

# Phase unwrapping in three dimensions with application to InSAR time series

Andrew Hooper<sup>1,2,\*</sup> and Howard A. Zebker<sup>1</sup>

<sup>1</sup>Department of Geophysics, Stanford University, Mitchell Building, Stanford, California 94305, USA

<sup>2</sup>Current address: Nordic Volcanological Center, Institute of Earth Sciences, University of Iceland, Sturlugata 7, Reykjavik, Iceland

\*Corresponding author: ahooper@hi.is

Received November 2, 2006; revised April 16, 2007; accepted May 1, 2007;  
posted May 4, 2007 (Doc. ID 76643); published August 1, 2007

The problem of phase unwrapping in two dimensions has been studied extensively in the past two decades, but the three-dimensional (3D) problem has so far received relatively little attention. We develop here a theoretical framework for 3D phase unwrapping and also describe two algorithms for implementation, both of which can be applied to synthetic aperture radar interferometry (InSAR) time series. We test the algorithms on simulated data and find both give more accurate results than a two-dimensional algorithm. When applied to actual InSAR time series, we find good agreement both between the algorithms and with ground truth. © 2007 Optical Society of America

OCIS codes: 100.6890, 120.3180, 280.6730, 350.5030.

## 1. INTRODUCTION

Phase unwrapping is the process of recovering unambiguous phase values from phase data that are measured modulo  $2\pi$  rad (wrapped data). Data of this form are found in many applications, but the advent in the early 1990s of synthetic aperture radar interferometry (InSAR), in particular, spurred interest in developing reliable two-dimensional (2D) phase unwrapping algorithms [1–5]. Recent exciting areas of development within InSAR include persistent scatterer (PS) processing [6–9] and the processing of multiple conventional interferograms simultaneously [10,11]. Both of these applications provide three-dimensional (3D) wrapped phase data, the third dimension being that of time. Treating the unwrapping problem as one 3D problem as opposed to a series of 2D problems leads to an improvement in the accuracy of the solution in a similar way to which 2D unwrapping provides an improvement over one-dimensional (1D) methods.

The true phase difference between two neighboring data points for which only wrapped phase values are known has an ambiguity that is an integer number of  $2\pi$  rad. For unwrapping purposes it is usually assumed that the sampling rate is high enough over most of the data set that aliasing is avoided. In other words, the true absolute phase difference between two neighboring data points is generally less than  $\pi$  rad. The unwrapping problem then reduces to integration of the phase difference between neighboring data points, with one proviso: only certain integration paths may be taken. Specifically, we should not allow integration paths between two adjacent points when the absolute difference between the two is greater than  $\pi$ , a condition referred to as a phase discontinuity in the rest of this paper. Phase discontinuities occur in areas that are locally undersampled or where true discontinuities occur in the data, e.g., due to layover in a

topographic interferogram or a surface-breaking fault in a deformation interferogram. Given the position of the phase discontinuities the problem is then easily solved, as long as the data sampling is dense enough that there are no disconnected regions, i.e., regions with no allowable paths connecting them. However, we do not generally know *a priori* where the phase discontinuities occur and the major goal of most unwrapping algorithms is, therefore, to best determine their positions.

In a 1D problem, any phase discontinuity will cause a regional disconnect. Furthermore, the only mechanism by which we can estimate the position of the phase discontinuities is by making assumptions about the functional form of the phase variation. Moving from one dimension to two dimensions is helpful in two respects. First, there are more potential paths to choose from, increasing the effective sampling, which in turn decreases the chances of disconnected regions. Second, we are provided with clues as to the position of the phase discontinuities by the presence of residues in the data. A residue, or phase singularity, is a point around which integrating the phase gradient does not return zero [1]. We might expect, therefore, similar benefits when moving from two dimensions to three and, indeed, this is the case. The effective sampling is again increased, decreasing the chance of disconnected regions. We are also provided with further clues as to the positions of phase discontinuities because residues, which are isolated points in two dimensions, are closed loops in three dimensions [12], thus providing *a priori* information on how residues are interconnected by phase discontinuities.

Several algorithms have recently been developed that take advantage of the third dimension including branch-cut algorithms [12–16] and an integer least-squares algorithm [17]. None of these algorithms are directly applicable to InSAR time series, however. We develop here a

theoretical minimum  $L^p$ -norm framework for 3D phase unwrapping and formulate the problem for the general case. We have not, as yet, developed an  $L^p$ -norm algorithm to find the optimal solution for the general case, but we describe here an algorithm we implement to find the solution in some cases. We also describe a stepwise 3D algorithm that can be applied in all cases. This algorithm is less accurate than the  $L^p$ -norm algorithm for cases where our current implementation of the  $L^p$ -norm algorithm is applicable. However, as the stepwise algorithm is applicable in the general case, it is still useful.

We show that both algorithms give more accurate results than repeated application of a 2D algorithm. We demonstrate the algorithms on a well-sampled case, PS time series acquired over Lost Hills in California, and show that both methods are reliable and robust. The stepwise algorithm outperforms the  $L^p$ -norm algorithm on a less well-sampled case, PS time series acquired over Long Valley caldera in California, as verified by agreement with ground truth.

Notationally, we use the terms “vertices” for data points and “edges” for connections between neighboring data points. Note that these are not equivalent to the “nodes” and “arcs” used in network approaches to phase unwrapping [3,18], which refer to the locations and connections of phase-difference loop integrals, rather than the data points themselves.

## 2. THEORY OF THREE-DIMENSIONAL UNWRAPPING

In 2D phase data, neighboring phase discontinuities link to form a “discontinuity network,” which resembles a tree-like structure [1]. Residues occur at points along the discontinuity network when the total phase skipped by phase discontinuities around a point changes, including most points where the discontinuity branches and at discontinuity termini. Often, one or more residues lies outside the observed region, and the discontinuity network appears to terminate at a region boundary.

In three dimensions, neighboring phase discontinuities link to form a “discontinuity surface” (see Fig. 1), across which the phase change is greater than  $\pi$  in magnitude. Any 2D slice through a discontinuity surface will result in a discontinuity network, which is bounded by residues at the termini from 2D theory. This implies that a discontinuity surface must be bounded at all edges by residues. As the edge of any surface is closed loop, the residues

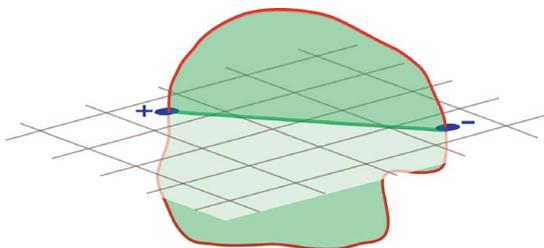


Fig. 1. (Color online) Simple phase discontinuity surface intersecting a 2D data set. The surface is bounded by a residue loop. Where the surface intersects the 2D data set results in a discontinuity line, which is bounded by a positive and a negative residue, arising where the residue loop intersects the data set.

along a discontinuity surface edge also form a closed loop. Furthermore, it can be shown that all residues, including those within discontinuity surfaces, form loops in 3D space [12,16]. These loops are referred to in previous publications as “phase singularity loops” [12,16], but we use the term “residue loop” to be consistent with our use of the term “residue” to describe phase singularities. Whenever a residue loop intersects a 2D data array a residue is detectable. Part, or parts, of a residue loop may lie outside the sampled volume, in which case we refer to the parts that lie within the sampled volume as “truncated residue loops.” Discontinuity surfaces associated with truncated residue loops appear to terminate at one or more of the sampled volume boundaries.

There is a special type of phase discontinuity in 2D space that produces no residues, which is a closed loop (not to be confused with a residue loop in 3D space). This occurs when, due to undersampling, an entire region is isolated from the rest of the data by a phase discontinuity. In this case it is not possible to unwrap reliably in two dimensions without making assumptions about the functional form of the phase variation. The equivalent in 3D space is a closed surface, which has no edges. In this case it is not possible to unwrap reliably in three dimensions without making assumptions about the functional form.

### A. Single-Cycle Discontinuity Surfaces

Single-cycle discontinuity surfaces are those where the discontinuity is always a single phase cycle. In this case, residue loops occur at all edges and branches of the surface and never within the surface. The simplest surface is an open surface bounded by one edge only (Fig. 1). Provided it cuts the edge, any 2D slice intersecting the surface will result in a discontinuity line connecting a positive residue to a negative residue, which is the simplest possible discontinuity network in two dimensions. Just as discontinuity networks in two dimensions can branch, so can discontinuity surfaces in three dimensions. A single-cycle 2D branch point will result in a residue as in Fig. 2(a). Similarly a single-cycle 3D branch line will result in a residue line, which will connect with an edge to complete a residue loop, e.g., Fig. 2(b). Branching surfaces can therefore be considered as multiple, touching, simple surfaces. A single-cycle discontinuity surface can also be bounded by multiple edges resulting in multiple residue loops, e.g., a surface in the form of a cylinder or catenoid. These surfaces naturally contain 2D closed-loop discontinuities. In other words, they are a result of undersampling in two dimensions.

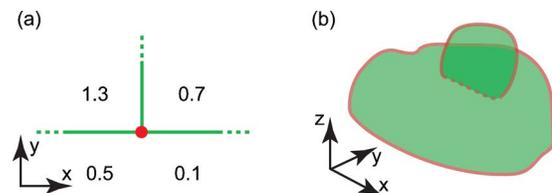


Fig. 2. (Color online) Single-cycle branch in (a) 2D and (b) 3D. The numbers in (a) represent unwrapped phase values in cycles, the lines represent single-cycle discontinuities, and the dot represents a residue at the branch point. In (b), the surfaces represent single-cycle discontinuity surfaces and the lines represent residue loops, with the dashed part being the branch line.

### B. Multiple-Cycle Discontinuity Surfaces

Multiple-cycle discontinuity surfaces are those that include phase discontinuities greater than one cycle. The relationship between multiple-cycle discontinuity surfaces and residue loops is more complex. Residue loops still occur at all edges, but not necessarily at all branches, and also occur internally on the surface (see Fig. 3).

Existing 3D branch-cut algorithms [12,14–16] make the assumption that discontinuity surfaces are (a) bounded by only one residue loop and (b) have no internal residue loops. Only when this is the case does the ambiguity of pairing positive and negative residues, which exists in two dimensions, disappear. In the case of multiple-loop surfaces, that can be single or multiple-cycle, residues may be paired between the loops, not only within each loop.

### C. Minimum $L^p$ -norm Framework

A framework was provided by Ghiglia and Romero [2] that unifies many of the unwrapping approaches in two dimensions. Phase unwrapping is expressed as an optimization problem within this framework. The goal is to find the solution that minimizes an objective function of the form:

$$\sum_{i,j} w_{i,j}^{(x)} |\Delta\phi_{i,j}^{(x)} - \Delta\psi_{i,j}^{(x)}|^p + \sum_{i,j} w_{i,j}^{(y)} |\Delta\phi_{i,j}^{(y)} - \Delta\psi_{i,j}^{(y)}|^p, \quad (1)$$

where  $\Delta\phi^{(x)}$  and  $\Delta\psi^{(x)}$  are the  $x$  direction components of the unwrapped and wrapped phase differences, respectively;  $\Delta\phi^{(y)}$  and  $\Delta\psi^{(y)}$  are the equivalent  $y$  direction components; and  $w$  represents user-defined weights. The summations are carried out in both  $x$  and  $y$  directions over all  $i$  and  $j$ , respectively. This is referred to as an  $L^p$ -norm objective function by Chen [19], although, strictly speaking, to meet the condition of positive scalability the sum must be raised to the power of  $1/p$ , and  $p$  must be greater than or equal to 1. For continuity, however, we follow the convention of Chen [19] in referring to the solution that minimizes the above function as an  $L^p$ -norm solution for  $p \geq 0$ , and note that for  $p \geq 1$  the solution that minimizes this objective function also minimizes the true  $L^p$ -norm. This framework can be extended to 3D by minimizing an objective function of the form:

$$\sum_{i,j,k} w_{i,j,k}^{(x)} |\Delta\phi_{i,j,k}^{(x)} - \Delta\psi_{i,j,k}^{(x)}|^p + \sum_{i,j,k} w_{i,j,k}^{(y)} |\Delta\phi_{i,j,k}^{(y)} - \Delta\psi_{i,j,k}^{(y)}|^p + \sum_{i,j,k} w_{i,j,k}^{(z)} |\Delta\phi_{i,j,k}^{(z)} - \Delta\psi_{i,j,k}^{(z)}|^p, \quad (2)$$

where  $\Delta\phi^{(z)}$  and  $\Delta\psi^{(z)}$  are the phase difference components in the third dimension, and the summations are extended to the  $z$  direction over all  $k$ .

An additional constraint is that of “congruence,” meaning that the unwrapped phase values differ from the wrapped phase values only by integer multiples of  $2\pi$ . Algorithms based on minimization of  $L^0$ - and  $L^1$ -norms typically enforce congruence, while  $L^2$ -norms often do not, due to the way in which they are implemented. The result of not enforcing congruence, however, is to systematically underestimate true unwrapped phase gradients [4,20], so congruence is desirable.

In two dimensions at least,  $L^0$ -norm solutions are seen empirically to give more accurate solutions than minimizing other  $L^p$ -norms [4,21] and, hence, we might expect this to be the case in three dimensions. However, finding the solution that minimizes the  $L^0$ -norm is a nondeterministic polynomial-time hard (NP-hard) problem even in two dimensions [19]. Thus, for all practical purposes, it is impossible to determine the exact solution, although various methods have been implemented in two dimensions that find an approximate solution, e.g., using network-flow theory [3,18]. Here, we do not attempt to find the  $L^0$ -norm solution for the general case but, instead, we consider higher order norms.

If congruence is enforced, the only difference between an  $L^0$ - and an  $L^1$ -norm is that the latter penalizes multiple-cycle discontinuities over single-cycle discontinuities [19]. Increasing  $p$  further penalizes multiple-cycle discontinuities and in the limit  $p \rightarrow \infty$  single-cycle discontinuities are always selected over multiple-cycle discontinuities. Clearly then, if there are multiple-cycle discontinuities present, the  $L^\infty$ -norm solution will not be correct, but in the case where only single-cycle discontinuities exist it will be identical to the  $L^0$ -norm solution. In other words, in the case where the data and sampling are such that there are no multiple-cycle discontinuities, the  $L^\infty$ -norm solution is also the best solution within the  $L^p$ -norm framework. Most interferograms that include topographic signature contain multiple-cycle discontinuities due to the presence of layover, and the  $L^\infty$ -norm solution is therefore not suitable. However, for interferograms where the topographic signature has been mostly subtracted, as in most deformation signal analyses, the  $L^\infty$ -norm solution may be useful.

Fitting an area-minimizing “minimal surface” to every residue loop gives a solution with no multiple-cycle discontinuities, yielding a congruent minimum  $L^\infty$ -norm solution. An  $L^\infty$ -norm solution with equal weights,  $w_{i,j,k}$ , would give more weight to more densely sampled regions and hence tend to place discontinuity surfaces in less densely sampled regions. To avoid this bias,  $w_{i,j,k}$  can be chosen to normalize by sampling density. A minimal surface automatically gives the solution to the problem with normalized values of  $w_{i,j,k}$ , as it is not biased by sampling density. Strictly speaking the solution that minimizes the

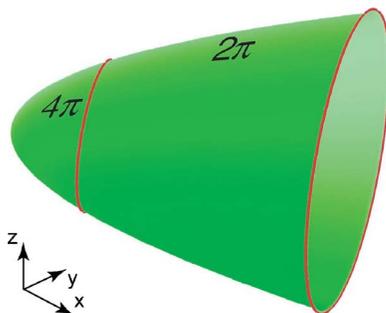


Fig. 3. (Color online) Multiple-cycle discontinuity surface. The loops represent residue loops, the right loop bounding the edge of the surface and the left loop separating the  $2\pi$  phase discontinuity region of the surface from the  $4\pi$  discontinuity region. Note that an  $L^\infty$ -norm solution would place a minimal surface within each residue loop.

$L^\infty$ -norm could include single-cycle discontinuity surfaces that are bounded by multiple loops, which result in closed-loop discontinuities in 2D. Although we expect these surfaces to be rare in well-sampled data sets, fitting a minimal surface to every loop does not allow for this type of surface, hence we refer to it as a quasi- $L^\infty$  approach. Existing branch-cut algorithms [12,14–16] also fall in this class, although the ways in which truncated residue loops are handled by these algorithms do not minimize the  $L^\infty$ -norm for the general case.

In the following section we describe an algorithm to implement the quasi- $L^\infty$  approach that applies for all residue loops, including those that are truncated. In the case where the data are sampled sufficiently well that there are no closed-loop or multiple-cycle discontinuities, and the underlying nature of the data is such that these discontinuity types do not arise, we expect this solution to be identical to the  $L^0$ -norm solution. For more general applications, a true  $L^0$ -norm solver should be developed and a network-flow model [3,18] may provide a means to achieve this. Optimization using generalized, nonlinear cost functions, as done in two dimensions by Chen and Zebker [5], could lead to a more accurate solution still, particularly in the case of InSAR, as we might expect cost functions in the space-time domain to be quite different to those purely in the spatial domain.

### 3. QUASI- $L^\infty$ -NORM THREE-DIMENSIONAL ALGORITHM

We have developed an algorithm to unwrap 3D data that are irregularly sampled in two dimensions in order to unwrap InSAR PS time series, which are distributed irregularly in space. Therefore, the objective function from Eq. (2), which we seek to minimize, is modified to

$$\sum_{i,j} w_{i,j}^{(s)} |\Delta\phi_{i,j}^{(s)} - \Delta\psi_{i,j}^{(s)}|^p + \sum_{i,j} w_{i,j}^{(t)} |\Delta\phi_{i,j}^{(t)} - \Delta\psi_{i,j}^{(t)}|^p, \quad (3)$$

where superscript  $s$  represents a phase difference between two neighboring data points in the two spatial dimensions, superscript  $t$  represents a phase difference in the time dimension, and the summations are carried out in 2D space, over all  $i$ , and, in time, over all  $j$ .

The algorithm allows for residue loops to be terminated any number of times at any number of data volume boundaries. For each loop, the algorithm selects a phase discontinuity surface that is a minimal surface, equivalent to the surface formed by an equilibrium soap film in zero gravity. In this algorithm we assume that each discontinuity surface is bounded by only one loop, as discussed in Subsection 2.C.

#### A. Residue Identification

Residues are identified by dividing the 3D data into a number of faces with a data point at each vertex, calculating the difference in phase along edges between adjacent vertices, wrapping the difference into the interval  $-\pi$  to  $\pi$ , and integrating the differences around the face. A nonzero sum indicates a residue. In two dimensions, if the data are regularly sampled the simplest approach is to divide the data into rectangular faces. If the data are irregularly sampled (sparse), Delaunay triangulation can

be used to divide the data into triangular faces [22]. Similarly, in three dimensions, if the data are regularly sampled the simplest approach is to divide the data into rectangular faces in three orthogonal orientations. However, to be applicable to InSAR data sets that are sparse in the two spatial dimensions, we use Delaunay triangulation to define edges connecting the data points within the two spatial dimensions and integrate around each triangle. In the time dimension, we assume that the data are regularly sampled, in the sense that for every sample in time the phase is sampled at every point in the two spatial dimensions. Therefore, we can define edges that divide the data into rectangular faces and integrate around the rectangles. The faces in this irregular grid outline a series of wedge-shaped elements, each with two triangular faces and three rectangular faces (see Fig. 4). The algorithm could be extended to the case where data are irregular in all three dimensions by connecting the data points through Delaunay tetrahedrization resulting in tetrahedron-shaped elements.

#### B. Linking Residues

As argued in Section 2, residues form loops in 3D space that are detectable wherever they intersect a face. This implies that any loop entering a wedge element through one face must exit the wedge element through another face [see Fig. 4(a)]. Residues can be linked, therefore, by tracing the loops through each wedge element in turn, until the loop either closes on itself or a data volume bound-

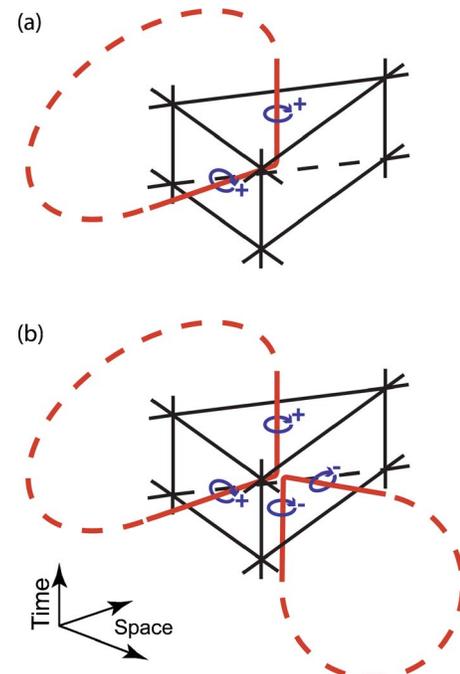


Fig. 4. (Color online) Residue identification and connection. Each apex represents a data point and the connecting lines represent the arcs along which phase differences are calculated. Residues, indicated by a + or -, are identified by integrating arc phase differences around each triangular or rectangular face. (a) shows an element with residues on two faces. A residue loop enters the element through one face and exits through the other. (b) shows an element with residues on four faces. The four residues could be linked to form either two separate loops (as shown) or one twisted figure-eight loop.

ary is reached. As each wedge element has five faces, it is possible for two residue loops to pass through the same element, intersecting four faces in total [see Fig. 4(b)]. The residues on the intersected faces could then be linked in two different ways: one way forming two distinct loops, and the other way forming only one loop in the form of a twisted figure eight, with the crossover point placed at the center of the element. However, the discontinuity surfaces found by a subsequent step of the algorithm would be identical for both scenarios, so it is not necessary to distinguish between them.

A residue naturally has sign and in two dimensions, the sum of all residues on a phase discontinuity network must be zero, assuming no truncation by the data set boundary [1]. That is to say, for any 2D slice through a 3D discontinuity surface the residues must sum to zero. If residues are only calculated in distinct 2D planes then, as is the case for a regular 3D grid, all residues in a closed residue loop will sum to zero. However, if residues are calculated on arbitrary planes, as in the algorithm we present here for sparse data, the sum of the residues in a closed residue loop need not be zero.

### C. Processing Closed Residue Loops

As discussed in Subsection 2.C, we seek a solution that fits a single-cycle phase discontinuity surface to each residue loop, which is equivalent to finding the quasi- $L^\infty$ -norm solution. Any given closed residue loop bounds an infinite number of surfaces, but minimizing the  $L^\infty$ -norm corresponds to finding the area-minimizing minimal surface. This is equivalent to the surface that would be formed by an equilibrium soap film in zero-gravity. We use an iterative program called surface evolver [23] to find an approximation to this surface for each closed residue loop.

### D. Processing Truncated Residue Loops

In practice, some residue loops may lie partially outside our data volume, causing truncation of the residue loops at the data volume boundary. Loops can pass through the boundary any even number of times, resulting in any even number of truncated residue loops (Fig. 5). The assumption that every truncated loop closes on itself [12] is, therefore, not necessarily correct, and a more complex algorithm is needed to close all the truncated loops. We allow for the linking of multiple truncated loops, including those truncated at different data volume boundaries, and attempt to minimize the total surface area of the surfaces

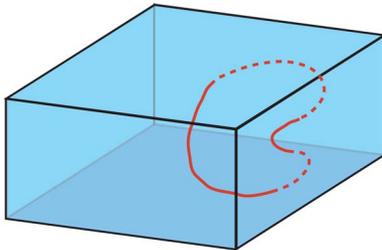


Fig. 5. (Color online) Residue loop that is truncated at the data volume boundaries. The continuous parts of the loop are contained within the data volume, while the dashed parts lie outside. The loop is truncated four times, once at the top boundary and three times at the right boundary.

formed. Up to now, surface area has had only relative meaning, as the units are not the same in all dimensions (two are in space and one is in time). This does not matter when fitting a minimal surface to a predefined closed loop in Subsection 3.C, as scaling in any dimension simply scales the minimal surface but does not alter its position. With the option of closing loops across differing dimensions, however, the dimension scaling does matter. We therefore scale the time dimension to be consistent with the two spatial dimensions. We assume that the probability that an edge bounds a residue is related to the length of the edge. Hence, we expect a relationship between the length of an edge and the proportion of edges of that length that bound a residue. We find empirically that in the two spatial dimensions the data are reasonably well-fit by the logarithmic model

$$\frac{N_{res}}{N_{edge}} = k \log L_{edge} + c, \quad (4)$$

where  $L_{edge}$  is the length of an edge,  $N_{res}$  is the number of edges of length  $L_{edge}$  bounding a residue,  $N_{edge}$  is the total number of edges of length  $L_{edge}$ , and  $k$  and  $c$  are constants (see Fig. 6). We invert the data from the two spatial dimensions to estimate  $k$  and  $c$ , in a least-squares sense, and use these values in the time dimension to estimate an equivalent  $L_{edge}$  between each interferogram.

As noted above, we seek the combination of closed loops that minimizes the total surface area of the surfaces formed, which we implement with a Monte Carlo algorithm. Connecting two close ends is more likely to lead to a minimal surface than two distant ends, as the connection between the two ends itself forms one of the boundaries of the resulting surface. Our approach, then, is to start with a solution in which all partial loops are self-closing. The total surface area of this solution becomes our initial minimum solution. We then select one of the loop connections, with probability of being chosen being proportional to its length, and break it. This leads to two free ends,  $A$  and  $B$ . For free end  $A$ , the next nearest end,  $C$ , is found, currently connected to  $D$ . There is a probability that we break the connection of  $C$  to  $D$  and make a new connection  $A$  to  $C$ , such that

$$P = \frac{AB}{AB + AC}, \quad (5)$$

where  $AB$  is the distance between  $A$  and  $B$  and  $AC$  is the distance between  $A$  and  $C$  (Fig. 7). In other words, if the

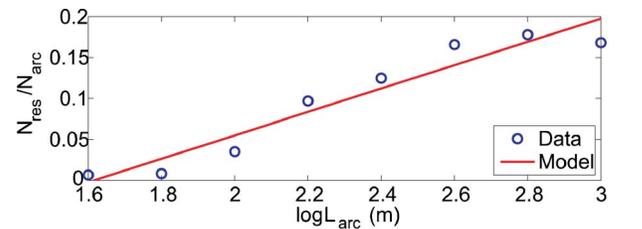


Fig. 6. (Color online) Relationship between the length of the arcs connecting data points,  $L_{arc}$ , and the fraction of arcs bounding residues,  $N_{res}/N_{arc}$ . These data are from the Lost Hills example (Subsection 5.B). The model we fit assumes a logarithmic relationship.

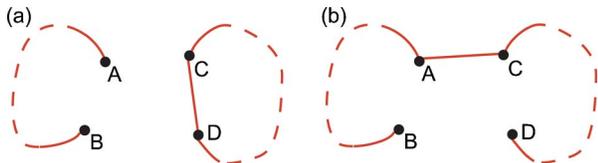


Fig. 7. (Color online) Optimizing the connection of truncated residue loops. In (a), the connection between truncated loop ends  $A$  and  $B$  is broken. In (b) the connection between end  $C$  and  $D$  is broken, and  $A$  is reconnected to  $C$ . The probability of proceeding from (a) to (b) is given by  $P=AB/(AB+AC)$  where  $AB$  is the distance between  $A$  and  $B$ , and  $AC$  is the distance between  $A$  and  $C$ .

change leads to a reduction in length, it is favored. If the connection is not broken, we find the next-nearest end and go through the same exercise, repeating until a new free end is formed. The same procedure is executed for the other free end,  $B$ . This process of breaking and forming connections proceeds until the two free ends connect to each other, at which point we have a new solution. If the total surface area of the new solution is less than the current minimum solution, it becomes the new minimum solution. We continue to seek new solutions until a specified number of consecutive new solutions are found without finding a new minimum solution. The number is arbitrary and typically we use 100.

### E. Integration of Phase

We next identify the edges that are intersected by discontinuity surfaces and use a flood-fill algorithm to integrate the phase, with the intersected edges being barriers to flow. Specifically, we start from an arbitrary data point and integrate the phase along unintersected edges to the neighboring data points. Starting from these points we then integrate the phase again along unintersected edges to neighboring data points not yet reached. This continues until no further data points can be reached.

### F. Application to InSAR Time Series

As is the case for 2D unwrapping algorithms, the 3D algorithm described above is applicable if the sampling rate is high enough over most of the data set that aliasing is avoided. In InSAR time series, however, the phase is undersampled in time for every point in space, due to the change in atmospheric delay, which can vary by greater than  $\pi$  in much less than the time between acquisitions for all existing SAR data sets. There is also a phase term due to error in orbital estimation that approximates a ramp in space. Though often small, this term can also be greater than  $\pi$  in magnitude.

We reduce these terms to less than  $\pi$  over most of the image by estimating and subtracting the longer wavelength components of the phase variation between each interferogram, which include most of the atmospheric and orbital error signal [24]. For each pair of interferograms in time, we estimate the highest frequency component of spatially correlated phase that is possible given our sampling density, i.e., we avoid aliasing. We transform the complex phase difference between the interferogram pair to the frequency domain and iteratively low-pass filter, starting with a broad frequency response and decreasing the width until the filtered phase contains no residues. Unwrapping of the filtered phase is therefore unambigu-

ous. To enable use of the fast Fourier transform the complex phase difference is first sampled to a grid, using a grid spacing over which little variation in phase is expected (typically 40–100 m). Where multiple pixels fall in the same grid cell, the complex phase is summed. There remains a component of the atmospheric phase between each interferogram, which is an integer number of  $2\pi$ , and which we are not able to estimate. However, we are not interested in the absolute value of the atmospheric phase, and a constant integer number of  $2\pi$  offset has no effect on unwrapping.

We subtract the low-pass filtered phase and unwrap it separately. Long-wavelength deformation may be included in this phase, but this is of no concern as it is only subtracted temporarily and, after unwrapping, is added back. We apply the quasi- $L^\infty$ -norm algorithm to the remaining phase.

### G. Optional Prefiltering

Optionally, if the data are very noisy, a prefiltering step can be run to filter the data spatially before unwrapping, as is common in two dimensions [25]. For each time step, the complex phase data are first sampled to a grid, as above. The gridded phase is then transformed and filtered in the frequency domain using an adaptive phase filter [25]. This preserves the dominant frequencies that are present in the data. After inverse transformation, the grid cells containing data are treated as the new data points for input into the unwrapping algorithm.

## 4. STEPWISE THREE-DIMENSIONAL ALGORITHM

As discussed in Subsection 2.C, the quasi- $L^\infty$ -norm solution is only applicable when there are no multiple-loop discontinuity surfaces in a data set. In the future we hope to implement an  $L^0$ -norm algorithm for these cases. In the absence, as yet, of an efficient algorithm to find the  $L^0$ -norm solution, we describe here an alternate 3D approach.

The algorithm first unwraps the data in one dimension: time in the case of InSAR time series. This results in an initial solution for optimization in the other two dimensions. Although not optimal in a 3D sense, this approach is applicable when multiple-loop discontinuity surfaces are present in the data, unlike the quasi- $L^\infty$  approach. Furthermore, several efficient 2D unwrapping routines already exist, e.g., minimum-cost flow [3], statistical-cost network-flow [5], and iterative least squares, which is described in this section.

As discussed in Subsection 3.F, when dealing with InSAR data, changes in atmospheric delay between passes can lead to total decorrelation in the time dimension, so it is not possible to unwrap the phase of individual data points in time. However, by using instead the phase difference between nearby pixels, the atmospheric contribution is largely canceled. Our approach is to use Delaunay triangulation to define edges connecting data points in the two spatial dimensions. For each edge, we calculate the phase difference in each PS interferogram and unwrap these phase differences in time. The unwrapping step consists first of low-pass filtering the complex phase

difference time series in the frequency domain using a Gaussian window. The phase difference between each filtered data point is then calculated, wrapped to be between  $-\pi$  and  $\pi$ , and integrated. The original edge time series is unwrapped on the basis that the absolute difference between it and the unwrapped filtered time series must be less than  $\pi$ .

Any 2D phase unwrapping algorithm for sparse data can be used to unwrap the data in the spatial domain. We implement our algorithm using an iterative weighted least-squares approach. For each interferogram, the unwrapped edge phase differences from the first step are inverted to give the phase at each pixel using weighted least-squares. For weighting we use the inverse of the standard deviation of the difference between the edge time series and the filtered edge time series. We then calculate residuals between the edge phase differences from the first step and edge phase differences predicted by the model, and drop the edges with the largest residuals. The process is repeated until all residuals are zero. The optimal solution using this approach would be achieved by dropping only one edge per iteration. Computationally this is expensive, however, so typically we initially drop a maximum of 0.1% of the edges per iteration. If dropping multiple edges will lead to the inversion becoming rank deficient we reduce the maximum number to be dropped by a factor of 10. We repeat this reduction until full rank is achieved, which is always the case once the maximum number becomes one. As in the case of the quasi- $L^\infty$  3D algorithm, if the data are very noisy, an optional prefiltering step can be run to filter the data spatially before unwrapping (Subsection 3.G).

## 5. EXPERIMENTAL RESULTS

### A. Simulated Data Example

We first assess our phase unwrapping algorithms by examining a set of interferograms we formed for a simulated center of dilation within a homogeneous elastic half-space [26] at 2 km depth, with volume changing at a rate defined by a random walk (see Fig. 8). We selected 4000 points randomly within a 20 km  $\times$  20 km area above the point source. The surface deformation was calculated at these points for 20 randomly spaced satellite passes, assuming a satellite repeat period of 35 days, and converted into line-of-sight phase difference with respect to the tenth pass. We added a random atmospheric phase signal for each scene with the spectral power following a  $-5/3$  power law for wavelengths larger than 2 km and  $-8/3$  for wavelengths less than 2 km [24]. The maximum variation of the atmospheric signal simulated for each interfero-

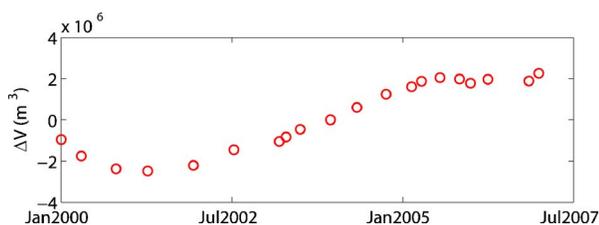


Fig. 8. (Color online) Simulated change in volume of a center of dilation that is used to calculate the deformation phase contribution to the simulated phase data in Fig. 9.

gram has a mean of 2.9 rad. We also added white noise, with a standard deviation of  $50^\circ$ , to every scene. Finally we differenced every scene with respect to the tenth scene as our “master” scene, and wrapped the data (Fig. 9).

The results of unwrapping using the quasi- $L^\infty$ -norm 3D algorithm and the stepwise 3D algorithm are shown in Fig. 10. Also shown for comparison are the results of unwrapping each interferogram separately using a 2D iterative least-squares algorithm. No prefiltering of the phase in the spatial domain was applied prior to any of the algorithms. The problem becomes progressively under-sampled as the deformation gradient increases so that even the quasi- $L^\infty$ -norm 3D algorithm is unable to unwrap accurately the peak deformation signal in every interferogram. The quasi- $L^\infty$ -norm 3D algorithm unwraps the peak deformation signal more accurately than the stepwise 3D algorithm, however, which in turn does better than the 2D algorithm. Figure 11 shows the residuals between the unwrapped values and the true values. While the spatial pattern of deformation is better retrieved by the quasi- $L^\infty$ -norm algorithm than the stepwise algorithm, there are more local, single-cycle errors. This

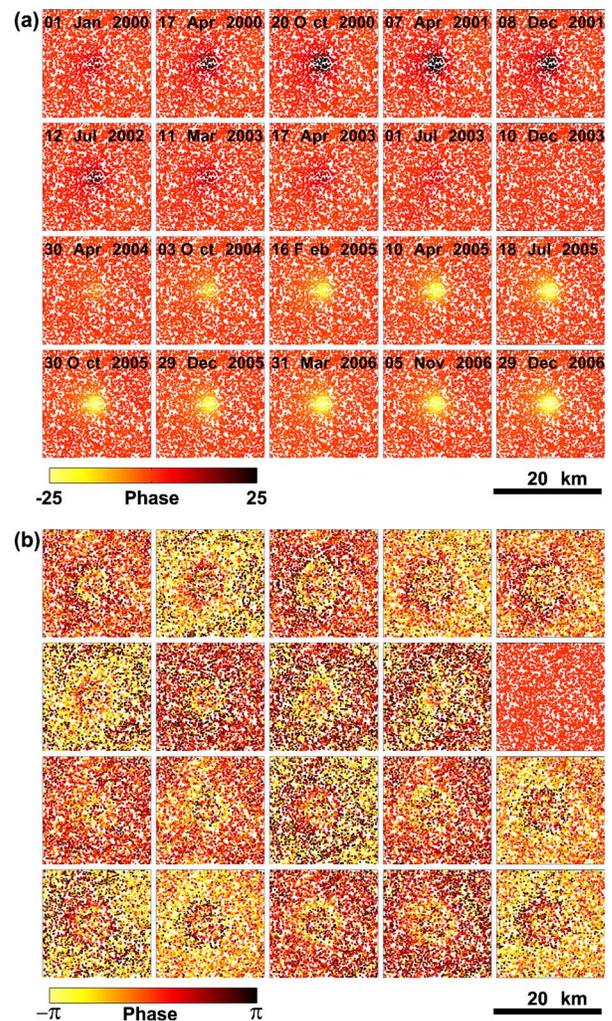


Fig. 9. (Color online) Simulated phase data in radians, (a) unwrapped and (b) wrapped, including deformation phase, atmospheric phase delay, and noise. The unwrapped phase is referenced to the top-right corner to enable comparison with Fig. 10.

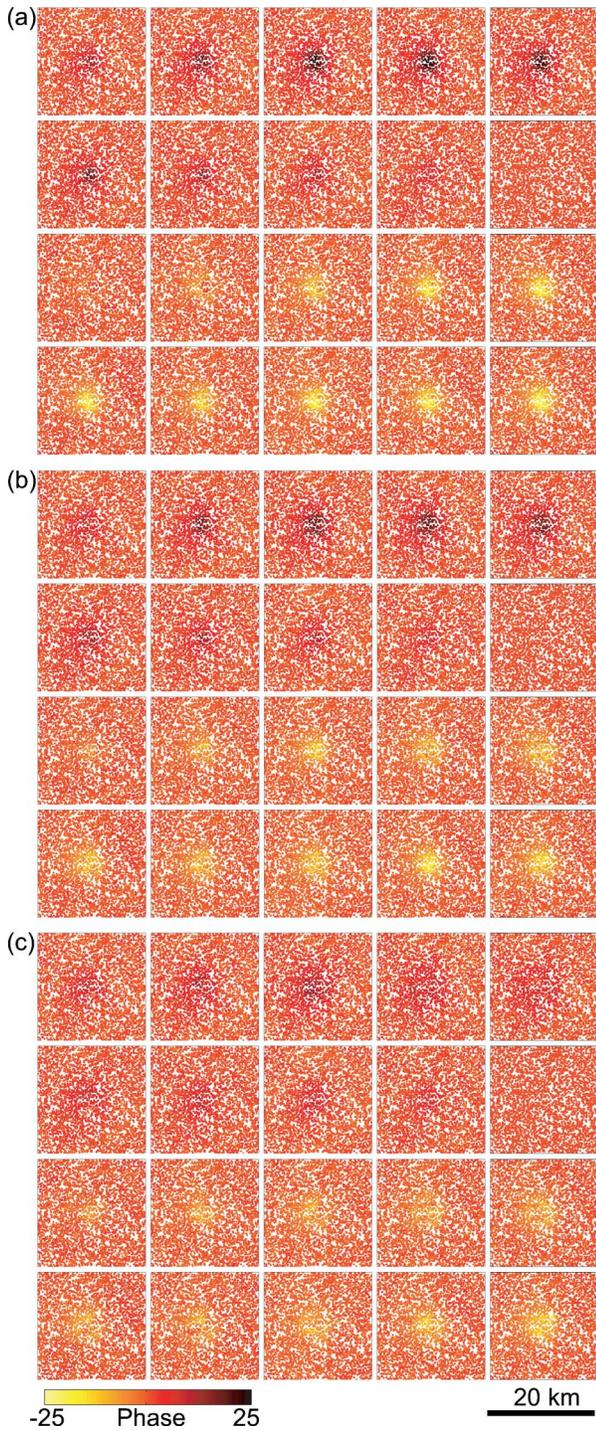


Fig. 10. (Color online) Simulated phase data in radians unwrapped using (a) the quasi- $L^\infty$  3D algorithm, (b) the stepwise 3D algorithm, and (c) an iterative least-squares 2D algorithm. In all cases the phase is referenced to the top-right corner.

is also apparent in Fig. 12, a comparison of unwrapping accuracy for the different algorithms. The quasi- $L^\infty$ -norm algorithm performs better than the stepwise algorithm in terms of multiple-cycle errors, but worse in terms of single-cycle errors. However, when recovering the deformation field we are more concerned with avoiding the systematic errors around the peak deformation than these randomly distributed errors. For our purposes, therefore, the quasi- $L^\infty$ -norm algorithm gives the best results.

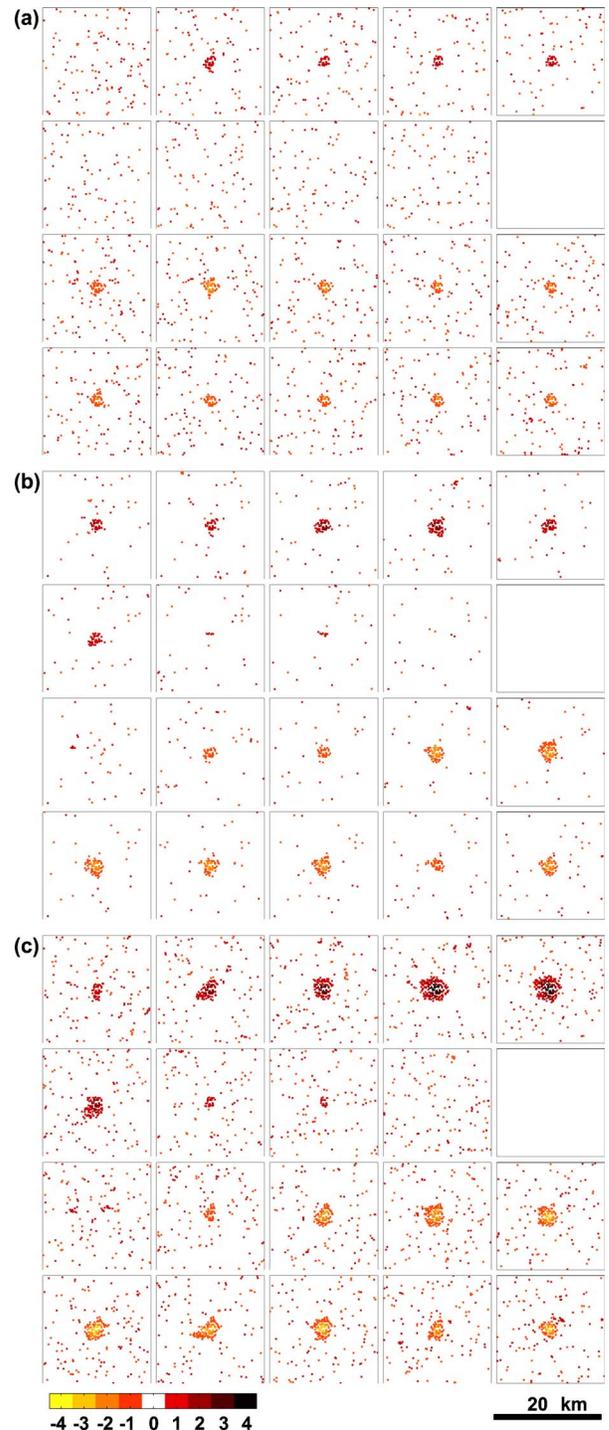


Fig. 11. (Color online) Difference in cycles between the simulated unwrapped phase and that estimated by the three algorithms, (a) the quasi- $L^\infty$  3D algorithm, (b) the stepwise 3D algorithm, and (c) an iterative least-squares 2D algorithm.

### B. Lost Hills InSAR Persistent Scatterer Example

We now present 3D phase unwrapping of two real data sets. The first data set consists of 28 interferograms for the Lost Hills oil field in California. This is an actively subsiding area due to oil extraction. The interferograms were formed from 29 fine mode (F1 beam) scenes acquired by the RADARSAT-1 satellite every 24 days between February 20, 2002 and February 10, 2004. All interferograms

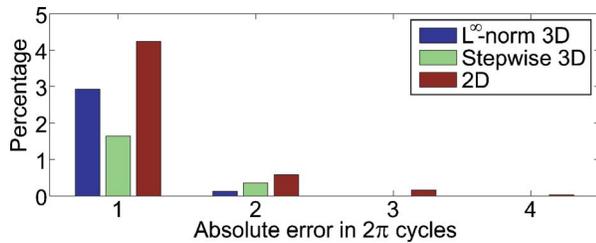


Fig. 12. (Color online) Comparison of unwrapping accuracy for the simulated data for the three different algorithms.

were formed with respect to the same master scene acquired on March 11, 2003. We used the method of Hooper *et al.* [8] to identify PS pixels and estimate and remove the look angle error terms, including those due to errors in the digital elevation model. We prefiltered the data as described in Subsection 3.G and unwrapped the phase using both the quasi- $L^\infty$ -norm and stepwise 3D algorithms. The results are shown in Fig. 13. Despite the difference in approach, the results are identical for 99% of the unwrapped phase values, and no systematic differences are evident, implying that both algorithms are feasible in this case.

### C. Long Valley InSAR Persistent Scatterer Example

We also applied the two algorithms to data acquired over Long Valley caldera in California. We processed 21 interferograms formed from 22 scenes acquired by the ERS-1 and ERS-2 satellites between June 4 1992 and August 20, 2000. All interferograms were formed with respect to the same master scene acquired on June 22, 1997. Within the caldera is a resurgent dome that, has both inflated and deflated during this period [7]. We used the method of Hooper *et al.* [8] to identify PS pixels and estimate and remove the look-angle error terms. Before unwrapping we prefiltered the data (Subsection 3.G).

The quasi- $L^\infty$  3D algorithm does not work well in this region, producing discontinuities in the unwrapped phase where none are expected. We expect this algorithm to be accurate only when the sampling is such that no multiple-loop discontinuities are present. Presumably, for these data, this is not the case.

Using the stepwise 3D unwrapping algorithm, however, the results are reasonable (Fig. 14). We validated the results against ground truth by comparison of relative vertical motion calculated from the unwrapped phase, to that measured by leveling and global positioning system (GPS) surveys, and inferred from electronic distance meter (EDM) measurements. The results are shown in Fig. 15. Horizontal motion measured by the EDM line is almost parallel to the satellite track and hence not present in our interferograms. However, this horizontal motion is almost proportional to the vertical motion of the resurgent dome [27] and, once scaled using less frequent leveling and GPS measurements, provides a proxy for relative vertical motion. We estimated the phase at each benchmark as the mean phase of all PS pixels within 500 m and the error bars show the standard deviation of these phases. In (a) we converted the phase values to relative vertical displacement by assuming that all the motion contributing to the change in phase was due to vertical motion only. In (b) we relaxed that assumption and

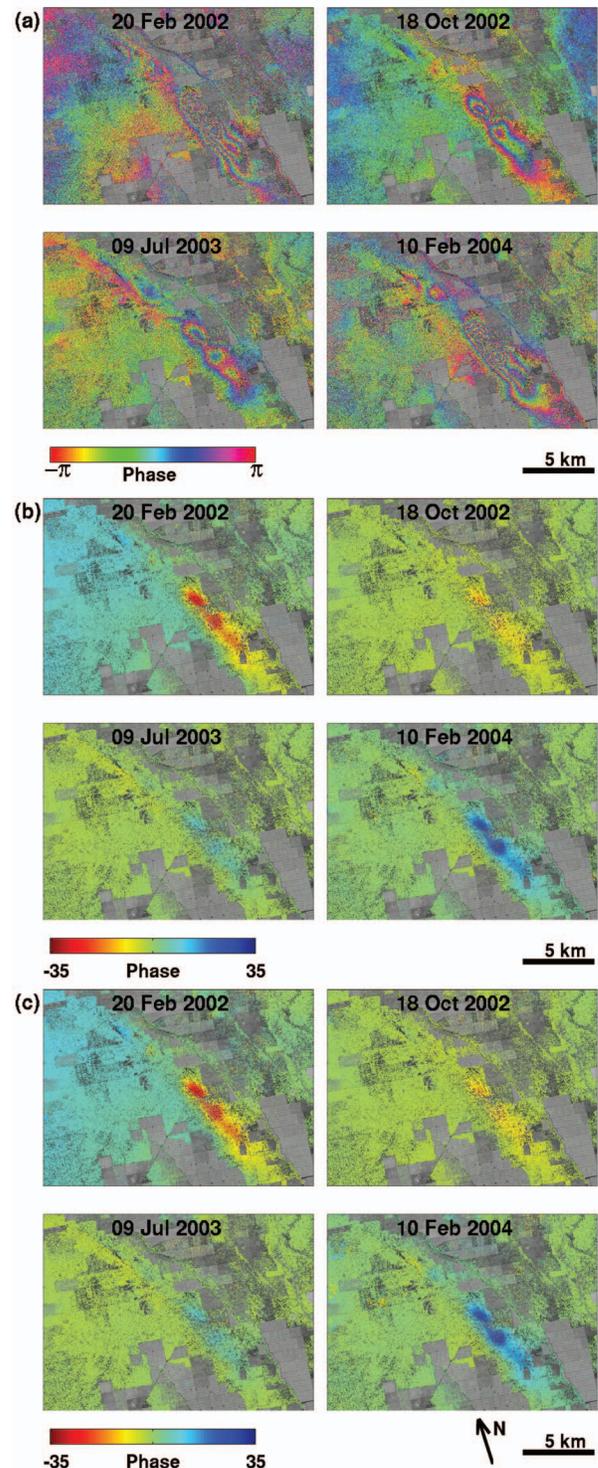


Fig. 13. Lost Hills region persistent scatterer interferograms in radians, (a) wrapped phase, (b) phase unwrapped using the quasi- $L^\infty$ -norm 3D algorithm, and (c) phase unwrapped using the stepwise 3D algorithm. The date of the master acquisition is February 22, 2003 and only every ninth interferogram is shown. The background image in gray is the mean SAR amplitude of all 28 passes and the points represent PS pixels, with color indicating the relative phase difference with respect to the northeast corner.

solved for a component of horizontal displacement proportional to the vertical component. We found the constant of proportionality through a least-squares inversion that

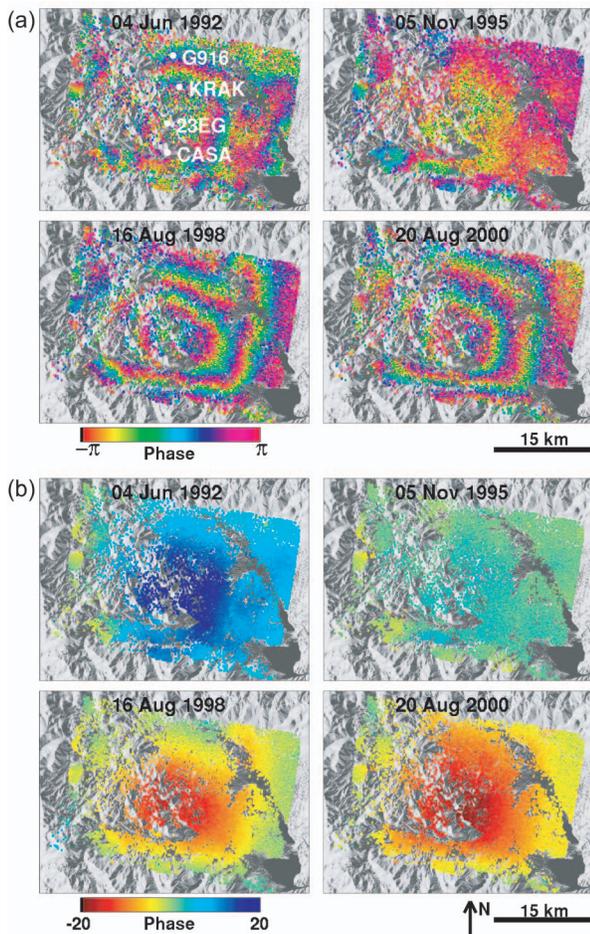


Fig. 14. Long Valley persistent scatterer interferograms in radians, (a) wrapped phase and (b) phase unwrapped using the stepwise 3D algorithm. The date of the master acquisition is June 22, 1997 and only every seventh interferogram is shown. The background image in gray is topography in shaded relief and the points represent PS pixels, with color indicating the relative phase difference with respect to the northwest corner.

minimized the difference between the vertical displacement and the scaled EDM measurements. Our results indicate that the unwrapping between the two benchmarks is reliable. Hooper *et al.* [7] unwrapped PS time series over Long Valley using a 2D algorithm and also found the unwrapping between the two benchmarks to be reasonable. However, the overall pattern of the displacements found by Hooper *et al.* [7] differs to that observed by Fialko *et al.* [28], especially on the west side of the caldera where there appears to be a discontinuity in the former data set. The overall pattern we find using the stepwise 3D algorithm, on the other hand, matches well.

## 6. CONCLUSIONS

We framed the problem of 3D unwrapping in terms of an optimization problem, and developed an algorithm to find an approximate  $L^\infty$ -norm solution. For a subset of cases, those with no multiple-cycle phase discontinuities and no 2D closed-loop discontinuities, the  $L^\infty$ -norm solution is identical to the  $L^0$ -norm solution that we expect to be the best solution within this optimization framework. We

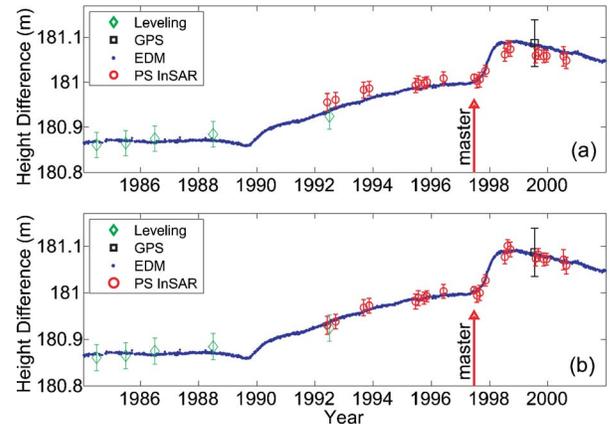


Fig. 15. (Color online) Comparison of relative vertical motion between benchmarks 23EG and G916 (see Fig. 14 for locations) from leveling and GPS, to that calculated from unwrapped PS phase, (a) assuming phase changes are due to vertical displacement only and (b) assuming there is also a horizontal component of displacement proportional to the vertical. The error bars represent 68% confidence bounds. Also shown is the scaled line length change between CASA and KRAK as measured by EDM, which is a proxy for vertical deformation.

demonstrated the accuracy of the algorithm on simulated and real data sets. We also developed a stepwise 3D algorithm that relies on first unwrapping in 1D, then iteratively improving this solution in the other two dimensions. This algorithm may be applied in the general case, although we would not expect it to be as accurate as the  $L^\infty$ -norm algorithm in the subset of cases mentioned above. We demonstrated the accuracy of this algorithm on both simulated and real data sets.

The next step is to develop an efficient algorithm to find the  $L^0$  solution to the general case, allowing application of the fully 3D approach in all cases. A further improvement still would be to formulate the optimization in terms of generalized, nonlinear cost functions, as done in two dimensions by Chen and Zebker [18].

## ACKNOWLEDGMENTS

Interferograms for the Lost Hills region were formed and corrected for orbital errors by MacDonald, Dettwiler and Associates Ltd. ERS data for Long Valley were provided by the European Space Agency, and interferograms formed using German Space Center (DLR) proprietary software. Delaunay triangulation was achieved using the Triangle program developed by Jonathan Shewchuk, University of California at Berkeley. We thank Paul Segall for useful discussions, Mark Simons for making data available, and a reviewer who helped to improve the manuscript. Support for this research was provided by the Robert and Marvel Kirby Stanford Graduate Fellowship and National Science Foundation (NSF) grant EAR-0511035.

## REFERENCES

1. R. M. Goldstein, H. A. Zebker, and C. L. Werner, "Satellite radar interferometry: two-dimensional phase unwrapping," *Radio Sci.* **23**, 713–720 (1988).
2. D. C. Ghiglia and L. A. Romero, "Minimum  $L^p$ -norm two-

- dimensional phase unwrapping," J. Opt. Soc. Am. A **13**, 1999–2013 (1996).
3. M. Costantini, "A novel phase unwrapping method based on network programming," IEEE Trans. Geosci. Remote Sens. **36**, 813–821 (1998).
  4. H. A. Zebker and Y. P. Lu, "Phase unwrapping algorithms for radar interferometry: residue-cut, least-squares, and synthesis algorithms," J. Opt. Soc. Am. A **15**, 586–598 (1998).
  5. C. W. Chen and H. A. Zebker, "Two-dimensional phase unwrapping with use of statistical models for cost functions in nonlinear optimization," J. Opt. Soc. Am. A **18**, 338–351 (2001).
  6. A. Ferretti, C. Prati, and F. Rocca, "Permanent scatterers in SAR interferometry," IEEE Trans. Geosci. Remote Sens. **39**, 8–20 (2001).
  7. A. Hooper, H. Zebker, P. Segall, and B. Kampes, "A new method for measuring deformation on volcanoes and other natural terrains using InSAR persistent scatterers," Geophys. Res. Lett. **31**, 611–615 (2004).
  8. A. Hooper, P. Segall, and H. Zebker, "Persistent scatterer InSAR for crustal deformation analysis, with application to Volcán Alcedo, Galápagos," J. Geophys. Res. **112**, B07407 (2007).
  9. B. M. Kampes, "Displacement parameter estimation using permanent scatterer interferometry," Ph.D. dissertation (Delft University of Technology, 2005).
  10. P. Berardino, G. Fornaro, R. Lanari, and E. Sansosti, "A new algorithm for surface deformation monitoring based on small baseline differential SAR interferograms," IEEE Trans. Geosci. Remote Sens. **40**, 2375–2383 (2002).
  11. D. A. Schmidt and R. Bürgmann, "Time-dependent land uplift and subsidence in the Santa Clara valley, California, from a large interferometric synthetic aperture radar data set," J. Geophys. Res. **108**, 2416–2428 (2003).
  12. J. M. Huntley, "Three-dimensional noise-immune phase unwrapping algorithm," Appl. Opt. **40**, 3901–3908 (2001).
  13. R. Cusack and N. Papadakis, "New robust 3D phase unwrapping algorithm: Application to magnetic field mapping and undistorting echo-planar images," Neuroimage **16**, 754–764 (2002).
  14. O. Marklund, J. M. Huntley, and R. Cusack, "Robust unwrapping algorithm for 3-D phase volumes of arbitrary shape containing knotted phase singularity loops," research report (Luleå University of Technology, 2005).
  15. M. F. Salfity, P. D. Ruiz, J. M. Huntley, M. J. Graves, R. Cusack, and D. A. Beauregard, "Branch cut surface placement for unwrapping of undersampled three-dimensional phase data: application to magnetic resonance imaging arterial flow mapping," Appl. Opt. **45**, 2711–2722 (2006).
  16. M. F. Salfity, J. M. Huntley, M. J. Graves, O. Marklund, R. Cusack, and D. A. Beauregard, "Extending the dynamic range of phase contrast magnetic resonance velocity imaging using advanced higher-dimensional phase unwrapping algorithms," J. R. Soc., Interface **3**, 415–427 (2006).
  17. M. Jenkinson, "Fast, automated  $N$ -dimensional phase-unwrapping algorithm," Magn. Reson. Med. **49**, 193–197 (2003).
  18. 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**, 401–414 (2000).
  19. C. W. Chen, "Statistical-cost network-flow approaches to two-dimensional phase unwrapping for radar interferometry," Ph.D. dissertation (Stanford University, 2001).
  20. R. Bamler, N. Adam, G. W. Davidson, and D. Just, "Noise-induced slope distortion in 2-D phase unwrapping by linear estimators with application to SAR interferometry," IEEE Trans. Geosci. Remote Sens. **36**, 913–921 (1998).
  21. D. C. Ghiglia and M. D. Pritt, *Two-Dimensional Phase Unwrapping: Theory, Algorithms and Software* (Wiley, 1998).
  22. M. Costantini and P. A. Rosen, "Generalized phase unwrapping approach for sparse data," in *Proceedings of the IEEE 1999 International Geoscience and Remote Sensing Symposium (IGARSS)* (IEEE 1999), Vol. 1, pp. 267–269.
  23. K. Brakke, "The surface evolver," Exp. Math. **1**, 141–165 (1992).
  24. R. F. Hanssen, *Radar Interferometry Data Interpretation and Error Analysis* (Springer, 2001).
  25. R. M. Goldstein and C. L. Werner, "Radar interferogram filtering for geophysical applications," Geophys. Res. Lett. **25**, 4035–4038 (1998).
  26. K. Mogi, "Relations between the eruptions of various volcanoes and the deformations of the ground surfaces around them," Bull. Earthquake Res. Inst., Univ. Tokyo **36**, 111–123 (1958).
  27. M. Battaglia, P. Segall, J. Murray, P. Cervell, and J. Langbein, "The mechanics of unrest at Long Valley caldera, California: 1. Modeling the geometry of the source using GPS, leveling and two-color EDM data," J. Volcanol. Geotherm. Res. **127**, 195–217 (2003).
  28. Y. Fialko, M. Simons, and Y. Khazan, "Finite source modelling of magmatic unrest in Socorro, New Mexico, and Long Valley, California," Geophys. J. Int. **146**, 191–200 (2001).