Very Fast Absolute Phase Estimation: G-PEARLS & I-PEARLS

Sharoze Ali [#1], Habibullah Khan [2], IdrisH Shaik [3], Firoz Ali [4]

[1] Research Scholar , Department of ECE ,KL University,
Vaddeswaram, Guntur, Andhra Pradesh, India.
[2] Professor & Dean (SA) , Department of ECE ,KL University,
Vaddeswaram, Guntur, Andhra Pradesh, India.
[3] Assistant Professor , Department of ECE ,Bapatla Engineering College,
Bapatla, Andhra Pradesh, India.
[4] Head of Department, Department of EEE, Nimra College of Engineering & Technology,
Vijayawada, Andhra Pradesh, India.
[1] sharu786786@gmail.com
[2] habibulla@kluniversity.in
[3] idrishshaik@gmail.com
[4] firozalimd1@gmail.com

Abstract—Phase unwrapping (PU) is the process of recovering the absolute phase ϕ from the wrapped phase ψ and it is key problem in interferometry SAR processing (In-SAR) and Sonar (In-SAS), MRI (medical imaging) & signal processing applications, etc. In all these applications the observation mechanism is an absolute phase (2π-periodic function of the true phase). PEARLS is an algorithm which attain absolute phase using two steps: the first step is to de-noise the wrapped phase by using the LPA-ICI algorithm and the next to unwrap phase by using global phase unwrapping algorithm. PEARLS algorithm is able to reconstruct any noisy phase with high resolution. However, computation speed and memory consumption often times limits the effective use of PEARLS especially for large images. To overcome this drawback we have proposed two faster algorithms. The first one is I-PEARLS, which reduce the complexity of the PEARLS algorithm to attain speed and the other is the G-PEARLS which uses the cache efficient techniques while unwrapping the phase. Proposed algorithms are described in detail and tested on simulated and real images (interferometry SAR data). Results show that the proposed algorithm works faster and suggested alternative to the PEARLS algorithm.

Keyword- Absolute Phase Estimation, PEARLS, Interferometry SAR, Fast PU, Graph Cuts.

I. INTRODUCTION

Estimation of an absolute (true,ϕ) phase from the measured phase (wrapped, principle, ψ) is a key problem for many imaging techniques. Phase unwrapping (PU) [1] is one of the well known and widely adopted techniques for phase estimation. For instance, in remote sensing applications [2] like Synthetic aperture radar (SAR) or Sonar (SAS), phase difference between the terrain and the radar is captured by two or more antennas. The measured phase by SAR or SAS is in the interval of $[-\pi, \pi]$. By using the Phase unwrapping process, we can obtain the absolute phases from the measured one (ψ).

In MRI (Magnetic Resonance imaging), PU technique is used to determine magnetic field deviation maps, chemical shift based thermometry, and to implement BOLD contrast based venography. PU also acts as a necessary tool for the three-point Dixon water and fat separation. In optical interferometry, phase measurements are used to detect objects shape, deformation, and vibration.

In all the above discussed applications, observation mechanism is a 2π-periodic function of the true phase or absolute phase. The mapping of this function in the interval $[-\pi; \pi]$ yields the so-called principal phase values, or wrapped phases, or interferogram; if the true phase is outside the interval [-π; π] the associated observed value is wrapped into it, corresponding to the addition/subtraction of an integer number of 2π. It is thus impossible to unambiguously reconstruct the absolute phase; unless additional assumptions are introduced .This whole process is called Absolute Phase estimation.

During the last three decades, redundant of algorithms were proposed for phase unwrapping. We can summarize them into two types: A) two step method including denoising and unwrapping and B) unwrapping along with denoising.

A. Two steps method including denoising and unwrapping

According to the first method, the phase is firstly filtered and later on it is unwrapped by the PU Algorithms. The key advantage of this method is the denoising step. The idea behind this approach is if the high quality of wrapped phase map can be achieved, then the phase unwrapping can be accomplished by a simple unwrapping algorithm. So the main problem turns to the denoising of the wrapped phase images. It has been tackled by lots of researchers and many wrapped phase denoising methods were proposed. Some methods propose the window filtering techniques [3], [4] to denoise the phase. Some other uses the local filter techniques [5], [6] to denoise the phase. But PEARLS [7] is a wrapped phase denoising method which smartly selects the window size adaptively according to the smoothness of the profile.

B. Unwrapping along with denoising.

A set of the unwrapping methods were proposed during the past ten years [8]–[12] to do unwrapping along with denoising. A large amount of them do the denoising and unwrapping simultaneously, and there are also algorithms which first performs the unwrapping of phase and then post processed by a denoising method (e.g., [13]). A fourth-order polynomial approximation is proposed to do denoising in [13], but it is more sensitive to phase discontinuities because of the fixed window size. Similarly, in this method [14] second order local polynomial approximation is proposed to suppress the noise of unwrapped phase.

By using the approach-A, we can achieve better quality image ,but we have to compromise with runtimes of attaining the results and it is vice-versa with approach -B .So, in order to satisfy the both extremities, here in we propose two algorithms named as I-PEARLS and G-PEARLS, which provides faster runtimes than PEARLS algorithm. I-PEARLS is a algorithm which reduce the runtimes of PEARLS by reducing the complexity of the algorithm, where as G-PEARLS is an algorithm which uses the optimization techniques to achieve faster runtimes .In both methods, denoising is performed firstly based on the LPA-ICI algorithm proposed in [7], and then unwrap the phase by using the fast unwrapping algorithms [15],[16].

The significant advantage of the proposed methods is that we can unwrap the phase easily by using the fast unwrapping algorithms and its speediness is more pronounced for larger images. So the proposed methods are more intelligent, faster and can compromise between protecting the details and preventing the noise. In this research, we use the low complex I-PUMA Algorithm and cache efficient techniques to unwrap the phase .From the experiments in this letter our proposed methods can achieve faster runtimes than other unwrapping methods. The remaining of the letter is organized as follows. Section II presents the PEARLS algorithm; in Section III, we present the new global unwrapping methods I-PUMA and G-PUMA; the new algorithms I-PEARLS and G-PEARLS methods are proposed in Section IV; Section V presents a set of experiments and results to compare with other algorithms .we conclude this letter in Section VI.

II. Phase Estimation using Adaptive Regularization based on Local Smoothing

Phase unwrapping (PU) is the process of recovering the absolute phase from the wrapped phase, formally as

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

In most of the SAR and MRI applications, the observation mechanism is a 2π -periodic function of the true phase, also known as absolute phase. It is thus impossible to unambiguously reconstruct the absolute phase, unless additional assumptions are introduced into this inference problem.

PEARLS (for Phase estimation using adaptive regularization based on local smoothing) is an algorithm to reconstruct the absolute phase from the wrapped phase. PEARLS algorithm consists of two steps, the first step is to pre-filter the noisy wrapped data using the LPA-ICI algorithm and the second step is to unwrap phase by using PUMA Algorithm. So PEARLS algorithm is a combination of LPA+ICI+PUMA techniques.

Let us assume that the absolute phase is a piecewise smooth function, which is well approximated by a polynomial in a neighbourhood of the estimation point. Besides the wrapped phase, the size and possibly the shape of this neighbourhood are estimated. The approach of filtering is first to design the non linear filters (estimators) by using the local polynomial approximation (LPA) and then to adapt these filters to unknown smoothness of the spatially varying absolute phase where ICI is used for adaptive selection of the size (scale) of the window. The adaptiveness introduced by ICI trades bias with variance in such a way that the window size stretches in areas where the underlying true phase is smooth and shrinks at discontinuities.

As shown in Figure 1, the phase ϕ is given by

$$\phi = angle(z) = \mathcal{W}(\varphi + \phi_n), \quad \phi \in [-\pi, \pi) \tag{2}$$

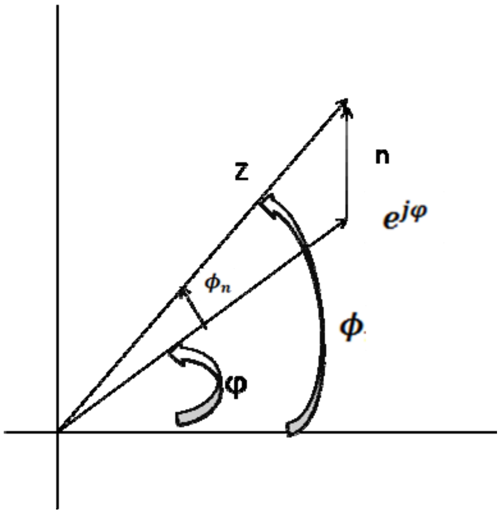


Fig.1. Illustration of the observed phase model: φ is the true phase, ϕ is the observed phase, and ϕ_n is the phase component of ϕ due to noise vector n.

Where ϕ is the true phase, ϕ_n is the phase component of ϕ due to noise vector 'n' and $\mathcal{W}$ is the wrapping operator mapping the noisy phase $\varphi + \phi_n$ into the basic phase interval $[-\pi, \pi)$.For the input noisy phase, the zero-order approximation is used to infer the window sizes and the first-order approximation to compute the wrapped phase estimates on to the windows determined by the zero-order approximation. Assume that the phase $\hat{\varphi}_h$ is well estimated and nearly approximate to the phase φ

Where $\hat{\varphi}_h(x, y)$ is in equation (3)

$$\hat{\varphi}_h(x, y) = angle\, [\sum_s w_{h,s} * Z_\phi(x + x_s, y + y_s) * e^{-j(c_2 x_s + c_3 y_s)}] \tag{3}$$

The zero –order phase estimates of $\hat{\varphi}_h(x, y)$ are calculated as below

$$\hat{\varphi}_h(x, y) = \hat{c}_1(x, y) = angle[\mathrm{F}_h(0,0)] \tag{4}$$

Where $\mathrm{F}_h(c_2, c_3)$ is the windowed discrete Fourier transform of the normalized data $Z_\phi = {}^{Z}/_{|Z|}$ at point (x, y) and frequency(c_2, c_3).If $c_2 = 0, c_3 = 0$ then

$$\hat{\varphi}_h(x, y) = angle[\mathrm{F}_h(0,0)] = c'_1 + angle[W_h(-c'_2, c'_3)] \tag{5}$$

Once the zero order estimates are calculated as shown in equation (5), then apply the ICI rule (explained as in below section) to these estimates $\hat{\varphi}_h(x, y)$ for the selection of the best window size $h^+(x, y)$.Using the equation(6) , calculate $\hat{\varphi}_{h+}$ the first-order phase estimates with adaptive window size

$$\hat{\varphi}_h(x, y) = angle[\mathrm{F}_h(\hat{c_2}, \hat{c_3})] = c'_1 \tag{6}$$

Once the $\hat{\varphi}_{h+}$ is calculated by using the above process, then unwrap phase using the PUMA [8] Algorithm.

A. ICI Rule

For small noise level and small 'h', the phase estimate $\hat{\varphi}_h(x, y)$ is unbiased with the variance

$$\sigma_h^2 = \sigma^2 \frac{\sum_s w_{\mathrm{h,s}}^2}{(\sum_s w_{\mathrm{h,s}})^2} \tag{7}$$

This result is used for the adaptive selection of the window size. Confidence intervals of $\hat{\varphi}_h(x, y)$ estimates are defined as below

$$Q_h = \{\hat{\varphi}_h - \Gamma.\sigma_h\, , \hat{\varphi}_h + \Gamma.\sigma_h\} \tag{8}$$

Where $\Gamma > 0$ is a parameter of the algorithm. To identify this adaptive h^+ the successive intersection of the confidence intervals Q_h is considered starting from Q_{h1} and Q_{h2} . Specifically, the pairwise intersection of the intervals Q_{h_j}, $1 \le h_j \le h_i$ is considered with increasing h_i.Let h^+ be the largest of those h_i for which the intervals Q_{h_j}, $1 \le h_j \le h_i$, have a point in common. This h^+ defines the adaptive window size and the adaptive estimate as $\hat{\varphi}_{h+}$ .For the varying point wise adaptive estimation, these calculations are produced for all points (pixels). In implementation, the ICI algorithm is used when the estimates for all points (x, y) are already calculated for all h. Then the algorithm works as a selector of the proper window size estimate for each point from a given set of the estimates for all window sizes.

III. New Global Unwrapping Methods: I-PUMA and G-PUMA

In this section, we present our recently developed algorithms I-PUMA and G-PUMA, before that lets have a look on the disadvantages of the PUMA. PUMA is a two-step phase unwrapping process. Firstly, the elementary graphs are constructed for a site by using the energy equation ψ and secondly, it is minimized by the graph cut optimization techniques. These two steps together are run for 'k' iterations to unwrap the phase. PUMA [8] is one of the novel technique which unwraps phase even at dis-continuities for both convex and non-convex potentials.

A. Dis-advantages of PUMA

The only disadvantage of PUMA algorithm is it consumes lot of time and memory to unwrap phase of larger images. This often time limits the utilization of PUMA algorithm. If we take a closer look at the algorithm, most of the time spent by the algorithm during the calculation of maxflow algorithm. As per our earlier research, in-order to overcome this difficulty and to increase the speed of PUMA Algorithm, we have to adopt either of the below two approaches.

1) To reduce the complexity of the PUMA Algorithm

2) To use the cache efficient techniques while calculating the max flow.

B. Phase Unwrapping via IBFS Graph cut: I-PUMA

In-order to overcome the difficulty and to faster the PUMA Algorithm, we have to reduce the complexity of the PUMA Algorithm. Computational complexity of the PUMA algorithm is $\mathrm{N_{bopt}} * \mathrm{N_{mf}}$ where $\mathrm{N_{bopt}}$ is number of binary optimization and $\mathrm{N_{mf}}$ stands for number of flops per max-flow computation .Authors of PUMA proved that the algorithm stops in k iterations where k is the number of 2π multiples and it is equal to $\mathrm{N_{bopt}}$. For $\mathrm{N_{mf}}$, authors used augmentation path type max flow and its worst case complexity $\mathrm{T(m,n)} = \mathrm{O(n^2m)}$. So, the complexity of PUMA is T (m, n) i.e. K times the worst case complexity of max flow algorithm. We have observed that in order to make PUMA faster and to reduce its complexity, we have to reduce the max flow algorithm's complexity, as its contribution in the complexity of PUMA is more. Different researchers have published different algorithms for max flow computation .Among them authors of IBFS [17] offers a faster and theoretically justified alternative to BK Algorithm. They have attained this by using the breadth-first search trees. So we utilize the advantage of the IBFS Graph cuts over BK and we used IBFS as an optimization technique in our algorithm. Like PUMA algorithm, I-PUMA algorithm also comprises of two steps .As a First step, we have to construct the elementary graphs by using the energy equations. Later on the minimization of energy equation with respect to δ is now mapped onto a max-flow problem .For construction of graph we have followed the approach of PUMA, where the vertices and edges corresponding to each pair of neighboring pixels are build first, and then join these elementary graphs together based on the additive theorem. Once the energy is represented onto the graph, we have used the IBFS max flow algorithm for minimizing the equation.

We have also make use of full compiler optimizations technique (/Ox in Visual Studio) while compiling the Mex. We have attained 20-40 % faster in run times .We has also observed that I-PUMA and PUMA have nearly equal error norm. For some profiles I-PUMA run for some more iteration's so that further energy is minimized. As PUMA, I-PUMA also deal well with discontinuity and unwraps the phase faster than PUMA.

C. Very fast Phase Unwrapping via Grid Cuts :G-PUMA

In our previous approach, I-PUMA we have followed the first approach of reducing the complexity of the PUMA Algorithm. But the runtimes can be further improved especially while un-wrapping phase of larger images. When we deeply analyze the PUMA algorithm for further improvement, especially the optimization part, we have highlighted below some issues faced by both of the algorithms PUMA and I-PUMA.

The run times will be very faster if we are able to solve these issues. The first issue, i.e. stress on memory bandwidth is resolved by employing a compact graph representation with cache-friendly memory layout that exploits the regular structure of grid-like graphs which reduces the stress on the memory bandwidth. Secondly (calculation of connectivity information), by exploiting prior knowledge of the graph structure we can eliminate the need of pointers altogether by determining connectivity information on the fly. Thirdly, Cache can be efficiently utilized by segregating the memory fields as Hot and Cold depending upon their accessibility (Structure splitting) and by dividing the given graph into blocks (Blocked array layout). As per our second approach, we are able to solve the above these stated issues and we name it as "G-PUMA". Among the available, the best cache efficient graph cut technique is Grid Cut. We name our new approach as G –PUMA where 'G' represents the Gridcut technique.

Like PUMA algorithm, G-PUMA algorithm also comprises of two steps. In G-PUMA the sequences of steps to unwrap the phase for both convex and non-convex will be remain same. As a First step, we have to construct the elementary graphs by using the energy equations. The power of PUMA lies in its energy minimization framework and shows greater attenuation to noise, so we have not modify or changed the energy equation ψ of PUMA. Once the elementary graph is constructed, the minimization of energy equation with respect to is now mapped onto a max-flow problem. In order to use the same graphs of PUMA on Gridcut, either we have to construct the full grid or we have to set zero capacity for edges which are not provided by PUMA and treat them as dummy nodes. If we follow the second approach of padding, then the performance of algorithm is degraded [18]. So we have adopted the first option of constructing the full grid. The grid can be fully constructed from either of the below two options.

1) To calculate the other pixel interactions which are not provided by PUMA. (or)

2) To reuse the same graph for calculating the pixel interactions which are not provided by the PUMA. Purge the missing Pixel interactions from its 1- offset neighbour and pad the edges with Zeros.

We have adopted the reuse approach to construct the full grid as the first approach is more time consuming. Once we constructed the elements of the graph, we have added them based on the additive theorem and provided as input to the min cut solver. In our algorithm, we have avoided passing supplementary information, as the same can be extrapolated during the max flow computation. We have also make use of full compiler optimizations technique (/Ox in Visual Studio) while compiling the Mex. We have attained 40-75% faster in run times for both 32-bit and 64-bit machines. Also our new algorithm consumes less memory than PUMA algorithm of about 20-40% for 32-bit and 40-60% for 64-bit machines.

Finally, I-PUMA algorithm reduces the complexity of the PUMA algorithm where as the G-PUMA algorithm achieves the faster runtimes by using the complexity. But runtimes of I-PUMA algorithm are not as fast as G-PUMA where as G-PUMA can be applied to the grid only. So both the algorithms works as an alternative to PUMA algorithm but have their own limitations.

IV. Very Fast Absolute Phase Estimation: G-PEARLS and I-PEARLS

At a high level, PEARLS is a two-step process. Firstly, the noise of the wrapped phase is removed and later on the denoised phase is provided as input to the Phase unwrapping algorithm to unwrap the phase. PEARLS [7] is one of the novel technique which unwraps phase even at dis-continuities. The only disadvantage of PEARLS algorithm is it consumes lot of time and memory to unwrap phase of larger images. This often time limits the utilization of PEARLS algorithm. In-order to overcome the difficulty and to increase the speed of PEARLS algorithm, we have to adopt either of the below two approaches.

1) To reduce the complexity of the PEARLS Algorithm
2) To use the cache efficient PU technique.

A. *I-PEARLS: Fast absolute phase estimation using the I-PUMA algorithm*

The major complexity of the PEARLS algorithm is at computing the adaptive window size and at PUMA phase unwrapping algorithm. For FFT's of size L^2, $(2L^2\log_2 L)*n$ is the computational complexity where 'n' is the number of image pixels. O $(n^{2.5})$ is the complexity of the PUMA phase unwrapping algorithm. So, most of the complexity of the algorithm is from the Phase unwrapping technique. I-PEARLS is an approach where the complexity of the PEARLS algorithm is reduced .In I-PEARLS we utilize the low-complexity I-PUMA Phase unwrapping algorithm to find the solution. We have reduced the complexity of the PU algorithm to $O(n^2 m)$ where PUMA complexity is more than $O(n^2 m)$. By doing so, we were successfully being able to reduce the complexity of PUMA algorithm and the runtimes were 15 to 25% faster than PEARLS.

Like PEARLS, I-PEARLS algorithm also comprises of two steps .As a First step, we have denoised the wrapped phase and later on the wrapped phase is unwrapped using the I-PUMA algorithm. Also we have analyse the coding parts of PEARLS and we have identified that there are some unnecessary calls between various programming parts of PUMA and by properly tunning it we can make the algorithm bit faster. As per [17] the run times will be faster if we use the optimization techniques while compiling the matlab mex's.

So, we have also make use of full compiler optimizations technique (/Ox in Visual Studio) while compiling the Mex. We have in co-operated all these above mentioned changes in PEARLS and from the experiments in Section VI, We have attained 15-25 % faster in run times .We have also observed that I-PEARLS and PEARLS have nearly equal error norm . For some profiles I-PEARLS ran for some more iteration's so that further energy is minimized. As PEARLS, I-PEARLS also deal well with discontinuity and unwraps the phase faster than PEARLS.

B. *G-PEARLS: Fast absolute phase estimation using the G-PUMA algorithm*

G-PEARLS is an approach to increase the runtimes and to reduce the stress on the memory bandwidth of the PEARLS algorithm. We have deeply analyse the PEARLS algorithm for further improvement, especially the optimization part, the below issues were faced by both of the algorithms PEARLS and I- PEARLS.

1) During the computation of the max flow algorithm, there will be frequent transfer of data between memory and CPU .Also constant updates are required, which causes heavy stress on the memory bandwidth.
2) More time consumed while finding the connectivity information using pointers on every time. The Pointers are generally used to represent the general graphs and it provides the connectivity information. These pointers often comprise the majority of the graph's memory footprint, in particular on 64-bit CPUs where single pointer occupies eight bytes.
3) Cache is poorly handled.

The run times will be very faster if we are able to solve the above three issues. As per our second approach G- PEARLS, we are able to solve the above three stated issues where we have employed the cache efficient techniques during the optimization step of PUMA. In this section, we introduce our new, fast and robust unwrapping algorithm "G-PUMA".

All the above three mentioned issues will play a crucial role in finding min cut of large graphs. The first issue (stress on memory bandwidth) is resolved by employing a compact graph representation with cache-friendly memory layout that exploits the regular structure of grid-like graphs which reduces the stress on the memory bandwidth. Secondly (calculation of connectivity information), by exploiting the prior knowledge of the graph structure we can eliminate the need of pointers altogether by determining connectivity information on the fly. Thirdly, Cache can be efficiently utilized by segregating the memory fields as Hot and Cold depending upon their accessibility (Structure splitting) and by dividing the given graph into blocks (Blocked array layout).

Among the available, the best cache efficient graph cut technique is Grid Cut. So we utilize Grid cut as an optimization technique and we name our new approach as G–PEARLS where 'G' represents the Gridcut technique.

Like PEARLS algorithm, G-PEARLS algorithm also comprises of twosteps. As a First step, we have to de-noise the wrapped phase by using the LPA-ICI techniques. The power of PEARLS lies in adaptive LPA-ICI algorithm which locally adapts the window according to smoothness of the profile and shows greater attenuation to noise, so we have not modify or changed the denoising step of PEARLS. Once the phase is denoised, the phase is unwrapped using the G-PUMA algorithm.

The main limitation of G-PEARLS optimization technique is hidden in the fact that it supports only graphs which has grid-like topology. Grid like topology is a graph topology, where nodes can be stored in memory in such a way that neighbours can be addressed by a set of fixed offsets. The graphs constructed by the G-PEARLS will not have all pixel interactions required to represent a grid-like topology.

We have analyse the coding parts of PEARLS and we have identified that there are some unnecessary calls between various programming parts of PEARLS and by properly tunning it we can make the algorithm bit faster. For Instance, in PEARLS along with the edge weights, also sending the other supplementary information related to the pixel interactions like node number of the starting pixel and ending pixel, etc to maxflow mex-part of coding. In our algorithm, we have avoided passing such supplementary information, as the same can be extrapolated during the max flow computation.

As per [18] the run times will be faster if we use the optimization techniques while compiling the matlab mex's. So, we have also make use of full compiler optimizations technique (/Ox in Visual Studio) while compiling the Mex. We have in-cooperated all these above mentioned changes in PEARLS and named it as G-PEARLS. From the experiments in Section V, We have attained 15-35% faster in run times. Also our new algorithm consumes less memory than PEARLS algorithm of about 20-40% for 32-bit and 40-60% for 64-bit machines. Like PEARLS, G-PEARLS also deal well with discontinuity and unwraps the phase faster than PEARLS.

V. EXPERIMENTAL RESULTS

The effectiveness of PEARLS algorithm is illustrated by conducting several experiments on continuous phase surfaces and discontinuous phase surfaces. To effectively compare our new algorithms G- PEARLS, I-PEARLS with PEARLS, we too conducted experiments on the same profiles used by PEARLS. Like PEARLS, LPA is also exploited with the uniform square windows w_h defined on the integer symmetric grid $\{(x, y): |x|, |y| \leq h\}$.The ICI parameter was set to $\Gamma = 2.0$, the window sizes to $H = \{1,2,3,4\}$ and the zero padded FFTs sizes to $L \times L = 64 \times 64$. Once the phase is unwrapped by LPA – ICI, then we use the new faster algorithms G-PUMA and I-PUMA.

Specification of computer in experimental is processor Intel Core Duo2 Processor, 32-bit, 4 GB RAM, 500 GB HDD, 2 MB Cache memory and LCD 15".

A. Continuous Phase Surfaces

1. Gaussian Surface

In this experiment, we consider continuous phase surfaces where continuous surface means the surface on which the phase differences between neighboring pairs of pixels are not larger than π in magnitude. Figure 3(a) shows a Gaussian shaped surface $\varphi\,(x, y)$ with $\sigma_x = 10$, $\sigma_y = 15$ and $A\varphi = 14\pi$, the maximum value of φ is 14π and the maximum values of the first differences are about 2.5 rad. As shown in Figure 3(b) is the LPA-ICI window size at each pixel and it adapts according to its smoothness.

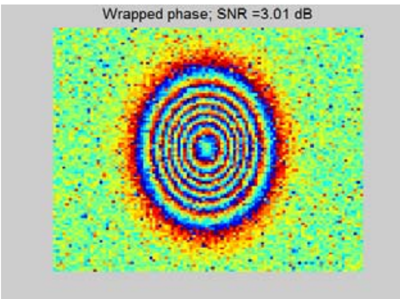
The noise standard deviation is set to σ = 0.5 corresponding to SNR = 3 dB. Such a noisy profile of high phase differences is provided as input to all the three algorithms (PEARLS, G-PEARLS, I-PEARLS).All these algorithms perform well and unwrap the phase correctly. The improvement in the SNR (ISNR) of the wrapped estimate [see Fig. 3(c)], which define as below.

$$ISNR = 10 \log 10 \frac{\left\| e^{j\phi} - e^{j\varphi} \right\|^2}{\| e^{j\widehat{\varphi}_{h+}} - e^{j\varphi} \|^2} \quad (9)$$

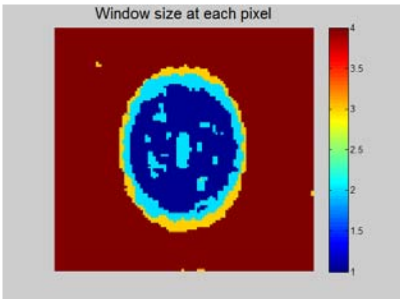
As mentioned in Section –III, PEARLS is a combination of LPA+ICI+PU techniques .The only change in the new G-PEARLS and I-PEARLS algorithms is in the phase unwrapping (PU) technique. As there is no change in the LPA and ICI techniques, the new algorithms also records same values of ISNR with respect to the denoised wrapped phase as listed in the Table I. The value of ISNR with respect to the un-wrapped phase (after the PU step) is bit low value for I-PEARLS than PEARLS where as for G-PEARLS is same as PEARLS as listed in Table I.

The reconstructed absolute phase estimate is displayed in figure 3(d),3(e),3(f) and root mean square error (RMSE) of reconstructed phase is RMSE=0.15 rad , i.e. equal for all the three algorithms .As shown in Table II , we have calculated the RMSE value for different values of standard deviation ‘σ’. The last column of Table II displays the time taken by each of the algorithm .There is just one column as the times depend very little on the noise variances. Notice the ICI ability to locally adapt the amount of smoothness: the larger windows are selected in areas where a first-order polynomial is a good approximation to the data and vice versa. We have verified it by varying the standard deviation of noise.

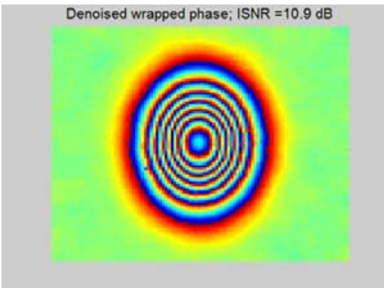
G-PEARLS is able to unwrap the phase very faster than PEARLS and the reconstruction error is same as PEARLS as shown in the table II. Memory optimization techniques allow us to unwrap the phase faster (i.e. within 2.29 seconds) than PEARLS (3.29 seconds). Also, I–PEARLS show faster run times (i.e. within 2.63 seconds) than PEARLS by reducing the complexity of the PEARLS algorithm. Gridcut PU technique provides faster run times even if the size of the profile increases and consumes less memory than PUMA. The memory consumption will be further reduced if we use 64–bits. But in this paper, we will limit our discussion to runtimes of the algorithm.



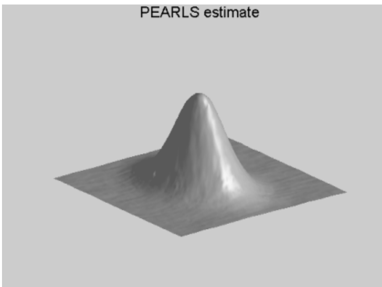
(a)



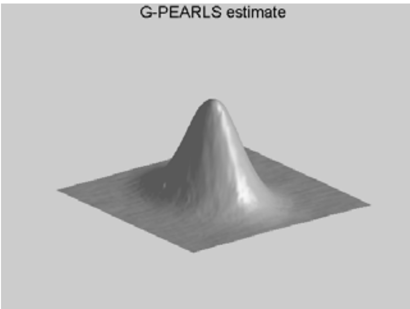
(b)



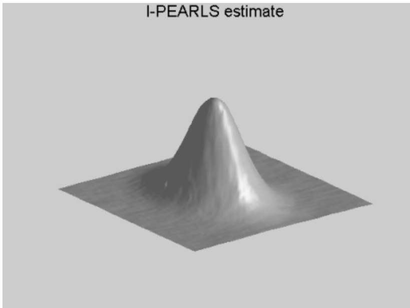
(c)



(d)



(e)



(f)

Fig. 3. (a) Gaussian shaped surface with σ=0.5 corresponding to SNR=3 dB (b) Window provided by LPA-ICI(c) Denoised Gaussian Phase (d) 3-D reconstruction of PEARLS (e) 3-D reconstruction of G-PEARLS (f) 3-D reconstruction of I-PEARLS .

TABLE I. PSNR and MSE Results

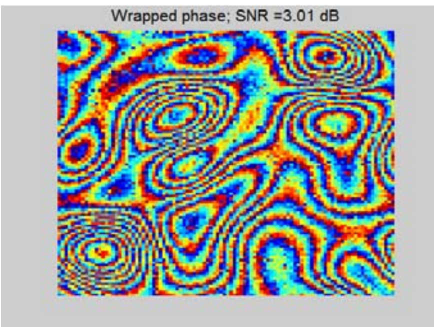
I.S.N.R/Phase	PEARLS	G-PEARLS	I-PEARLS
w.r.t Wrapped Phase	10.93	10.93	10.93
w.r.t Unwrapped Phase	12.63	12.63	11.58

TABLE II. RMSE Results

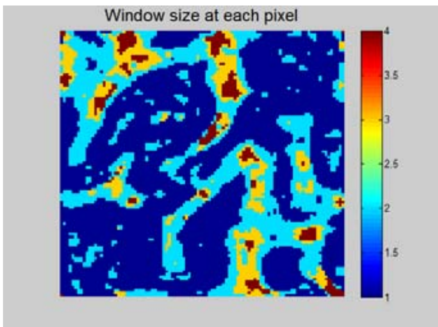
Algorithm	σ =0.75	σ =0.5	σ =0.01	Time (sec)
PEARLS	0.33324	0.14352	0.035129	3.29
I-PEARLS	0.33324	0.14352	0.035129	2.29
G-PEARLS	0.33324	0.14352	0.035129	2.63

2. Random Elevation Surface

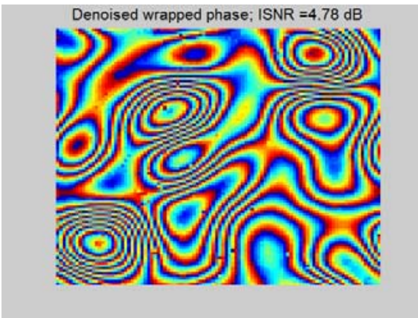
A noisy wrapped image is generated by low-pass filtering an image of independent and identically distributed Gaussian noise on a surface. The additive noise level is SNR = 3 dB and the magnitude of the first order differences is larger than π in many areas of the image. Figure 4(b) is the estimated window size provided by pre-filtering (LPA-ICI) where ICI is able to locally adapt the amount of smoothness. The improvement in SNR (ISNR) due to LPA-ICI is 4.8 dB and it is same for the new algorithms as well. Figure 4(c) is the de-noised wrapped image provided as the input to PU, instead of noisy profile as in figure 4(a). The improvement in SNR (ISNR) with respect to unwrapped phase to noisy wrapped phase is 13.03 dB where as the ISNR for I-PEARLS is 12.62 dB, bit low when compared to PEARLS. The reconstructed error (RMSE = 0.33 rad) is same for all the three algorithms. The new algorithm G-PEARLS unwraps phase within 2.48 sec, where as PEARLS finds the solution in 3.72 Sec. Similarly I-PEARLS find solution within 2.75 sec. Both the new algorithms show greater improvement in time (G-PEARLS of 33 % and I-PEARLS of 26%) and able to find the solution very faster than PEARLS.



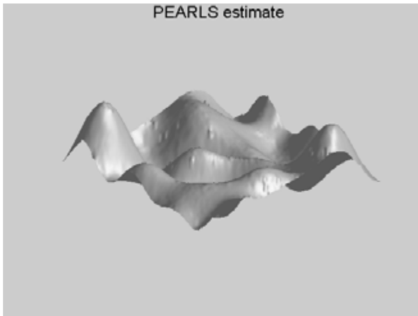
(a)



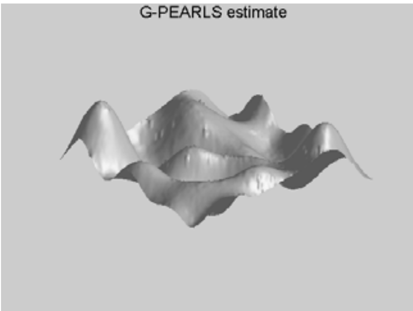
(b)



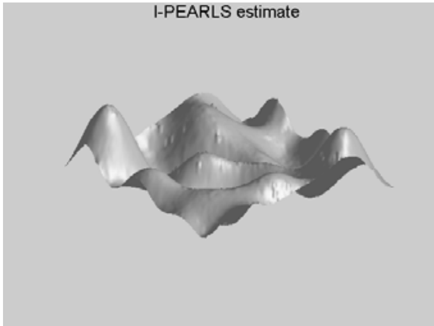
(c)



(d)



(e)



(f)

Fig. 4. (a) Random elevation surface with σ=0.5 corresponding to SNR=3 dB (b) Window provided by LPA-ICI(c) Denoised Random Elevation Phase (d) 3-D reconstruction of PEARLS (e) 3-D reconstruction of G-PEARLS (f) 3-D reconstruction of I-PEARLS .

TABLE III. RMSE Results

Algorithm/Method	**RMSE**	**time**
PEARLS	0.33555	3.72
I-PEARLS	0.33555	2.48
G-PEARLS	0.33555	2.75

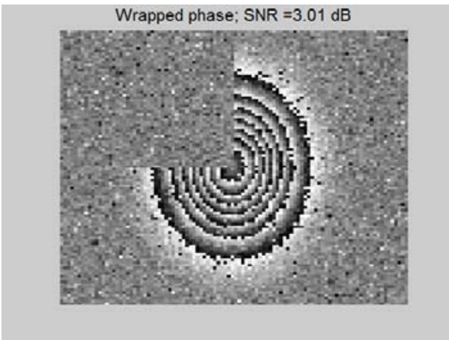
B. Discontinuous Phase Surfaces

1. Clipped Gaussian

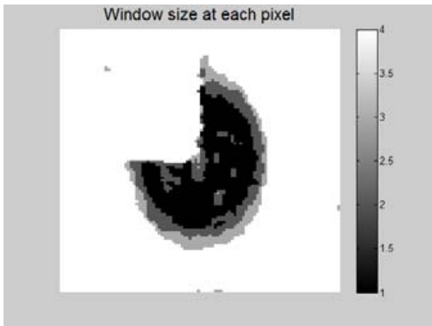
PEARLS is a combination of pre filtering and phase unwrapping techniques. Phase unwrapping methods are very sensitive to noise so it's better to pre filter the wrapped data. But the Pre filtering should not be so strong enough to filter the essential data especially of discontinuities. PEARLS algorithm is able to nicely adapt the window with respect with its smoothness so that it avoids chopping the sensitive data of discontinuous.

As we are using the same pre-filtering technique, so new algorithms should also be able effectively deal well with discontinuous. In order to check the effectiveness of new algorithm we conducted experiment on clipped Gaussian profile. Clipped Gaussian profile is a Gaussian surface as shown in Fig. 3(a) with one quarter of its set to zero. The noise standard deviation is set to σ = 0.5, corresponding to SNR = 3 dB. The de-noised wrapped phase shown in Fig. 5(c) is a clear improvement of the noisy version presented in Fig. 5(a).The window of pre-filtering technique (as shown in Fig 5(b))stretches in area where there are no discontinuities and shrinks at areas of discontinuities thereby ensuring that no data related to the discontinuities is washed out. Thereafter, the well known PUMA PU technique is able to deal well with discontinuities. Even the new algorithms G-PEARLS and I-PEARLS are able to unwrap the discontinuities well as shown in figure 5(e), 5(f).

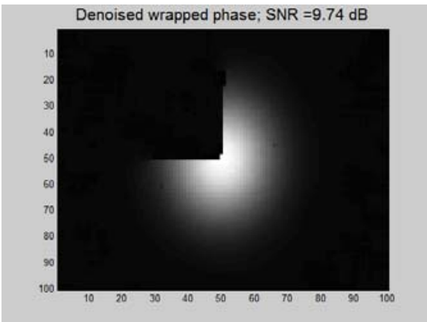
The importance and the usage of the de-noising step can be well explained if we ran the Phase unwrapping algorithm with and without filtered phases. As shown in figure 6(d), is the noisy wrapped phase of a standard deviation (σ = 0.75) where as 6(a) is the de-noised phase of 6(d).Both the wrapped phases are unwrapped by the both the new algorithms as shown in the figures 6(b),6(c) and 6(e),6(f). The de-noising step adds value and helps in unwrapping the phase. As shown in the figure 6(b),6(c) the phase is well estimated if we pre-filter the phase prior to unwrapping step where as the unwrapping totally fails for noisy wrapped phase as shown in figure 6(e),6(f).



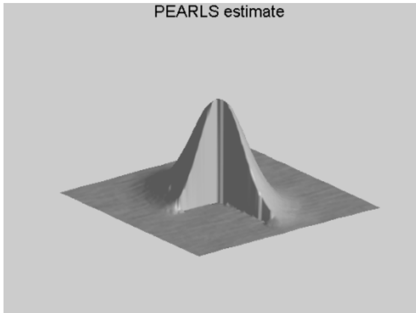
(a)



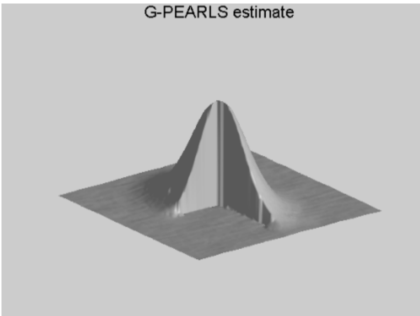
(b)



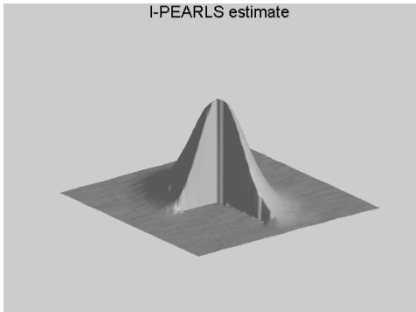
(c)



(d)

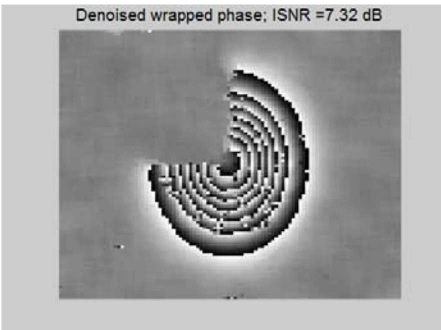


(e)

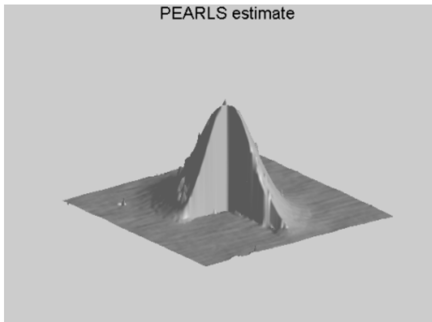


(f)

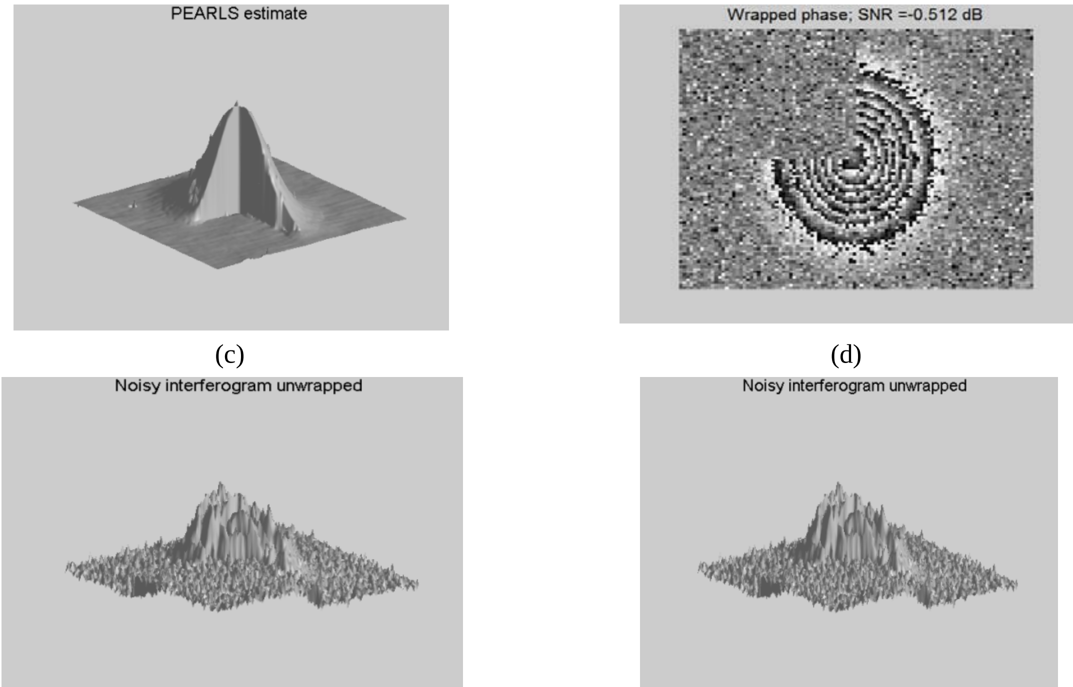
Fig. 5. (a) Clipped Gaussian profile with σ=0.5 corresponding to SNR=3 dB (b) Window provided by LPA-ICI(c) Denoised Clipped Gaussian profile (d) 3-D reconstruction of PEARLS (e) 3-D reconstruction of G-PEARLS (f) 3-D reconstruction of I-PEARLS .



(a)



(b)



(c) (d)

(e) (f)

Fig. 6. (a) Denoised Clipped Phase (b) 3-D reconstruction of G-PEARLS (c) 3-D reconstruction of I-PEARLS (d) Noisy Clipped Gaussian surface (e) 3-D reconstruction of G-PEARLS (f) 3-D reconstruction of I-PEARLS .

2. *Simulated SAR Data Based on a Real Surface with Quality Maps*

Fig 7(a) shows a phase image associated with noise (152 X 458 pixels) to be unwrapped. It was obtained from an original absolute phase surface that corresponds to a (simulated) In-SAR acquisition for a real steep-relief mountainous area inducing, therefore, many discontinuities and posing a very tough PU problem. This area corresponds to Long’s Peak, CO, and the data is distributed with book [1]. The wrapped image is generated according to an In-SAR observation statistics, producing an interferometric pair; by computing the product of one image of the pair by the complex conjugate of the other and finally taking the argument, the wrapped phase image is then obtained as in Fig 7(b). Fig. 7(c) shows a quality map (also distributed with book [1]) computed from the In-SAR coherence estimate (see [1, Ch.3]) for further details). The window estimates by LPA-ICI are as shown in figure 7(d). True phase of the area corresponds to Long’s Peak, CO is in Fig 7(e). The two flat regions in gray on the top and on the bottom of Fig. 7(b) correspond to undefined data due to the projection of the high terrain relief into the slant plane. Due to SAR layover phenomenon, fringes are formed in some regions of the interferogram and it is difficult to reconstruct the true phase from these fringes .Lets denote ‘X0’ the area corresponds to fringes as shown in figure (quality map).We have to extrapolate the phase from the area excluding the fringes (X-X0) where the area corresponding to X0 can be extrapolate from its neighbors (X-X0).

LPA-ICI technique is able to adapt the window according to the smoothness and provides the denoised phase as shown in figure 7(c). Later on the denoised phase, is unwrapped by the Phase unwrapping step. In the new algorithms, G-PEARLS and I-PEARLS are able to unwrap the phase faster than the PEARLS for the same error norm of 0.23 rad as shown in figure 7(f) and 7(g) .PEARLS finds the solution in 21 seconds where as the G-PEARLS are able to finds the solution within the 14.33 seconds with an improvement of 28% in runtimes .I – PEARLS finds the solution in 17.5 seconds leads to an improvement of 16% in runtimes than PEARLS. The error norm is nearly same for all the three algorithms.

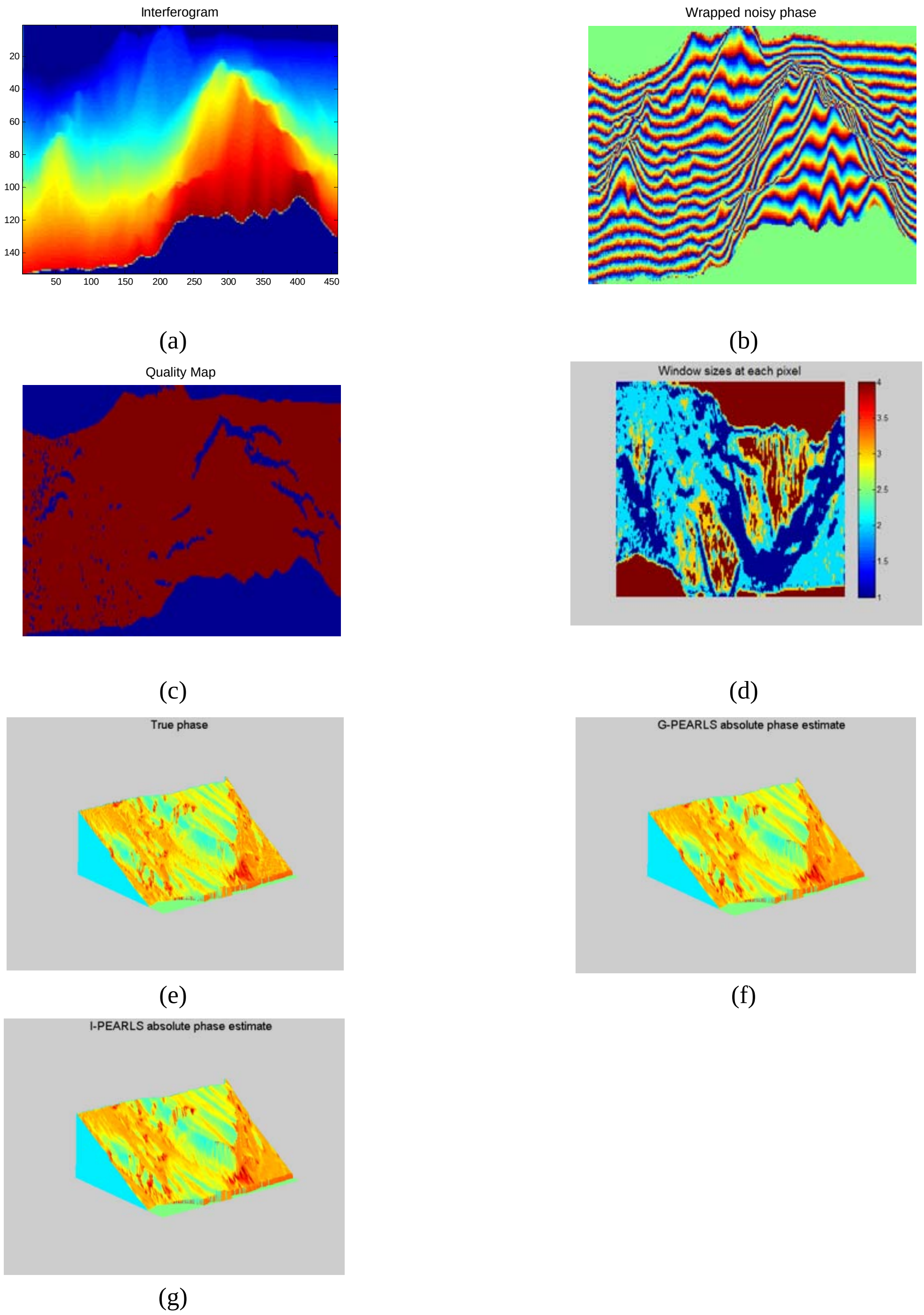


Fig. 7(a) Wrapped phase image obtained from a simulated InSAR acquisition from Long's Peak, CO (data distributed with [1]). (b)Wrapped Noise Phase (c) Quality Map of (a) distributed with [1]. (d) window estimates provided by LPA-ICI (e) True Image of (a). (f) 3-D reconstruction of G-PEARLS (g) 3-D reconstruction of I-PEARLS.

VI. Conclusion

Even though PEARLS is good at unwrapping the phase, but runtimes are affected due to the additional step of de-noising. The runtimes can be improved by either reducing the complexity of the PEARLS or by using cache efficient graph cut techniques. We have reduced the complexity of the PEARLS algorithm by utilizing the new low complex I-PUMA algorithm and we name it as I-PEARLS .Similarly we have followed other process of using cache efficient techniques to increase the speed of PEARLS and we name it as G-PEARLS .Both the algorithms showed greater speed of about 10% - 35% for the same attenuation to noise and are alternative to PEARLS.

ACKNOWLEDGMENT

The authors would like to express great thanks for the help from Professor J. M. Bioucas-Dias [7] in the Instituto de Telecomunicações, Instituto Superior Técnico, Lisbon

REFERENCES

[1] Ghiglia, Dennis C., and Mark D. Pritt. Two-dimensional phase unwrapping: theory, algorithms, and software. New York:: Wiley, 1998.
[2] Graham, Leroy C. "Synthetic interferometer radar for topographic mapping." Proceedings of the IEEE 62, no. 6 (1974): 763-768.
[3] Kemao, Qian, Le Tran Hoai Nam, Lin Feng, and Seah Hock Soon. "Comparative analysis on some filters for wrapped phase maps." Applied optics 46, no. 30 (2007): 7412-7418.
[4] Lorenzo-Ginori, Juan V., Konstantinos N. Plataniotis, and Anastasios N. Venetsanopoulos. "Nonlinear filtering for phase image denoising." In Vision, Image and Signal Processing, IEE Proceedings-, vol. 149, no. 5, pp. 290-296. IET, 2002.
[5] Villa, Jesús, Juan Antonio Quiroga, and Ismael De la Rosa. "Regularized quadratic cost function for oriented fringe-pattern filtering." Optics letters 34, no. 11 (2009): 1741-1743.
[6] Li, Yan-Hua, Shi-Liang Qu, Xiang-Jun Chen, and Zhi-Yong Luo. "Phase pattern denoising using a regularized cost function with complex-valued Markov random fields based on a discrete model." Applied optics 49, no. 36 (2010): 6845-6849.
[7] Bioucas-Dias, Jose, Vladimir Katkovnik, Jaakko Astola, and Karen Egiazarian. "Absolute phase estimation: adaptive local denoising and global unwrapping." Applied optics 47, no. 29 (2008): 5358-5369.
[8] Bioucas-Dias, José M., and GonçaloValadão. "Phase unwrapping via graph cuts." Image Processing, IEEE Transactions on 16, no. 3 (2007): 698-709.
[9] Ferraioli, Giampaolo, AymenShabou, Florence Tupin, and Vito Pascazio. "Multichannel phase unwrapping with graph cuts." Geoscience and Remote Sensing Letters, IEEE 6, no. 3 (2009): 562-566.
[10] Ferraioli, Giampaolo, AymenShabou, Florence Tupin, and Vito Pascazio. "Fast InSAR multichannel phase unwrapping for DEM generation." In Urban Remote Sensing Event, 2009 Joint, pp. 1-6. IEEE, 2009.
[11] Shabou, Aymen, Fabio Baselice, and GiampaoloFerraioli. "Urban digital elevation model reconstruction using very high resolution multichannel InSAR data." Geoscience and Remote Sensing, IEEE Transactions on 50, no. 11 (2012): 4748-4758.
[12] Zhong, Heping, Jinsong Tang, Sen Zhang, and Xuebo Zhang. "A quality-guided and local minimum discontinuity based phase unwrapping algorithm for insar/insasinterferograms." (2014): 1-1.
[13] Kemao, Qian, Wenjing Gao, and Haixia Wang. "Windowed Fourier filtered and quality guided phase unwrapping algorithm: on locally high-order polynomial phase." Applied optics 49, no. 7 (2010): 1075-1079.
[14] Hongxing, Hao, and Wu Lingda. "PUMA-SPA: A Phase Unwrapping Method Based on PUMA and Second-Order Polynomial Approximation." Geoscience and Remote Sensing Letters, IEEE 11, no. 11 (2014): 1906-1910.
[15] Sharoze Ali, Habibulla Khan, IdrisShaik, Firoz Ali. "I-PUMA: Fast Phase Unwrapping Via IBFS Graph Cuts."International Journal of Engineering and Technology 7, no. 1 (2015): 254-265.
[16] Sharoze Ali, Habibulla Khan, IdrisShaik, Firoz Ali. "Very Fast Phase Unwrapping Via Grid-Cuts: G-PUMA."International Journal of Applied Engineering Research 10, no. 14 (2015): 34120-34131.
[17] Goldberg, Andrew V., Sagi Hed, Haim Kaplan, Robert E. Tarjan, and Renato F. Werneck. "Maximum flows by incremental breadth-first search." In Algorithms–ESA 2011, pp. 457-468. Springer Berlin Heidelberg, 2011.
[18] Jamriska, Ondrej, Daniel Sykora, and Alexander Hornung. "Cache-efficient graph cuts on structured grids." In Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, pp. 3673-3680. IEEE, 2012.

AUTHOR PROFILE

SHAROZE ALI born in India, 1985 .He pursued his B.Tech degree in E.C.E from Nimra college of Engineering & Technology (NCET), Vijayawada during 2002-2006 and M.Tech degree in Radar and Communications Engineering from the Koneru Lakshmaiah college of Engineering, Guntur, in 2008, where he is currently pursuing the Ph.D. degree in Electronics and Communications engineering, KL University,Guntur, under the supervision of Prof. Habibullah Khan. His research interests include phase unwrapping, and more broadly, remote sensing, Radar image processing, and graph cuts.

Dr. HABIBULLAH KHAN born in India, 1962. He obtained his B.E. from V R Siddhartha Engineering College, Vijayawada during 1980-84. M.E from C.I.T, Coimbatore during 1985-87 and PhD from Andhra University in the area of antennas in the year 2007. Dr. Habibullah khan presently working as Professor &Dean (SA), Department of the ECE at K.L.University. His research interested areas includes Antenna system designing, microwave engineering, Electromagnetic and RF system designing.

FIROZ ALI received the B.Tech degree in Electrical & Electronics Engineering from Nimra college of Engineering and Technology in 2003 and M.E degree in Power Electronics and Industrial Drives, from Satyabhama University, Chennai, in 2005 .He is currently working as Head of Department (H.O.D), Associate Professor in the Department of E.E.E, Nimra college of Engineering & Technology, Vijayawada. His research interests include Graph cuts, network optimization techniques, Fuzzy Logic, Remote sensing.

IDRISH SHAIK received the B.Tech degree in Electronics & Communication Engineering from Malineni Lakshmaiah Engineering College (MLEC) in 2009 and M.Tech degree in Embedded Systems from Vignan University, Guntur, in 2011 .He is currently an Assistant Professor with the Department of Electronics & Communication Engineering, Bapatla Engineering College, Bapatla. His research interests include image processing, Graph cuts, network optimization techniques, remote sensing.