Journal of the Korean Physical Society, Vol. 71, No. 2, July 2017, pp. 82∼87

Denoising Phase Unwrapping Algorithm for Precise Phase Shifting Interferometry

Phan Huy Phuc, Hyug-Gyo Rhee* and Young-Sik Ghim†

Center for Space Optics, Korea Research Institute of Standards and Science,
Daejeon 34113, Korea, and Department of Science of Measurement,
University of Science and Technology, Daejeon 34113, Korea

(Received 2 June 2017, in final form 20 June 2017)

Phase unwrapping refers to the process of recovering the absolute phase ϕ from a wrapped phase φ. Phase unwrapping arise in many applications, such as wavefront measurements in interferometry, field mapping in magnetic resonance imaging, the interferometry SAR process, measurements in adaptive optics and even a deflectometry. Gaining attention for a long time, many algorithms have been developed in relation to phase unwrapping problem. Jose's phase unwrapping algorithm via graph cuts (PUMA) is one of the most efficient algorithms given its ability to process various phase types with high accuracy levels. However, the drawback of PUMA is its computation speed when processing large complex phases, and its lack of a pre-filter, which raises issues when processing noisy data. In this paper, we propose a new algorithm which combines two structures: the incremental breadth-first search, which modifies the Boykov-Kolmogorov algorithm with regard to how it finds a path from the source to the sink of a graph in the max-flow problem in order to help reduce the processing time of the PUMA algorithm; and a pre-filter which operates on the principle of adaptive local denoising. Simulations and experimental implementations were used to demonstrate the ability of the proposed algorithm.

PACS numbers: 42.87.-d, 42.86.+b, 42.79.Bh
Keywords: Phase unwrapping, Phase shifting interferometry
DOI: 10.3938/jkps.71.82

I. INTRODUCTION

Phase unwrapping is the process of estimating the absolute (true) phase ϕ from the wrapped (measured, principle) phase φ. The need for phase unwrapping algorithms arises in many imaging techniques, such as phase measuring interferometry (PSI) [1, 2], field mapping in magnetic resonance imaging (MRI) [3–5], the interferometry SAR process (InSAR) [6–8], measurements in adaptive optics [9] and even a deflectometry [10–12].

Essentially, the goal of a phase unwrapping algorithm is to find the integer numb er k that satisfies the equation $\phi = 2\pi k + \varphi$, where ϕ is the estimated unwrapped phase and φ is the given wrapped phase, whose value exists in the interval of $[-\pi, \pi]$. The phase unwrapping approach mainly falls into one of four classes: path following [13–15], minimum L^p norm [16, 17], Bayesian/regularization [7, 18], and parametric modeling [19, 20]. Path following algorithms apply line integration schemes over the wrapped phase image and basically rely on the assumption that the Itoh condition holds along the integration path. Wherever this condition fails, different integration paths may lead to different unwrapped phase values. Techniques employed to handle these inconsistencies include what are known as branch cuts and quality maps. Minimum norm methods attempt to find a phase solution for which the norm of the difference between absolute phase differences and wrapped phase differences (*i.e.*, a second-order difference) is minimized. This is, therefore, global minimization in the sense that all of the observed phases are used to compute a solution. The Bayesian approach relies on a data-observation mechanism model as well as a priori knowledge of the phase to be modeled. For instance, in on study [21], a nonlinear optimal filtering is applied, while in another [22], an InSAR observation model is considered, taking into account not only the image phase but also the backscattering coefficient and correlation factor images, which are jointly recovered from InSAR image pairs. Other studies propose a fractal-based prior [23] and dynamic programming techniques [24]. Finally, parametric algorithms constrain the unwrapped phase to a parametric surface. Low-order polynomial surfaces have also been used [25]. Very often in actual applications, a single polynomial is not enough to describe accurately a complete surface. In such cases, the image is partitioned and different parametric mod-

*E-mail: hrhee@kriss.re.kr
†E-mail: young.ghim@kriss.re.kr; Fax: +82-42-868-5814

els are applied to each partition [25]. Among the algorithms that have been developed thus far, PUMA [26] is considered to be a minimum method, proving to be one of the most efficient phase unwrapping algorithms. As the first technique to adapt graph cutting for energy minimization in the phase unwrapping problem, PUMA has the ability to unwrap various types of phases at a short amount of time with high accuracy levels, even when encountering discontinuities. Despite its superiority, PUMA is often not feasible when used to unwrap large-sized phases and its sensitivity of noisy phases.

In this paper, we propose an algorithm that modifies the Boykov-Kolmogorov (BK) algorithm using the incremental breadth-first search (IBFS) method [27] to find paths from the source to the sink of a graph. Among the following sections, section II will present the PUMA algorithm and its operational principles. Section III introduces the proposed algorithm with the ideas of IBFS and a pre-filter. In sections IV and V, there are simulations and experimental implementations which demonstrate the performance capabilities of the proposed algorithm.

II. PREVIOUS PUMA ALGORITHM

The main idea of the PUMA algorithm is the use of an energy minimization framework for phase unwrapping, where minimization is carried out by a sequence of max-flow/min-cut calculations. Generally, PUMA consists of two algorithms that are classified according to the clique potential. Here, the clique refers to a set of sites that are mutually neighbors. If the clique potential is greater than one, such cliques are termed a convex potential (p $\geq$ 1) and PUMA undergoes precise energy minimization. For non-convex clique potentials (p $\leq$ 1), PUMA offers an approximation solution owing to its ability to preserve discontinuity. Both algorithms solve energy minimization problems by computing a sequence of binary optimizations, each one solved by graph cut techniques.

Firstly, let define the energy for site of (i,j) as shown in Fig. 1,

$$E(k|\psi) = \sum_{i,j \in G_0} V(\triangle\phi^h_{ij})v_{ij} + V(\triangle\phi^v_{ij})h_{ij}, \qquad (1)$$

where $k \equiv \{k_{ij} \in Z : (i,j) \in G_0\}$ (Z is the integer set, G_0 is image pixel indexing 2-D grid) is an image of the integer denoting 2π multiples, also known as the wrap-count image; $\psi \equiv \{\psi_{ij} \in [-\pi,\pi) : (i,j) \in G_0\}$ is the observed wrapped phase image, $h_{ij}, v_{ij} \in \{0,1\}$ and $h_{ij}, v_{ij} = 0$ signal a discontinuity in the horizontal and vertical direction, respectively; V is the clique potential function; $(.)^h$ and $(.)^v$ denote pixel horizontal and vertical differences, as given by:

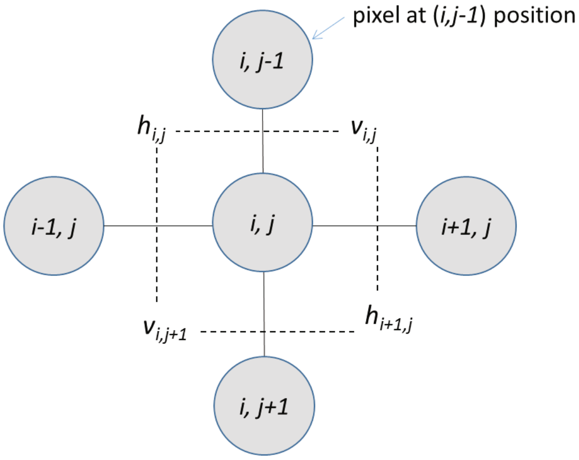

Fig. 1. Representation of the site (i,j) and its first-order neighbors along with the variables h$_{ij}$ and v$_{ij}$ signaling horizontal and vertical discontinuities, respectively [26].

$$\begin{aligned}\triangle\phi^h_{ij} &\equiv [2\pi(k_{ij} - k_{ij-1}) - \triangle\psi^h_{ij}], &(2)\\ \triangle\phi^v_{ij} &\equiv [2\pi(k_{ij} - k_{i-1j}) - \triangle\psi^v_{ij}], &(3)\\ \triangle\psi^h_{ij} &\equiv \psi_{ij-1} - \psi_{ij}, &(4)\\ \triangle\psi^v_{ij} &\equiv \psi_{i-1j} - \psi_{ij}. &(5)\end{aligned}$$

The main goal of the PUMA algorithm is to find the integer image k that minimizes the energy equation (1), k being such that $\phi = 2\pi k + \psi$, where ϕ is the estimated unwrapped phase image.

By using the proof of Equivalence between the local and global minimization, a convergence analysis and by mapping binary optimizations on graph max-flows, the energy equation can be written as follows:

$$\begin{aligned}E(k^t + \delta|\psi) \;=\; \sum_{ij\in G_0} & V[2\pi(\delta_{ij} - \delta_{ij-1}) + a^h]v_{ij}\\ & + V[2\pi(\delta_{ij} - \delta_{i-1j}) + a^v]h_{ij}. \qquad (6)\end{aligned}$$

The authors of PUMA then exploit a one-to-one map existing between the energy function (6) and cuts on a directed graph $G = (V, E)$ (V and E denote the set of vertices and edges, respectively) with nonnegative weights. The graph has two special vertices, *i.e.*, the source s and the sink t. An $s - t$ cut C = S, T is a partition of vertices V into two disjoint sets S and T such that $s \in S$ and $t \in T$. The number of vertices is 2+MxN (two terminals, the source and the sink, plus the number of pixels). The cost of the cut is the sum of the costs of all edges between S and T.

Then following four energy terms then introduced:

$$\begin{aligned}E^{ij}(0,0) &= V(a)d_{ij},\\ E^{ij}(1,1) &= V(a)d_{ij},\\ E^{ij}(0,1) &= V(-2\pi + a)d_{ij},\\ E^{ij}(1,0) &= V(2\pi + a)d_{ij}. &(7)\end{aligned}$$

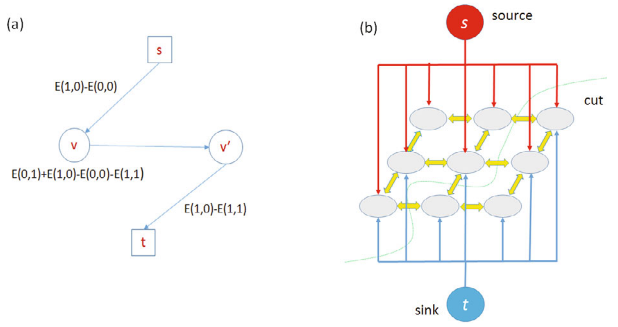

Fig. 2. (Color online) (a) Elementary graph for a single energy term, and (b) The graph obtained by adding elementary graphs [27]

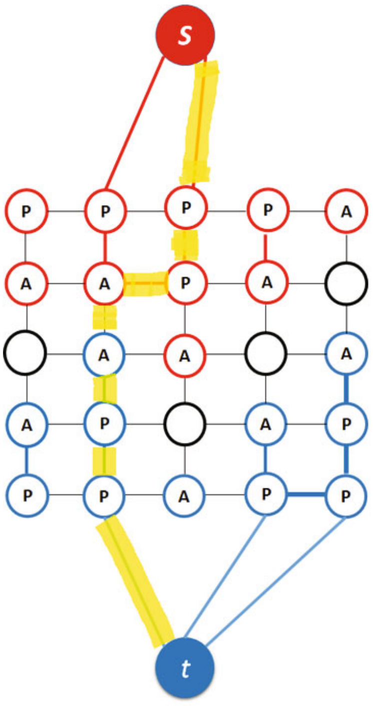

Fig. 3. (Color online) Example of search trees S (red nodes) and T (blue nodes) at the end of the growth stage when a path (yellow line) from the source s to the sink t is found. Active and passive nodes are labeled by the letters A and P, correspondingly. Free nodes appear in black.

Consequently, for each term E_h^{ij} and E_v^{ij}, the authors construct an elementary graph with four vertices $\{s, t, v, v'\}$ where $\{s, t\}$ represents the source and the sink, common to all terms, and $\{v, v'\}$ represents the two pixels involved [v being the left (up) pixel and v' being the right (down) pixel]. A directed edge $\{v, v'\}$ can then be defined with the weight $E(0,1) + E(1,0) - E(0,0) - E(1,1)$. Moreover, if $E(1,0) - E(0,0) > 0$, an edge $\{s, v'\}$ is defined with the weight $E(1,0) - E(0,0)$; otherwise, we have an edge $\{v, t\}$ with the weight E(0,0)-E(1,0). In a similar manner, we can define edges for v'. Figure 2(a) shows an example where $E(1,0) - E(0,0) > 0$ and $E(1,0) - E(1,1) > 0$. Figure 2(b) illustrates the complete graph obtained at the end. After the energy is mapped onto the graph, the energy is easily minimized by using max-flow/min-cut on the constructed graph. To solve the max-flow/min-cut problem, the authors of PUMA chose the BK algorithm, as illustrated in Fig. 3. PUMA runs for k iterations to unwrap the phase of a profile, where k is the number of 2π multiples.

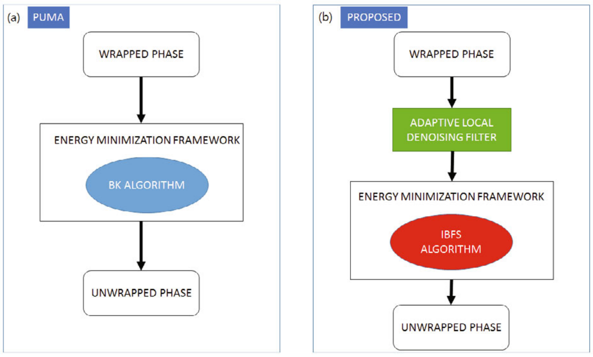

Fig. 4. (Color online) Flow charts of (a) the PUMA, and (b) the proposed algorithms.

Although, the PUMA algorithm of the convex potential does not fit the non-convex potential, the authors of PUMA proved that this issue can be resolved by applying majorize-minimize concepts to the energy function and extending the ranges of the allowed moves and sequences of s-jumps instead of only 1-jump.

III. PROPOSED ALGORITHM

In the proposed algorithm, we have two objectives: reducing the processing time of the PUMA algorithm and providing a pre-filter for the algorithm. There are two main factors which make the PUMA be a complex and time-consuming algorithm - the algorithm used to minimize energy by max-flow/min-cut method (BK algorithm) and the property of PUMA itself which need several iteration until find out the image integer k for the equation: $\phi = 2\pi k + \psi$. In order to reduce the processing time of the PUMA algorithm, we use IBFS - an extension of the BK algorithm used in the PUMA algorithm, to maintain the breath-first search tree, which leads to a polynomial time bound ($O(n^2m)$) as shown in Fig. 4. The pre-filter selected here is an adaptive local denoising scheme. Adaptive local modulo-2π phase denoising is a new algorithm based on local polynomial approximations. The zero-order and the first-order approximations of the phase are calculated in sliding windows of varying size. Zero-order approximation is used for pointwise adaptive window size selection, whereas the first-order approximation is used to filter the phase in the obtained windows.

1. Incremental Breath First Search(IBFS)

As noted in the previous section, in PUMA, the authors use BK as the max-flow/min-cut solver, and the BK algorithm attains this by using the concept of augmenting paths. Essentially, the BK algorithm has three steps: the growth, the augmentation and the adoption step. During the growth step, two trees are expanded: the S-tree from the source and the T-tree from the sink. The active nodes explore adjacent non-saturated edges and acquire new children from a set of free nodes. The newly acquired nodes become active members of the corresponding search trees. As soon as all neighbors of a given active node are explored the active node becomes passive. The growth stage terminates if an active node encounters a neighboring node that belongs to the opposite tree. In this case we detect a path from the source to the sink. The BK algorithm for finding max-flow/min-cut of a graph is an algorithm based on Ford-Fulkerson style "augmenting paths". Generally, the augmenting path from source to sink should be the shortest path, it make the complexity of the Ford-Fulkerson style "augmenting paths" algorithm is O(mn^2). The main drawback of BK is that, the augmenting path is not always the shortest path, we cannot make any assumptions about the structure of the trees and thus no assumption on the length of the augmenting paths as well, results in worse than the complexity O(mn^2). IBFS modifies BK algorithm to maintain breadth-first search trees, which leads to a polynomial time bound O(mn^2); therefore reduce the calculation time comparing to PUMA algorithm which use BK method. As the BK algorithm, IBFS also includes three steps: the growth, the augmentation and the adoption step. The algorithm introduces distance labels $d_s(v)$ and $d_t(v)$ for every vertex v. The two trees, S and T, satisfy the tree invariants, for some values D_s and D_t, the trees contain all vertices at distances up to D_s from s and up to D_t to t, respectively. We also maintain the invariant that $L = D_s + D_t + 1$ is a lower bound on the length of any augmenting path, such as the disjoint trees. If a vertex v is in S, $d_s(v)$ is the meaningful label value and $d_t(v)$ is unused. The situation is symmetric for vertices in T. Labels of N-vertices (not in any tree) are irrelevant. Because at most one $d_s(v)$ or $d_t(v)$ value is used at any given time, one can use a single variable to represent both labels. The first step is the growth step, where the S-tree is grown by scanning all nodes whose distance D_s is equal to the current max distance. Any free node is found and is added to the tree and is assigned the distance $ds(u) = D_s + 1$. In such a search of a free node, if a T-node is found, the growth step is interrupted by the augmentation step. After the growth step, if there are no nodes with distance $D_s + 1$, the algorithm terminates; otherwise, D_s incremented and the growth step begins anew. Growing the T-tree is done symmetrically. Augmenting the path creates S and T orphans. The S-orphan process is starts with a search of a potential parent (u) which satisfies either $d_s(v) = d_s(u) + 1$. Alternatively, it should minimize $d_s(u)$. If none such elements exists, then V is freed and all its children are made S-orphans. Otherwise, we set $p(v) = u$ and $d_s(v) = d_s(u) + 1$. Processing the T-orphans is done in the same manner and V is freed only if it satisfies $(u) \geq D_t$ as it is not advisable to grow T at the growth stage of S. During the adoption stage, in order to maintain the validity of distance labelling, a parent is found at same distance labeling level .Otherwise, IBFS attempts to reattach the orphan as close to the root of the tree as possible. During this process, most likely the orphans will be re-adopted by their previous parent, continuing until their new distance exceeds the maximum distance of the tree.

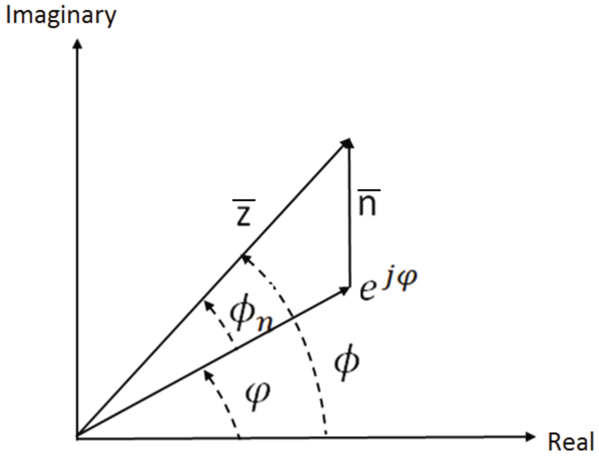

Fig. 5. Illustration of the observed phase model: φ is the true phase, ϕ is the observed phase, and ϕ_n is the phase component of ϕ due to noise vector n.

2. Adaptive Local Modulo -2π Phase Denoisng

Details of observation models linking the noisy wrapped phase to the true phase depend on the coherent imaging system under consideration. Nevertheless, the essential aspect of all of these observation mechanisms is captured by the following relationship,

$$z = A\exp(j\varphi) + n \qquad A > 0, \tag{8}$$

where $n = n_I + jn_Q$ denotes the complex-valued zero-mean circular white noise of variance $2\sigma^2$ (*i.e.*, n_I and n_Q are zero-mean independent Gaussian random variables with variance σ^2). Given that the noise is additive, we define the signal-to-noise-ratio as $SNR \equiv 1/(2\sigma^2)$. Figure 5 illustrates the different components of the observed model in the complex plane: φ is the true phase, ϕ is the observed phase, and ϕ_n is the phase component of ϕ due to noise vector n. The phase ϕ is given by,

$$\phi = angle(z) = W(\varphi + \phi_n), \qquad \phi \in [-\pi, \pi), \tag{9}$$

where W represents the wrapping operator mapping the noisy phase $\varphi + \phi_n$ into the basic phase interval $[-\pi, \pi]$. For $\phi_n = 0$, there is a relation between the wrapped ϕ and the non-wrapped absolute phase φ, $\varphi = \phi + 2k\pi$, $\phi \in [-\pi, \pi)$, where k is an integer. Let us define the parameterized family of first-order polynomials as,

$$\tilde{\varphi}(u, v|c) = p^T(u, v)c, \qquad (10)$$

where $p = [p_1, p_2, p_3]^T = [1, u, v]^T$ and $c = [c_1, c_2, c_3]^T$ is a vector of parameters.

Given the optimal solution ,

$$(\hat{c}_2, \hat{c}_3) \in \arg\max |F_h(c_2, c_3)|, \qquad (11)$$
$$\hat{c}_1 = angle[F_h(\hat{c}_2, \hat{c}_3)] \qquad (12)$$

where $F_h(c_2, c_3)$ is the windowed discrete Fourier transform of the normalized data $z_\phi = z/|z|$ at point (x, y) with frequency (c_2, c_3); *i.e.*,

$$F_h(c_2, c_3) = \sum w_{h,z} Z_\phi(x + x_s, y + y_s) \times e^{-j(c_2 x_s + c_3 y_s)}. \qquad (13)$$

By setting $c_2 = 0$ and $c_3 = 0$ in Eq. (10), a minor modification, the calculus carried out for the first-order polynomial approximation leads to a zero-order polynomial based estimate:

$$\hat{\varphi}_h(x, y) = \hat{c}_1(x, y) = angle[F_h(0, 0)]. \qquad (14)$$

Therefore, the zero-order approximation produces biased estimates, whereas the first-order outcome does not. The zero-order approximation based on symmetric windows tends to yield unbiased wrapped phase estimates, at least for low window sizes and low frequencies (phase slopes) c_2 and c_3.

In the next step of the denoising stage, we apply the intersection of confidence intervals (ICI) for window size adaptation. The ICI rule defines the adaptive window size, denoted by h^+, as the largest window size parameter $h \in$ H for which the estimate $\hat{\varphi}_h$ does not differ significantly from the estimates corresponding to the smaller window sizes. 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$_{hj}$, $1 \leq hj \leq h_i$, is considered with increasing h_i. Let h^+ be the largest of those h_i values for which the intervals Q$_{hj}$, $1 \leq h_j \leq h_i$, have a point in common. This h^+ defines the adaptive window size and the adaptive estimate as $\hat{\varphi}_h$.

IV. SIMULATION AND EXPERIMENTAL RESULT

In order to demonstrate the superiority of the proposed algorithm relative to the PUMA algorithm, initially we create simulation data and input them into these two algorithms. Four wrapped phases with increasing sizes: 100 × 100, 250 × 250, 500 × 500, and 1000 × 1000 are inputted into the PUMA algorithm and the proposed algorithm. The results show regarding the processing time that the proposed algorithm outperforms PUMA at all data sizes, with improvements ranging from 14 to 22%, as shown in Fig. 6. In the next step, the denoising ability of the proposed algorithm was tested under different levels of noise, as follows: without noise, 2%, 4%, 8%, and 10%. Again, the proposed algorithm can reduce the unwrapped phase error 10 - 40% in root-mean-square value compared with the previous PUMA algorithm. It should be note that this rms compound of error of PUMA algorithm and small amount of error coming from the process of simulation by wrapping the original phase, rounding the number at several steps. Without noise, PUMA algorithm will give no error, the small rms error showed in Fig. 7 purely comes from the process of simulation.

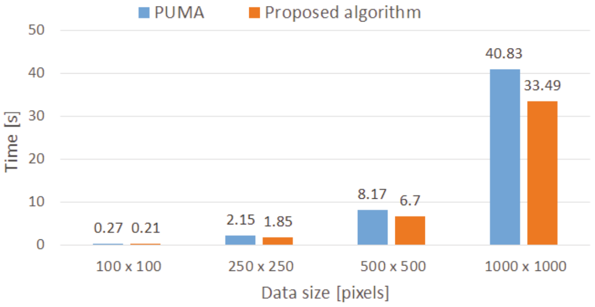

Fig. 6. (Color online) Processing time comparison between PUMA and the proposed algorithm.

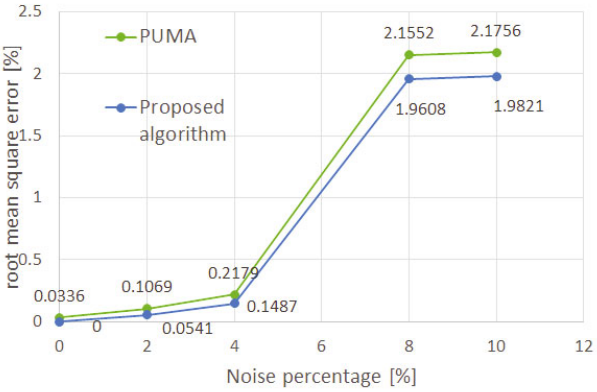

Fig. 7. (Color online) Unwrapped phase errors after adopting two algorithms under affection of noise.

In the experimental implementation, we used PUMA and proposed algorithm to process data of a wrapped phase from a commercial phase shifting interferometry. Figure 8 is the images of fringes captured directly from the screen. The PUMA algorithm processes the data in 255 seconds while the proposed algorithm runs in 216 seconds proving the proposed algorithm outperform PUMA in terms of processing time. Subsequently, we consider the error level of two algorithms. Using the proposed al-

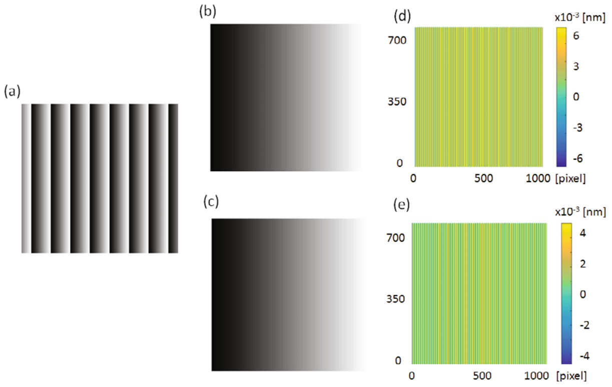

Fig. 8. (Color online) (a) Measure wrapped phase obtained from a flat mirror, (b) unwrapped phase with the previous PUMA algorithm, (c) unwrapped phase with the proposed algorithm, (d) error of the PUMA and the (e) proposed algorithm. The error was defined as the difference from the standard Fizeau interferometer.

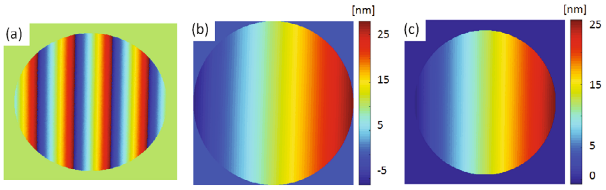

Fig. 9. (Color online) (a) Measure wrapped phase obtained from an aspheric mirror, (b) unwrapped phase with the previous PUMA algorithm, and (c) unwrapped phase with the proposed algorithm.

gorithm to process the experimental data shows that this algorithm does not result in a good phase at the edges and four corners. However, at other areas inside the phase map, the proposed algorithm indeed improves the unwrapped phase accuracy compared to PUMA about 20 - 30%. Next, we processed the data wrapped phase data of an aspheric mirror by the proposed algorithm, and then compare the unwrapped phase to the result of PUMA algorithm, as shown in Fig. 9. The peak-to-valley value of the proposed algorithm was smaller than the previous PUMA's result.

V. CONCLUSION

In this paper, we propose a phase unwrapping algorithm that improves upon PUMA, an effective existing algorithm. Simulation and experimental results demonstrate the superiority of the proposed algorithm compared to PUMA in terms of both speed and ability when processing noisy data. However, the performance of the proposed algorithm was less satisfactory with noisy data at the edges and four corners. This is slated for future work in an effort to improve the algorithm when used for these tasks.

REFERENCES

[1] J. E. Greivenkamp and J. H. Bruing, *Optical Shop Testing 2nd edition*, edited by D. Malacara (Wiley, New York, 1992), Chap. 14.

[2] D. Malacara, *Optical Shop Testing 2nd edition*, edited by D. Malacara (Wiley, New York, 1992), Chap. 2.

[3] P. Lauterbur, Nature **242**, 190 (1973).

[4] M. Hedley and D. Rosenfeld, Magn. Res. Med. **24**, 177 (1992).

[5] K. N. Kim, P. Heo, Y. B. Kim, G. C. Han, J. Kor. Phys. Soc. **66**, 175 (2015).

[6] L. Graham, Proc. IEEE **62**, 763 (1974).

[7] H. Zebker and R. Goldstein, J. Geophys. Res. B **91**, 4993 (1996).

[8] J. Dias and J. Leitao, IEEE Trans. Image Process. **11**, 336 (1975).

[9] D. L. Fried, J. Opt. Soc. Amer. A **67**, 370 (1997).

[10] P. H. Phuc, N. T. Manh, H. G. Rhee, Y. S. Ghim, H. S. Yang and Y. W. Lee, J. Korean Phys. Soc. **70**, 469 (2017).

[11] W. Li, P. Huke, J. Burke, C. Kopylow and R. B. Bergmann, Proc. SPIE. **9203**, F-1 (2015).

[12] S. Na, Y. Yu and S. Shin, J. Kor. Phys. Soc., **69**, 286 (2016).

[13] R. Goldstein, H. Zebker and C. Werner, Proc. Symp. Ionospheric Effects on Communication and Related System **23**, 713 (1988).

[14] S. Madsen, H. Zebker and J. Martin, IEEE Trans. Geosci. Remote Sens. **31**, 246 (1993).

[15] W. Xu and I. Cumming, Proc. Int. Geoscience and Remote Sensing Symp. **4**, 2044 (1996).

[16] T. Flynn, J. Opt. Soc. Amer. A **14**, 2692 (1997).

[17] M. Costantini, IEEE Trans. Geosci. Remote Sens. **36**, 813 (1998).

[18] J. Marroquin and M. Rivera, J. Opt. Soc. Amer. **12**, 2393 (1995).

[19] B. Friedlander and J. Francos, IEEE Trans. Signal Process. **44**, 2999 (1996).

[20] Z. Liang, IEEE Trans. Med. Imag. **15**, 893 (1996).

[21] J. Leitão and M. Figueiredo, IEEE Trans. Image Process. **7**, 868 (1997).

[22] J. Dias and J. Leitão, *Proc. Eur. Conf. Synthetic Aperture Radar* (Friedrichshafen, May, 1998), p. 373.

[23] M. Datcu and G. Palubinskas, Proc. SPIE. **3497**, 155 (1998).

[24] L. Ying, Z. Liang, D. Munson Jr., R. Koetter and B. Frey, IEEE Trans. Med. Imag. **25**, 128 (2006).

[25] J. M. Bioucas-Dias and G. Valadao, IEEE Trans. Image Process. **16**, 698 (2007).

[26] Y. Boykov and V. Kolmogorov, IEEE Trans on PAMI **26**, 1124 (2004).

[27] R. Yan, L. Shao, L. Liu and Y Liu, Signal Processing **103**, 36 (2014).