. Moreover, we noticed that the LBP, GC, and TRW-S algorithms unwrap the peaks surface correctly, unlike the ICM algorithm. With respect to the discontinuous surface, we noted that the ICM and LBP algorithms are both ineffective, whereas the GC and TRW-S algorithms perform much better than other two algorithms.
2. We analyzed the minimum energy levels reached versus the running time characteristic of each algorithm for PU process. Among them, the ICM algorithm is very rapid, but it is not guaranteed to converge in peaks and two discontinuous surfaces. The GC algorithm converges more slowly but to a slightly lower energy than the solution found by the TRW-S algorithm in all cases. The LBP is very time consuming, and it never actually attains the global minimum in two discontinuous cases.
3. We examined the performances of each algorithm against convex and nonconvex clique potentials. It can be noted that the four algorithms with convex and nonconvex potentials prove to be effective to unwrap the continuous phase surfaces. Given this situation, convex potential has priory over nonconvex potential because of its stability and efficiency. It can also be noticed that the GC and TRW-S algorithms with nonconvex potentials have more strong discontinuity preserving ability, whereas the ICM and LBP algorithms are both useless regardless of whether using convex or nonconvex potentials.
4. We tested the noise robustness of the four algorithms against different coherence coefficient. Regarding the continuous phase surface, we noticed that the decorrelation effect on the GC, LBP, and TRW-S algorithms is not evident as the exception is ICM algorithm. With respect to the discontinuous surfaces, we observed that the GC and TRW-S algorithms suffer from less decorrelation effect compared with other two algorithms.
5. We examined the performance of the four algorithms on TerraSAR-X data to reconstruct the DEM of Grand Canyon, USA. We observed that the four algorithms on real data have a similar performance as those on simulated data, where the GC algorithm unwraps the phase image correctly, with the TRW-S algorithm resulting in a slightly larger errors. In contrast, the LBP algorithm performs worse compared with the former two and the ICM algorithm fails to unwrap the phase image completely.

We summarized evaluating the performances of the four algorithms as followers: the GC algorithm is clearly the winner among the four optimization methods in all cases but is slightly slower than the TRW-S algorithm, presumably due to the computational burden, which lies in computing the minimum cut via max-flow algorithm. However, the max-flow solution in GC algorithm has potential for parallelization, which is suitable for GPU acceleration.[44] The well-known older ICM algorithm, although is very rapid, performs much less accurately than the other three methods, especially in the terrain with the violent change or discontinuities. This is because that the ICM algorithm can often be trapped in the local minimum due to the rapid-phase variations or phase discontinuities. There is also a large gap between the performance of the two message-passing methods—the LBP and TRW-S algorithms. Although the LBP algorithm is a well regarded and widely used algorithm in vision applications, it performs surprisingly poorly on solving phase discontinuities issue. This is partly because it may go into an infinite loop switching between two labelings as the surface is discontinuous. The TRW-S algorithm, which has not been widely used in vision, performs much faster while closely achieving the performance of the GC method, and the reason may be the strategy to pass messages on grids differed from the LBP algorithm. What is more, it is noted that the ICM and two message-passing algorithms can also be easily parallelized, making them much more efficient.

It is worth mentioning that we only conduct a comparative study of the four algorithms on four-connected MRF model, it would be interesting to investigate different grid topologies such as an eight-connected topology, as well as nongrid topologies such as those used with Delaunay triangulation. Moreover, this MRF model for PU can be extended into a 3-D version to help unwrapping interferogram series in multibaseline InSAR. Thus, it is also worth comparing the four algorithms in multibaseline configuration. In addition to that, considering parallelization would be very appealing to take advantage of modern computer architectures, we expect that an evaluation of the parallelizability of optimization algorithms for PU will be given. In the future, we will focus on these aspects to conduct a comparative study of the four algorithms.

## Acknowledgments

This research was jointly supported by the National Natural Science Foundation of China (Grant No. 41501461), the Sponsorship of Jiangsu Overseas Research and Training Program for University Prominent Young and Middle-aged Teachers and Presidents, the Natural Science Foundation of Jiangsu Province of China (Grant No. BK20140419), and the open subject of Provincial Key Laboratory of Soochow University (Grant No. KJS1522). The authors would like to thank the anonymous reviewers for their constructive comments, which were taken into account in revising this paper.

## References

1. H. Zebker and R. Goldstein, "Topographic mapping from interferometric SAR observations," *J. Geophs. Res.* **91**, 4993–4996 (1986).
2. D. Massonnet and K. L. Feigl, "Radar interferometry and its application to changes in the Earth's surface," *Rev. Geophys.* **36**(4), 441–500 (1998).
3. R. Bürgmann, P. A. Rosen, and E. J. Fielding, "Synthetic aperture radar interferometry to measure Earth's surface topography and its deformation," *Annu. Rev. Earth Planet Sci.* **28**, 169–209 (2000).
4. D. C. Ghiglia and M. D. Pritt, *Two-Dimensional Phase Unwrapping: Theory, Algorithms and Software*, Wiley-Interscience, New York (1998).
5. R. M. Goldstein, H. A. Zebker, and C. L. Werner, "Satellite radar interferometry: two-dimensional phase unwrapping," *Radio Sci.* **23**(4), 713–720 (1988).
6. S. Madesn, H. Zebker, and J. Martin, "Topographic mapping using radar interferometry: processing techniques," *IEEE Trans. Geosci. Remote Sens.* **31**(1), 246–256 (1993).
7. W. Xu and I. Cumming, "A region-growing algorithm for InSAR phase unwrapping," *IEEE Trans. Geosci. Remote Sens.* **37**(1), 124–134 (1999).
8. H. Zhong et al., "A quality-guided and local minimum discontinuity based phase unwrapping algorithm for InSAR/InSAS interferograms," *IEEE Geosci. Remote Sens. Lett.* **11**(1), 215–219 (2014).
9. D. Gao and F. Yin, "Mask cut optimization in two-dimensional phase unwrapping," *IEEE Geosci. Remote Sens. Lett.* **9**(3), 338–342 (2012).
10. T. Flynn, "Two-dimensional phase unwrapping with minimum weighted discontinuity," *J. Opt. Soc. Am. A* **14**(10), 2692–2701 (1997).
11. J. Xu et al., "An efficient minimum-discontinuity phase-unwrapping method," *IEEE Geosci. Remote Sens. Lett.* **13**(5), 666–670 (2016).
12. D. Ghiglia and L. Romero, "Direct phase estimation from phase differences using fast elliptic partial differential equation solvers," *Opt. Lett.* **14**(20), 1107–1109 (1989).
13. D. Ghiglia and L. Romero, "Robust two-dimensional weighted and unweighted phase unwrapping that uses fast transforms and iterative methods," *J. Opt. Soc. Am. A* **11**(1), 107–117 (1994).
14. M. Costantini, "A novel phase unwrapping method based on network programming," *IEEE Trans. Geosci. Remote Sens.* **36**(3), 813–821 (1998).
15. C. Chen, "Statistical-cost network-flow approaches to two-dimensional phase unwrapping for radar interferometry," PhD Dissertation, Stanford University, Stanford, Connecticut (2001).
16. C. W. Chen and H. A. Zebker, "Network approaches to two-dimensional phase unwrapping: intractability and two new algorithms," *J. Opt. Soc. Am. A* **17**(3), 401–414 (2000).

17. A. Ferretti, C. Prati, and F. Rocca, "Permanent scatterers in SAR interferometry," *IEEE Trans. Geosci. Remote Sens.* **39**(1), 8–20 (2001).
18. P. Berardino et al., "A new algorithm for surface deformation monitoring based on small baseline differential SAR interferograms," *IEEE Trans. Geosci. Remote Sens.* **40**(11), 2375–2383 (2002).
19. B. M. Kampes and R. F. Hanssen, "Ambiguity resolution for permanent scatterer interferometry," *IEEE Trans. Geosci. Remote Sens.* **42**(11), 2446–2453 (2004).
20. A. Hooper and H. A. Zebker, "Phase unwrapping in three dimensions with application to InSAR time series," *J. Opt. Soc. Am. A* **24**(9), 2737–2747 (2007).
21. A. Pepe et al., "New advances of the extended minimum cost flow phase unwrapping algorithm for SBAS-DInSAR analysis at full spatial resolution," *IEEE Trans. Geosci. Remote Sens.* **49**(10), 4062–4079 (2011).
22. M. Costantini, F. Malvarosa, and F. Minati, "A general formulation for redundant integration of finite differences and phase unwrapping on a sparse multidimensional domain," *IEEE Trans. Geosci. Remote Sens.* **50**(3), 758–768 (2012).
23. G. Nico, G. Palubinskas, and M. Datcu, "Bayesian approach to phase unwrapping: theoretical study," *IEEE Trans. Signal Process.* **48**(9), 2545–2556 (2000).
24. J. Besag, "Spatial interaction and statistical analysis of lattice systems," *J. R. Stat. Soc. Ser. B* **36**(2), 192–236 (1974).
25. X. Descombes, F. Kruggel, and D. Cramon, "Spatio-temporal fMRI analysis using MRF's," *IEEE Trans. Med. Imaging* **17**(6), 1028–1039 (1998).
26. J. Besag, "On the statistical analysis of dirty pictures (with discussion)," *J. R. Stat. Soc. Ser. B* **48**(3), 259–302 (1986).
27. S. Barnard, "Stochastic stereo matching over scale," *Int. J. Comput. Vision* **3**(1), 17–32 (1989).
28. G. Ferraiuolo, V. Pascazio, and G. Schirinzi, "Maximum a posteriori estimation of height profiles in InSAR imaging," *IEEE Geosci. Remote Sensing Lett.* **1**(2), 66–70 (2004).
29. L. Ying et al., "Unwrapping of MR phase images using a Markov random field model," *IEEE Trans. Med. Imaging* **25**(1), 128–136 (2006).
30. R. Chen et al., "Integrated denoising and unwrapping of InSAR phase based on Markov random fields," *IEEE Trans. Geosci. Remote Sens.* **51**(8), 4473–4485 (2013).
31. Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Trans. Pattern Anal. Mach. Intell.* **23**(11), 1222–1239 (2001).
32. V. Kolmogorov and R. Zabih, "What energy functions can be minimized via graph cuts," *IEEE Trans. Pattern Anal. Mach. Intell.* **26**(2), 147–159 (2004).
33. J. Yedidia, W. Freeman, and Y. Weiss, "Generalized belief propagation," in *Advances in Neural Information Processing Systems*, pp. 689–695 (2000).
34. M. Wainwright, T. Jaakkola, and A. Willsky, "Tree consistency and bounds on the performance of the max-product algorithm and its generalizations," *Stat. Comput.* **14**, 143–166 (2004).
35. V. Kolmogorov, "Convergent tree-reweighted message passing for energy minimization," *IEEE Trans. Pattern Anal. Mach. Intell.* **28**(10), 1568–1583 (2006).
36. J. Frey, R. Koetter, and N. Petrovic, "Very loopy belief propagation for unwrapping phase images," in *Advances in Neural Information Processing Systems*, pp. 737–743 (2001).
37. J. Bioucas-Dias and D. Valadao, "Phase unwrapping via graph cuts," *IEEE Trans. Image Process.* **16**(3), 698–709 (2007).
38. G. Ferraioli et al., "Multichannel phase unwrapping with graph cuts," *IEEE Geosci. Remote Sens. Lett.* **6**(3), 562–566 (2009).
39. R. Szeliski et al., "A comparative study of energy minimization methods for Markov random fields with smoothness-based priors," *IEEE Trans. Pattern Anal. Mach. Intell.* **30**(6), 1068–1080 (2008).
40. J. Dias and J. Leitão, "The ZM algorithm for interferometric image reconstruction in SAR/SAS," *IEEE Trans. Image Process.* **11**(4), 408–422 (2002).
41. Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," *IEEE Trans. Pattern Anal. Mach. Intell.* **26**(9), 1124–1137 (2004).

42. P. Felzenszwalb and D. Huttenlocher, “Efficient belief propagation for early vision,” *Int. J. Comput. Vision* **70**(1), 41–54 (2006).
43. Airbus, “Sample imagery,” http://www.intelligence-airbusds.com/en/23-sample-imagery (8 June 2018).
44. V. Vineet and P. J. Narayanan, “CUDA cuts: fast graph cuts on the GPU,” in *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition Workshops*, pp. 1–8 (2008).

**Lifan Zhou** received his BS degree in geographic information system from Wuhan University in 2006, and his PhD in geographic information system from Zhejiang University in 2014. He is a lecturer at the Changshu Institute of Technology. He is the author of more than 10 journal papers. His main research interests are in the fields of phase unwrapping, and synthetic aperture radar interferometry signal processing and application.

**Dengfeng Chai** is an associate professor at the Zhejiang University. He received his BS degree in surveying engineering from Wuhan University, his MS degree in photogrammetry and remote sensing from Wuhan University, and his PhD in applied mathematics from Zhejiang University, in 1997, 2000, and 2006, respectively. He is the author of more than 20 journal papers. His research interests include developing statistical approaches for object recognition and extraction from remotely sensed images.

**Yu Xia** is an associate professor at the Changshu Institute of Technology. He received his MS degree in computer science and his PhD in computer vision both from Jiangnan University in 2009 and 2013, respectively. He is the author of more than 20 journal papers. His research interests include computer vision, digital image processing, and pattern recognition.

**Peifeng Ma** is a postdoctor at the Chinese University of Hong Kong. He received his MS degree in signal and information processing from Chinese Academy of Sciences in 2012, and his PhD in earth system and geoinformation science from Chinese University of Hong Kong in 2016. He is the author of more than 10 journal papers. His research interests include SAR tomography and persistent scatterer interferometry and their application to infrastructural health monitoring.