Multiresolution Network Flow Phase Unwrapping

G.F. CARBALLO
Centro de Cálculo, Fac. Ingeniería
Montevideo, 11300, Uruguay

P.W. FIEGUTH
University of Waterloo
Waterloo, Ont., N2L-3G1, Canada

gabriel@fing.edu.uy Tel: (598-2)711-4229 Fax: (598-2)711-5446

ABSTRACT

Phase Unwrapping is a key step in deriving topographic information from Interferometric SAR. Several recent methods are based on a minimum cost network flow approach. In this paper we develop a new, computationally efficient, "divide-and-conquer" strategy for phase unwrapping.

INTRODUCTION

The well-studied Interferometric Synthetic Aperture Radar (InSAR) problem for DEM generation involves the derivation of topographic information from radar phase. The topography is proportional to the full phase, whereas the *measured* phase is modulo 2π, necessitating the process of recovering full phase values via phase unwrapping. In general, the presence of noise, phase discontinuities, and the sheer size of the problem make phase-unwrapping challenging.

Our research is motivated by recent work[1, 2] which models the phase unwrapping problem as a discrete optimization problem. We define $\phi(i,j)$ and $\psi(i,j)$ as the unwrapped and wrapped phase functions respectively, where the indices i, j live in a rectangular $M \times N$ grid. Our measured phase obeys

$$\psi(i,j) = \mathcal{W}(\phi(i,j)) = \phi(i,j) + 2\pi n(i,j) \qquad (1)$$

where $n(i,j)$ are integers such that $-\pi < \psi(i,j) \leq \pi$ and $\mathcal{W}$ is the wrapping operator. We define the residuals

$$k_q = k_{i,j,d} = \frac{1}{2\pi}\left[\Delta_d \phi(i,j) - \mathcal{W}(\Delta_d \psi(i,j))\right] \qquad (2)$$

for each individual arc $q = \{i,j,d\}$, where Δ_d is the discrete difference operator along direction $d \in \{x, y\}$. If we let $\psi_q = \psi_{i,j,d} = \mathcal{W}(\Delta_d \psi(i,j))$, then the phase unwrapping problem can be formulated as

$$\min \sum_q c_q |k_q| \qquad (3)$$

such that all simple loop integrals be zero:

$$k_a + k_b + k_c + k_d = -\frac{1}{2\pi}[\psi_a + \psi_b + \psi_c + \psi_d] \qquad (4)$$

[0] **Funding acknowledgment:** Research supported by the Canadian Centre for Remote Sensing, the Canadian International Development Agency, and the Canadian Natural Science & Engineering Research Council.

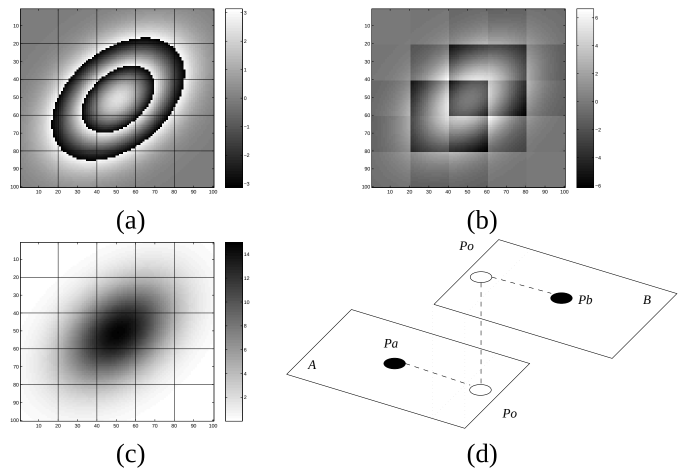


Figure 1: Example of a Divide & Conquer approach applied to phase unwrapping. An interferogram of a simple 2-D gaussian surface is partitioned (a) into squared blocks. Each block is unwrapped independently (b) and then these partial results are unwrapped among themselves to produce the final surface (c). The process of unwrapping two neighboring blocks is shown in (d).

where the c_q are nonnegative real numbers weighting the *a priori* confidence on the residuals. Equation (4) embodies the essence of network flow, stating that a surface will be correctly unwrapped if any closed loop integral in it is zero.

In this paper we will concentrate on algorithm performance issues. Available minimum cost flow implementations, like RELAX-IV[4], are known for their substantial resource needs[1], requiring gigabytes of RAM in order to process a moderately sized interferogram on the order of a million pixels [1, 2]. The only attempt to decrease the computational effort has been by Eineder et al [2]. Our particular research interest is the development of computationally efficient algorithms, applicable to huge problems ($\gg$ 10 million pixels).

We have developed a multiresolution phase unwrapping algorithm, based on network flow, which eliminates previous limitations of computing time or memory usage.[1] Ours is the first algorithm that has the ability to detect failures and correct them without human intervention.

[1] Although our development is based on linear network flow, the reader may find it useful to apply these ideas to other phase unwrapping algorithms.

MULTIRESOLUTION NETWORK FLOW

Divide-and-conquer

Our proposed strategy is to address the unwrapping problem by "divide-and-conquer"; that is, to successively decompose the problem until it becomes easy to solve and to recombine the elementary solutions, as illustrated in Figure 1:

1. The input interferogram is partitioned in blocks, as shown in Figure 1(a) and each block is unwrapped independently of each other.

2. Reconstruction of the individually unwrapped blocks, as shown in Figure 1(c), with respect to a reference point, common to the whole image.

By definition, the latter is an **unwrapping process** again, since there are elements (blocks) whose heights have been recovered ambiguously, due to the lack of a reference point with known absolute height, and they need to be restored to their original vertical location.

Consider the simplest case, provided by two adjacent blocks A, B with unwrapped phases ϕ_A and ϕ_B, as shown in Figure 1(d). If there is no information connecting the blocks the unwrapping problem is unsolvable. However, if the blocks overlap in a non-null set $\mathcal{O}$ (shown in dotted lines), we can use any point $P_o \in \mathcal{O}$ to unwrap; the final surface ϕ_S is obtained by mosaicking

$$\phi_S = \phi_A \bigcup (\phi_B + \Delta h), \quad \Delta h = \phi_A(P_o) - \phi_B(P_o). \quad (5)$$

The relative height difference on the final surface ϕ_S between any pair of points $P_a \in A$ and $P_b \in B$ can be computed through P_o:

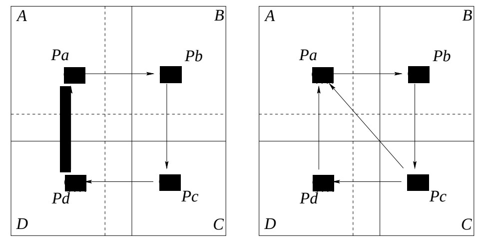
$$\begin{aligned} \Psi_{ab} &= \phi_S(P_b) - \phi_S(P_a) & (6)\\ &= [\phi_B + \Delta h](P_b) - \phi_A(P_a) & (7)\\ &= \phi_B(P_b) + [\phi_A(P_o) - \phi_B(P_o)] - \phi_A(P_a). & (8) \end{aligned}$$

It is significant to note that this "divide-and-conquer" process can be applied repeatedly, naturally leading to an unwrapping problem on multiple scales, applicable to problems of almost arbitrary size.

Generalized Network Flow

We now show how the individually unwrapped blocks can be unwrapped among themselves using network flow. The requirement is that every closed-loop path integral at the block level equal zero.

Consider four neighboring blocks A, B, C, D, as shown in Figure 2. Declare points P_a, P_b, P_c, P_d in each block to be "representatives" or **ICPs** (interferogram control point). We can define a "generalized residual" k_Q on arc Q, exactly as



(a) (b)

Figure 2: The loop integral computed along a closed path (connecting P_a, P_b, P_c and P_d) crossing four overlapping adjacent blocks must be zero to guarantee that the final result is a surface. The overlapping areas are determined by the spaces between the dashed and solid lines.

in equation (2), such that the corrected height difference is $\Psi_Q + 2\pi k_Q$:

$$k_{ab}+k_{bc}+k_{cd}+k_{da} = -\frac{1}{2\pi}\left[\Psi_{ab} + \Psi_{bc} + \Psi_{cd} + \Psi_{da}\right] \quad (9)$$

which is identical to equation (4).

We can do this at the block level because k_Q is guaranteed to be integer, since the unwrapped phase in each block has an error of an integer number of cycles. Any set $\{k_Q\}$ that satisfies equation (9) will produce feasible solutions. Clearly we want to select the one that minimizes the impact of the residuals, having the smallest (weighted) norm, exactly as we did in equation (3):

$$\min \sum_Q c_Q |k_Q| \quad (10)$$

where the costs c_Q now reflect the *a priori* confidence on the residuals. Although we are unwrapping at the block level, the network flow unwrapping problem translates exactly.

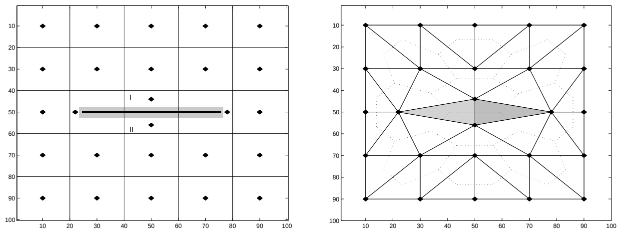
Arbitrary Topology Network Flow

If discontinuities are present, inconsistencies may arise within the overlapping areas. We propose to split the blocks into smaller regions, which can be conceived as **connected sets of pixels within a block that we consider to be** ***reliably*** **unwrapped**. These regions can assume arbitrary shapes; fortunately network theory naturally accomodates irregular configurations of nodes and arcs, arbitrary topologies being the rule and not the exception.

Consider the example shown in Figure 2(b). With the same configuration of ICPs as in (a), we can select different closed loop paths such as triangles. We then rewrite the constraints (9) as

$$k_{ab} + k_{bc} + k_{ca} = -\frac{1}{2\pi}\left[\Psi_{ab} + \Psi_{bc} + \Psi_{ca}\right]. \quad (11)$$

The objective function $\sum_Q c_Q |k_Q|$ in equation (10) needs to be modified to accomodate the new residuals.



(a) (b)

Figure 3: Splitting of center block into regions (labeled I and II) separated by a discontinuity. The black diamonds are the ICPs. The Delaunay triangulation is shown in solid lines (b). The arcs (dotted segments) connect neighbouring triangles. The shaded triangles indicate that the loop integrals are non-zero.

Transforming this linear programming problem into network flow is a bit more complicated than the regular-topology case. Figure 3(a) shows a topographic discontinuity spreading across several blocks. The center block is split into two regions, labeled I and II. Starting with a cloud of ICPs we build a Delaunay triangulation, shown in Figure 3(b), with the arcs of the generalized network shown in dotted lines. A "generalized charge dipole" (non-zero loop integral) is created within the shaded triangles as a result of the discontinuity.

REGIONS AND RELIABILITY

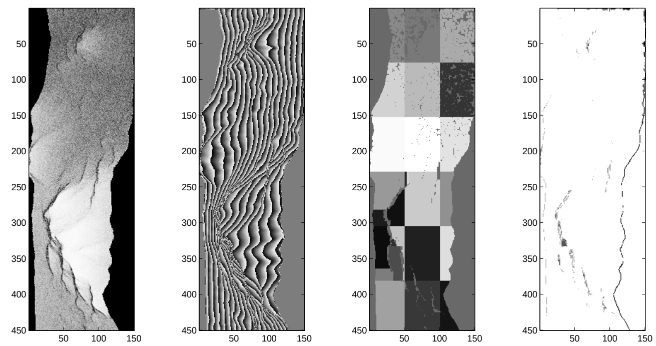
The key idea to properly define usable regions is to utilize redundancy; we apply the same algorithm with different input data, such that varied charge configurations offer different challenges and opportunities to find a correct solution. Our regions are then defined as those connected components, within the overlapping areas, which do not contain low-cost areas, and for which the multiple unwrapped surfaces are identical.

This definition is fault-tolerant: information is cross-checked for consistency over redundant areas, and whenever errors are detected, they are utilized to further split the regions, increasing the probability that each one of them is correctly unwrapped.

The ICP-to-ICP costs c_Q are still *ad-hoc*; our current rules have been designed such that inter-block arcs have high costs while intra-block arcs have low costs.

RESULTS

Our algorithm has been successfully tested with several datasets, some synthetic and some real SAR interferograms. Figure 4 shows one example (extracted from [3]) which compares strongly to current state-of-the-art phase unwrapping algorithms [3]. This particular result has been computed with



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

Figure 4: Synthetic example extracted from [3]. The coherence map is shown in (a), where the colormap runs from zero being black to white corresponding to one. The interferogram is shown in (b). The regions map is depicted with a random gray colormap in (c). The intensity-coded difference between our method and the original surface is shown in (d).

5×2 overlapping blocks and the threshold for determining low-coherence areas has been set to $t = 0.3$.

From an efficiency point of view, the solution can be computed sequentially, demanding a small amount of resources at each subproblem. Also, if execution time is constrained, the blocks can be unwrapped simultaneously, by distributing them into several processors.

This paper has described and illustrated a multiresolution methodology for phase unwrapping using network flow. It has originated from the confluence of different technologies and strategies, that converged into a single robust algorithm, reliable, efficient and cappable of unwrapping images with unlimited size. Also, it generated a favorable practical framework to introduce in the future scale-dependent models that describe terrain.

REFERENCES

[1] M. Costantini, "A novel phase unwrapping method based on Network Programming," *IEEE T. GRS*, V. 36, #3, 1998.

[2] M. Eineder, M. Hubig, B. Milcke, "Unwrapping large interferograms using the minimum cost flow algorithm," *Proc. IGARSS '98*.

[3] D. Ghiglia, M. Pritt, *Two-Dimensional Phase Unwrapping: Theory, Algorithms, and Software*, Wiley, New York, 1998.

[4] D. Bertsekas, P. Tseng, "The relax codes for linear minimum cost network flow problems," *Annals of Operations Research*, V. 13, 1988