Quality-guided phase unwrapping technique: comparison of quality maps and guiding strategies

Ming Zhao,[1] Lei Huang,[2] Qican Zhang,[3] Xianyu Su,[3] Anand Asundi,[2] and Qian Kemao[1,*]

[1]School of Computer Engineering, Nanyang Technological University, 639798, Singapore

[2]School of Mechanical and Aerospace Engineering, Nanyang Technological University, 639798, Singapore

[3]Opto-Electronics Department, Sichuan University, Chengdu 610064, China

*Corresponding author: mkmqian@ntu.edu.sg

Received 31 May 2011; revised 19 August 2011; accepted 20 August 2011;
posted 8 September 2011 (Doc. ID 148376); published 16 November 2011

Quality-guided phase unwrapping is a widely used technique with different quality definitions and guiding strategies reported. It is thus necessary to do a detailed comparison of these approaches to choose the optimal quality map and guiding strategy. For quality maps, in the presence of noise, transform-based methods are found to be the best choice. However in the presence of discontinuities, phase unwrapping is itself unresolved and hence quality-guided phase unwrapping is not sufficient. For guiding strategies, classical, two-section, and stack-chain guiding strategies are chosen for comparison. If accuracy is the foremost criterion then the classical guiding strategy with a data structure of indexed interwoven linked list is best. If speed is of essence then the stack-chain guiding strategy is the one to use. © 2011 Optical Society of America

OCIS codes: 100.2650, 100.5088.

1. Introduction

In many phase-measuring techniques, such as interferometric synthetic aperture radar (InSAR) [1], fringe projection profilometry (FPP) [2,3], and optical interferometric methods [4,5], the retrieved phase is wrapped within a range from $-\pi$ to π, i.e., modulo 2π. The wrapped phase needs to be unwrapped in order to construct a continuous phase distribution. This process is called phase unwrapping.

Many phase unwrapping algorithms have been proposed, including temporal phase unwrapping methods such as dynamic unwrapping method [6], multifrequency method [7–9], and heterodyne method [10] and spatial phase unwrapping methods such as Goldstein's method [11], quality-guided method [12], Flynn's method [13], and minimum L^p-norm method [14]. Temporal phase unwrapping methods are effective and robust, but require multiframes of wrapped phase along the time axis, or multifrequency fringe patterns. On the other hand, spatial phase unwrapping methods have fewer restrictions, but disjoint regions and phase discontinuities are difficult to handle.

The quality-guided phase unwrapping (QGPU) technique has been well studied in the past two decades [15–26] as an efficient, fast, and automatic spatial phase unwrapping method. Figure 1 shows the profile of a real box measured by fringe projection profilometry to exemplify the concept of QGPU. The wrapped phase in Fig. 1(a) is unwrapped into Fig. 1(d) by following a certain unwrapping path in Fig. 1(c) according to the quality map in Fig. 1(b). The complicated quality map is due to the texture of the box which is intentionally selected to demonstrate the capability of QGPU. For the unwrapping path, the pixels in light yellow are unwrapped first and those in dark red are unwrapped last. In QGPU, definition of quality and strategies for guiding the

0003-6935/11/336214-11$15.00/0

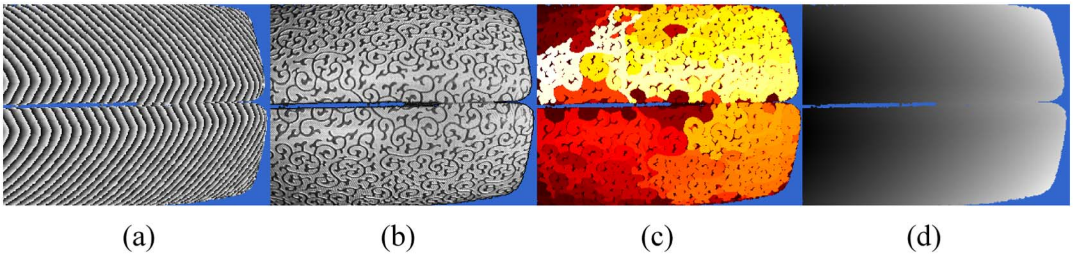

Fig. 1. (Color online) Concept of of QGPU. (a) Wrapped phase map, (b) quality map, (c) unwrapping path map (pixels in light yellow are unwrapped first and those in dark red are unwrapped last), and (d) unwrapped phase map.

unwrapping are two major aspects affecting the accuracy and speed of the result. Many quality maps have been proposed, including the correlation coefficient map in InSAR [27], modulation map or reliability map from optical metrology [16,18], pseudo correlation coefficient map [28], phase derivative variance map [29], maximum phase gradient map [28], first phase difference map [16], second phase difference map [19,30,31], and amplitude map or ridge map from transform methods [16,21,22]. For the guiding strategy, the most widely used approach is the classical quality guiding strategy [16], which always unwraps the pixel with the highest quality first. Other guiding strategies include two-section guiding strategy and stack-chain guiding strategy [15].

How to select a proper quality map and guiding strategy is thus important and relevant in fringe processing. This paper provides a detailed comparison of the different quality maps and guiding strategies. The principle and problems of spatial phase unwrapping and the process of QGPU are introduced in Section 2. The different quality maps are compared in Section 3. The guiding strategies for flood fill and related programming using MATLAB and C++ are discussed in Section 4. The discussion and conclusions on the choice of quality maps and guiding strategies are highlighted in Section 5.

2. Principle and Problems of Spatial Phase Unwrapping and QGPU

A. Basic Principle of Spatial Phase Unwrapping

In spatial phase unwrapping, a pixel 'a' that neighbors an unwrapped pixel 'b', can be unwrapped as follows:

$$\varphi_a^u = \varphi_a^w + 2\pi \times \mathrm{round}\left(\frac{\varphi_b^u - \varphi_a^w}{2\pi}\right), \tag{1}$$

where the function of round$(\cdot)$ converts the input to its nearest integer; φ_b^u is the unwrapped phase of pixel b; φ_a^w, and φ_a^u represent wrapped and unwrapped phase of pixel a, respectively. The basic requirement of Eq. (1) is that the difference of the unwrapped phase between neighboring pixels is within $(-\pi, \pi]$. In Fig. 2, the unwrapped dark green pixels are used along with Eq. (1) to unwrap the light pink pixels. If noise reduction techniques are adopted, the unwrapping results of original noisy wrapped phase maps can also be obtained from the denoised unwrapped phase maps through a phase congruence operation [28].

B. Unwrapping Failure

Although the phase unwrapping process described in Eq. (1) seems straightforward, problems arise if:

i. Undersampling causes the absolute phase difference between two neighboring pixels to exceed the range of $(-\pi, \pi]$ and thus violates the requirement of Eq. (1). This situation is not considered in this paper.

ii. Noise can give wrong unwrapping values in Eq. (1) and result in the failure of phase unwrapping. Suppressing noise before phase unwrapping is thus vital.

iii. Invalid areas do not reflect the actual physical quantity. Decorrelation in InSAR images and the shadow problems in FPP are examples where invalid areas occur. The unwrapping results within the invalid areas are not of interest, but they may propagate errors into other pixels, which is a major concern.

iv. Discontinuities in true phase cannot be recognized by the phase unwrapping algorithm and are treated as continuous by Eq. (1), resulting in incorrect unwrapping.

v. If a pixel is incorrectly unwrapped due to reasons highlighted above, the subsequent unwrapping based on this pixel will also be wrong.

C. Basic Process of QGPU

All light pink pixels in Fig. 2 can be simultaneously unwrapped based on their unwrapped (dark green) neighbors using Eq. (1). Rather than unwrapping all light pink pixels with same priority, they are placed in a queue called the adjoin list and only the pixel with the highest quality is chosen for unwrapping. Noisy pixels and invalid areas that are

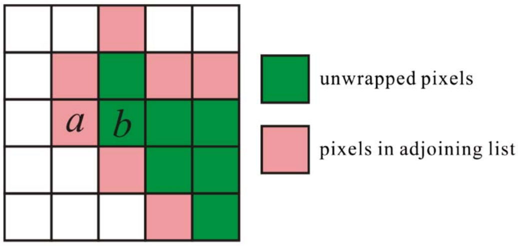

Fig. 2. (Color online) Diagram of spatial unwrapping process.

assigned low-quality values are moved to the back of the queue in a quality-guided phase unwrapping process. Problems of discontinuities can also be avoided by QGPU in some cases. The process can be described as follows.

Step 1: Find the seed pixel with the highest quality from the quality map and push it into the adjoin list.

Step 2: Find the pixel with the highest quality from the adjoin list, unwrap it, and remove it from the adjoin list. Push the unprocessed neighbors into the adjoin list.

Step 3: Repeat step 2 until the adjoin list is empty.

3. Comparison of Quality Maps

A. Resources for Quality Calculation

There are two possible approaches for obtaining the quality map from measurement data.

Case A: a quality map is calculated from the raw measurement data. For example in phase-shifting methods a complex-valued fringe pattern f_c can be obtained as [32]

$$f_c = M \cdot \exp(j \cdot \varphi^w), \qquad (2)$$

where j is the imaginary unit, φ^w is the wrapped phase, and M is the intensity modulation. From f_c both φ^w and M can be delineated.

Case B: The quality map can be calculated from the wrapped phase or the normalized complex-valued fringe pattern f_n as:

$$f_n = 1 \cdot \exp(j \cdot \varphi^w) = \frac{f_c}{|f_c|}. \qquad (3)$$

B. Classification of Quality Maps

Quality maps are classified according to resources used to generate them:

i. Quality maps only from case A, including the correlation coefficient map in InSAR [28]; the modulation map Q_{MOD} [16], and the reliability map Q_{REL} [16,18] from least squares fitting in phase-shifting method.

ii. Quality maps only from case B, including the pseudocorrelation coefficient map Q_{PCC} [28], the phase derivative variance map Q_{PDV} [28], the first phase difference map Q_{FPD} [16], the second phase difference map Q_{SPD} [19], and their combinations [20,33].

iii. Quality maps from both case A and case B, which are based on transform methods, such as the Fourier transform (FT) method [2,34,35], the windowed Fourier filtering (WFF) method [22,32,36,37], the windowed Fourier ridges (WFR) method [22,32,36,37], and the wavelet transform (WT) method [21,38]. The quality maps include the amplitude map from Fourier transform method Q_{FT} [16], amplitude map from windowed Fourier filtering method Q_{WFF} [22], ridge map from windowed Fourier ridges method Q_{WFR} [22], and ridge map from wavelet transform method Q_{WT} [22].

In summary, quality maps from case A include Q_{MOD}, Q_{REL}, Q_{FT}, Q_{WFF}, Q_{WFR}, and Q_{WT}, while quality maps from case B include Q_{PCC}, Q_{PDV}, Q_{PFG}, Q_{SPD}, Q_{FT}, Q_{WFF}, Q_{WFR}, and Q_{WT}. These two cases will be compared separately, where noise and discontinuities are two major concerns.

C. Comparison of Different Quality Maps for Noisy Phase Maps

1. *Generation of Noisy Phase Maps*

A four-step phase-shifted fringe pattern (256 × 256 pixels) with both speckle noise and normally distributed random noise are simulated as shown in Fig. 3(a). The true wrapped phase map shown in Fig. 3(b) is simulated as

$$\varphi(x,y) = \frac{2\pi}{8}(x-128) + 3 \times \mathrm{peaks}(x,y), \qquad (4)$$

where φ denote the true phase map and the function, *peaks* is a MATLAB built-in function. The true modulation map M in Fig. 3(c) is simulated as

$$M(x,y) = \frac{255}{2} \times \mathrm{normalize}(\mathrm{peaks}(x,y)), \qquad (5)$$

where the function normalize(·) linearly scales the input so that it is between 0 and 1. The wrapped phase is then retrieved using the least squares phase-shifting algorithm [3] and is shown in

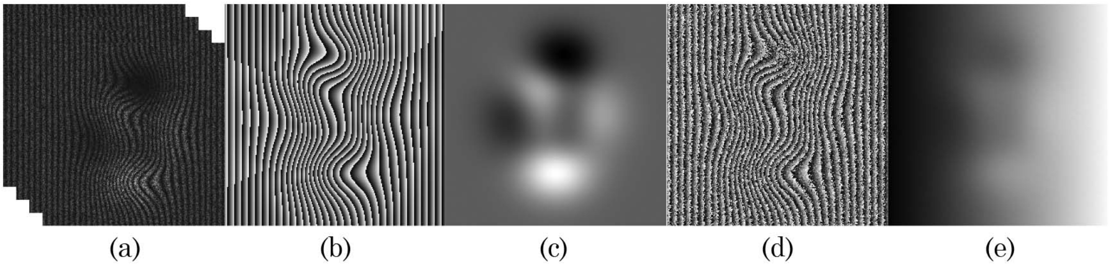

Fig. 3. Simulated of noisy phase maps. (a) Four-step phase-shifted fringe patterns with speckle and random noise, (b) true wrapped phase, (c) true modulation, (d) retrieved noisy wrapped phase, and (e) retrieved unwrapped phase (with phase noise).

Fig. 3(d). The wrapped phase, unwrapped using a congruence operation based on the simulated true phase is shown in Fig. 3(e). It has to be noted that the signal-to-noise ratio is almost zero in regions of low modulation, and the wrapped phase and unwrapped phase results in these regions are not reliable. However, this is intentionally simulated to investigate the performance of the quality maps.

2. *Comparison of Quality Maps in Case A*

Six quality maps, Q_{MOD}, Q_{REL}, Q_{FT}, Q_{WFF}, Q_{WFR}, and Q_{WT}, are calculated from raw fringe patterns and then their performances are compared. The results using these quality maps are shown in Fig. 4. The results show that in the presence of noise, transform-based quality maps, Q_{FT}, Q_{WFF}, Q_{WFR}, and Q_{WT}, perform better than Q_{MOD} and Q_{REL}.

3. *Comparison of Quality Maps in Case B*

Eight quality maps, Q_{PCC}, Q_{PDV}, Q_{FPD}, Q_{SPD}, Q_{FT}, Q_{WFF}, Q_{WFR}, and Q_{WT}, are calculated from the wrapped phase map and then their performances are compared. The results are given in Fig. 5, which show that, once again, transform-based quality maps perform better than others.

4. *Further Comparison of Transform-Based Quality Maps*

To further compare the performances within transform-based methods, a more complicated example is simulated, where the fringe patterns, true phase, true modulation, and retrieved wrapped phase are shown in Figs. 6(a)–6(d), respectively. The phase is designed to have a wider spectrum with no carrier fringes.

Unwrapping results with transform-based methods in case A are shown in Fig. 7. Figure 7(a) shows that FT as a global method is not able to reduce noise in high frequency regions. Figures 7(b) and 7(c) demonstrate good results by using two local methods, WFF and WFR, because of their good noise reduction abilities. Figure 7(d) shows the unwrapping result by using the WT method. The WT method is not suitable in low-frequency regions that require a larger window. As shown in Fig. 7(d), the phase unwrapping error is mainly in the low-frequency regions. Similar results are observed in case B.

To summarize, under noisy condition, transform-based methods show better performances because they provide both denoised wrapped phases and reasonable quality maps.

D. Comparison of Different Quality Maps for Discontinuous Phase Maps

1. *Generation of Discontinuous Phase Maps*

Another problem in phase unwrapping is that the true phase distribution is not continuous. For comparison of influence of discontinuity, eight frames of phase-shifted fringe patterns of a coffee cup cover are captured using FPP and shown in Fig. 8(a). Figures 8(b) and 8(c) show the retrieved wrapped phase and modulation maps, respectively. The unwrapped phase map with temporal phase unwrapping technique is shown in Fig. 8(d) for benchmarking.

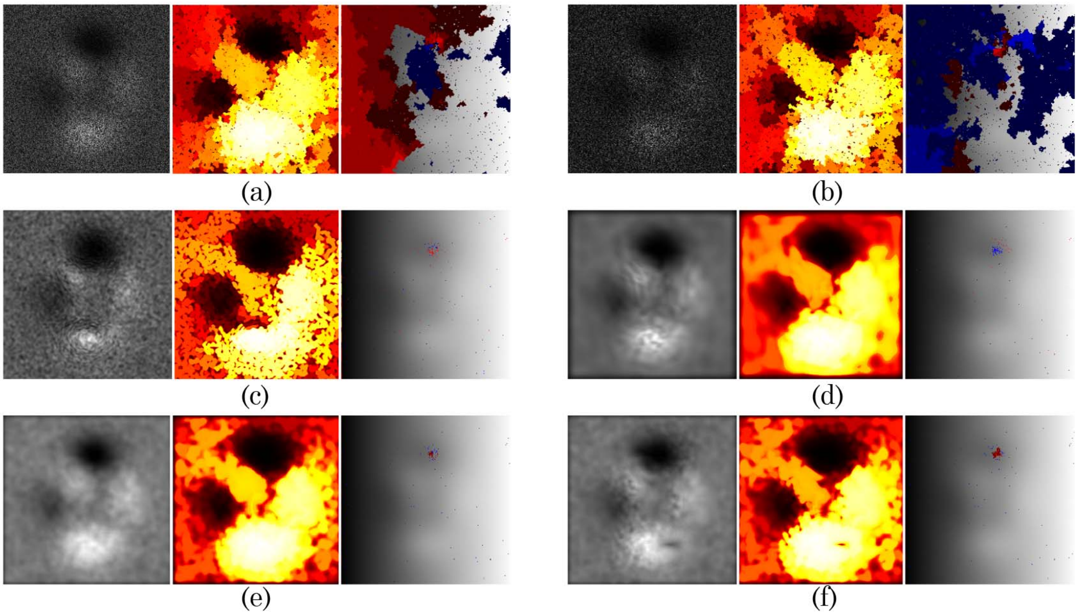

Fig. 4. (Color online) Comparison of phase unwrapping from noisy phase maps. The quality maps used are (a) Q_{MOD}, (b) Q_{REL}, (c) Q_{FT}, (d) Q_{WFF}, (e) Q_{WFR}, and (f) Q_{WT}, respectively. From left to right, each group shows quality maps, path maps, and unwrapping error maps. (In unwrapping error maps, correctly unwrapped pixels are shown in gray scale and incorrectly unwrapped pixels are shown in red or blue. Red and blue indicate that the unwrapped phase value is larger and smaller than its true value, respectively. Darker shades indicate larger errors.)

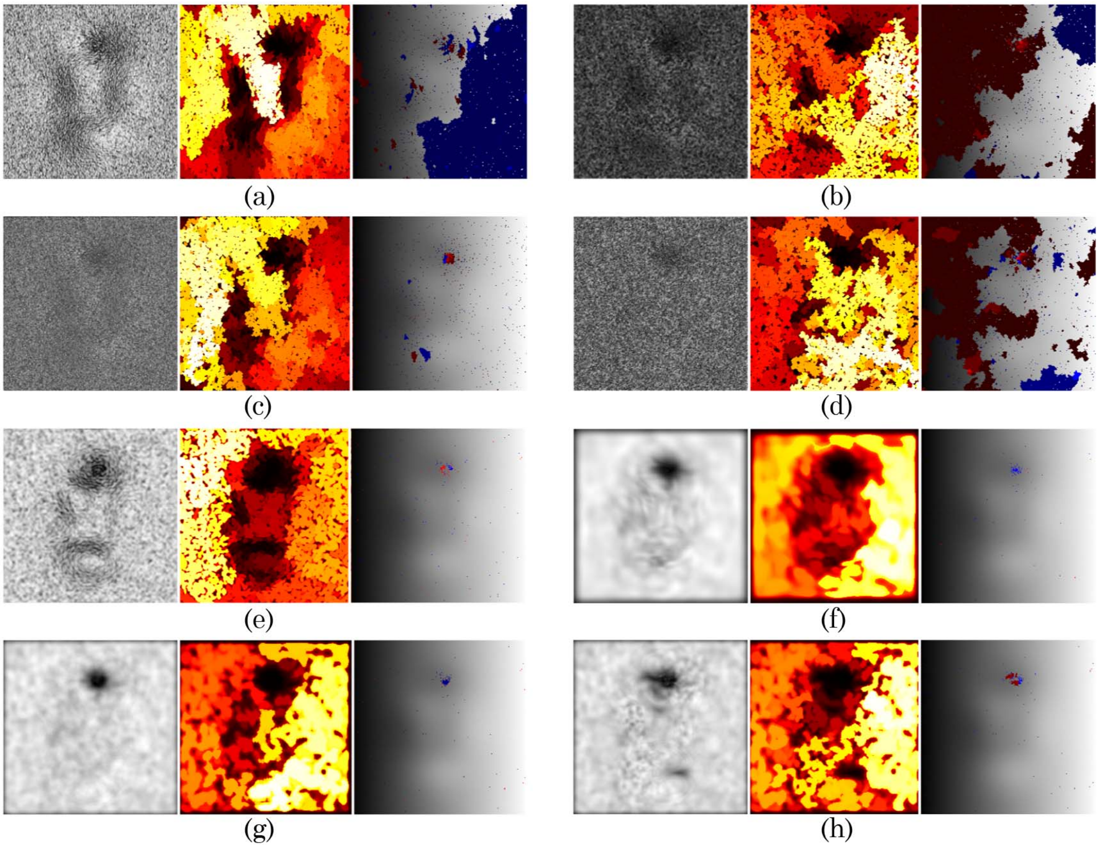

Fig. 5. (Color online) Comparison of phase unwrapping from noisy phase maps. The quality maps used are (a) Q_{PCC}, (b) Q_{PDV}, (c) Q_{FPD}, (d) Q_{SPD}, (e) Q_{FT}, (f) Q_{WFF}, (g) Q_{WFR}, and (h) Q_{WT}, respectively. From left to right, each group shows quality maps, path maps, and unwrapping error maps. The meanings of colors in the unwrapping error maps are the same as in Fig. 4.

2. Comparison of Quality Maps in Case A

Six quality maps are compared for discontinuous phase in case A. The results are shown in Fig. 9, which indicate that the phase unwrapping procedure for this discontinuous example is successful with assistance of quality maps of Q_{MOD}, Q_{REL}, Q_{FT}, and Q_{WFF}. The ridges based quality maps of Q_{WFR} and Q_{WT} fail in leading the unwrapping process with directly passing through discontinuity.

3. Comparison of Quality Maps in Case B

Eight quality maps are compared for discontinuous phase in case B. The results are shown in Fig. 10, which indicate that quality maps of Q_{PDV}, Q_{FPD}, Q_{SPD}, Q_{FT}, and Q_{WFF} and Q_{WFR} successfully guide phase unwrapping in this experiment. Q_{PCC} and Q_{WT} fail in leading the unwrapping process with directly passing through discontinuity.

4. Failure of All Quality Maps

In presence of discontinuity, it is hard to draw a conclusion as to which quality map can best guide the phase unwrapping process. Figure 11 shows an example that all quality maps fail.

Phase unwrapping under discontinuity condition is thus still a challenging problem and unresolved using QGPU alone.

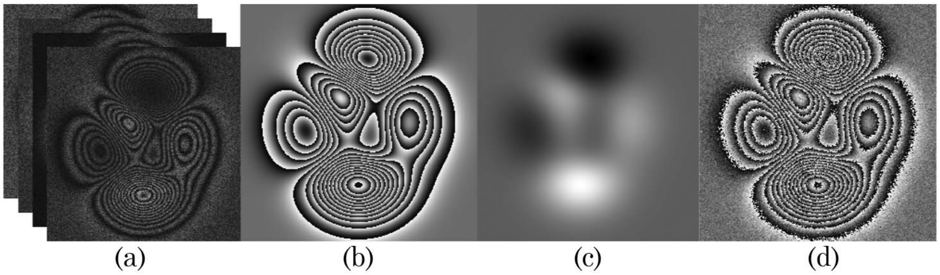

Fig. 6. Phase unwrapping simulation with wide spectrum and no carrier fringes. (a) Phase-shifted fringe patterns, (b) wrapped phase without noise, (c) modulation distribution, and (d) retrieved wrapped phase with phase noise.

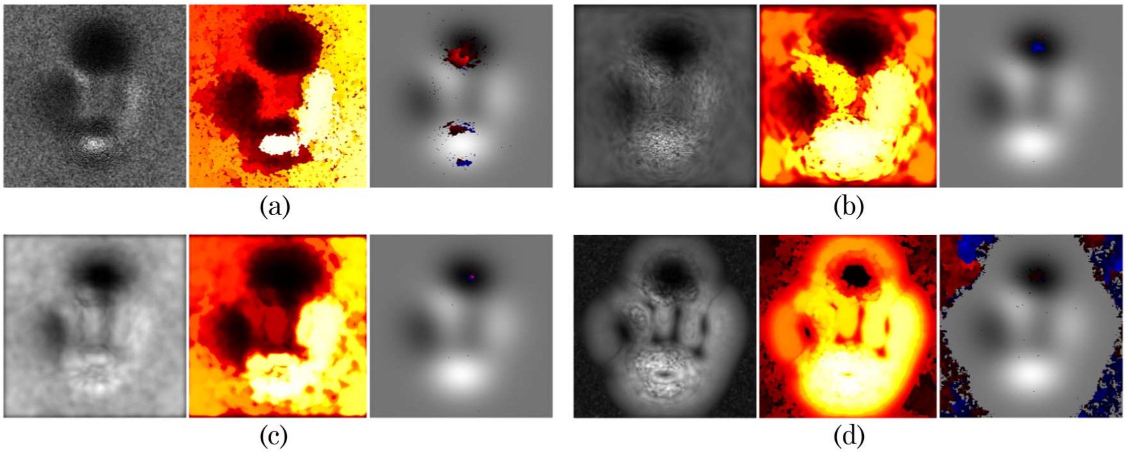

Fig. 7. (Color online) Comparison of unwrapping results from transform-based methods in case A. The quality maps used are (a) Q_{FT}, (b) Q_{WFF}, (c) Q_{WFR}, and (d) Q_{WT}, respectively. From left to right, each group shows quality maps, path maps, and unwrapping error maps.

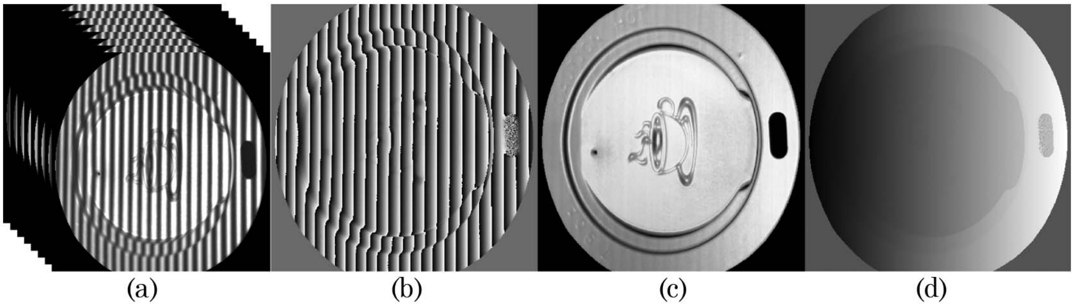

Fig. 8. Influence of fringe discontinuity on phase unwrapping. (a) Eight frames of phase-shifted fringe pattern using FPP, (b) retrieved wrapped phase, (c) retrieved modulation map, (d) unwrapped phase by using temporal phase unwrapping technique.

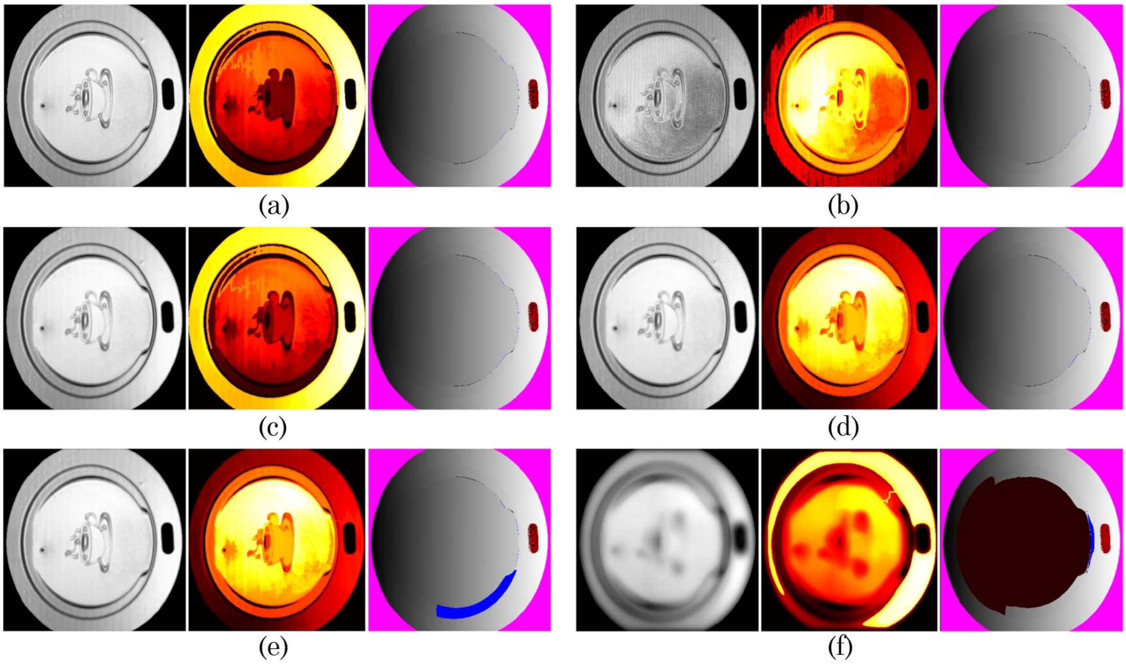

Fig. 9. (Color online) Comparison of phase unwrapping under discontinuity condition in case A. The quality maps used are (a) Q_{MOD}, (b) Q_{REL}, (c) Q_{FT}, (d) Q_{WFF}, (e) Q_{WFR}, and (f) Q_{WT}, respectively. From left to right, each group shows quality maps, path maps, and unwrapping error maps.

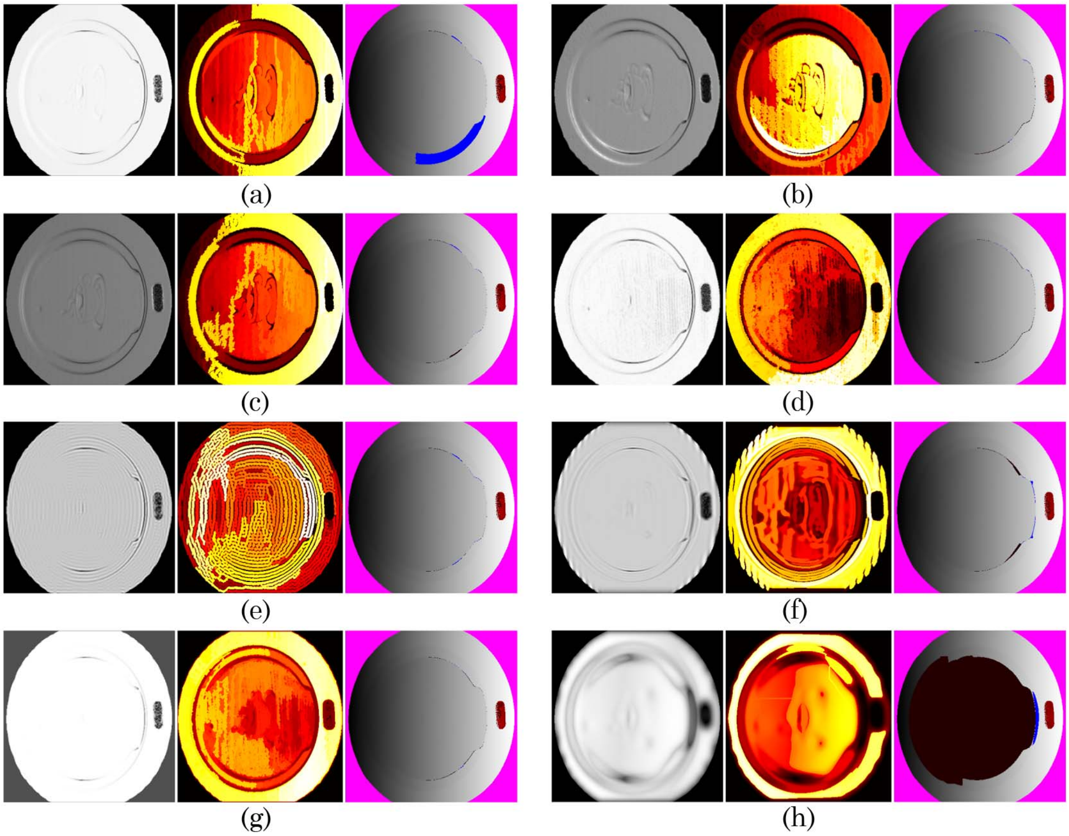

Fig. 10. (Color online) Comparison of phase unwrapping under discontinuity condition in case B. The quality maps used are (a) Q_{PCC}, (b) Q_{PDV}, (c) Q_{FPD}, (d) Q_{SPD}, (e) Q_{FT}, (f) Q_{WFF}, (g) Q_{WFR}, and (h) Q_{WT}, respectively. From left to right, each group shows quality maps, path maps, and unwrapping error maps.

4. Guiding Strategies

A good quality map helps to achieve correct phase unwrapping, while a good guiding strategy helps to reduce the phase unwrapping time. In this section, different guiding strategies are introduced, followed by a comparison of their performances.

A. Introduction to Guiding Strategies

1. *Classical Quality Guiding Strategy*

The simplest and classical guiding strategy was described in Section 2.C. However, it is time-consuming to process the pixels in the adjoin list. The pixels in the adjoin list can be unordered or ordered according to their quality values. If the pixels are unordered, to maintain the adjoin list needs three operations: inserting a new pixel, searching the pixel with the highest quality value to unwrap, and removing the unwrapped pixel. On the other hand, if the pixels are ordered, there are also three operations: locating the inserting position for a new pixel, inserting the new pixel, and removing the pixel with the highest quality value. The ordered list only needs to visit a part of the adjoin list to find the inserting position, while the unordered one needs to traverse the whole list to find the highest quality value. Thus the ordered adjoin list is preferred and used in this paper by default.

The adjoin list can be implemented by several data structures. Figure 12(a) shows the data structure of an array [39]. The array is stored in a continuous memory area and it is very inefficient to insert or remove pixels due to memory reallocation. Figure 12(b) shows the data structure of a linked list (LL) [39]. A node of a linked list contains pixel properties and a pointer to the next node according to the order of quality values. Unlike an array, an LL is very efficient to insert or remove pixels by modifying the pointers. However, traversing the whole LL is still not efficient because the LL is usually very long.

One possible solution to improve the efficiency of LL is to cut the LL into several sections based on the quality values. Each section can be viewed as a "mini-LL", while the whole structure forms array of pointers and each pointer points to a mini-LL. This data structure is called an indexed linked list (ILL) and is shown in Fig. 12(c), where quality values decrease from an upper row to a lower row and also decrease from left to right in each mini-LL. The ILL is efficient in inserting and removing pixels, similar to an LL, but its traversal is much faster as the length of each section is much shorter than that of an LL.

For each mini-LL in an ILL, rather than dynamically allocating and releasing memory for each pixel,

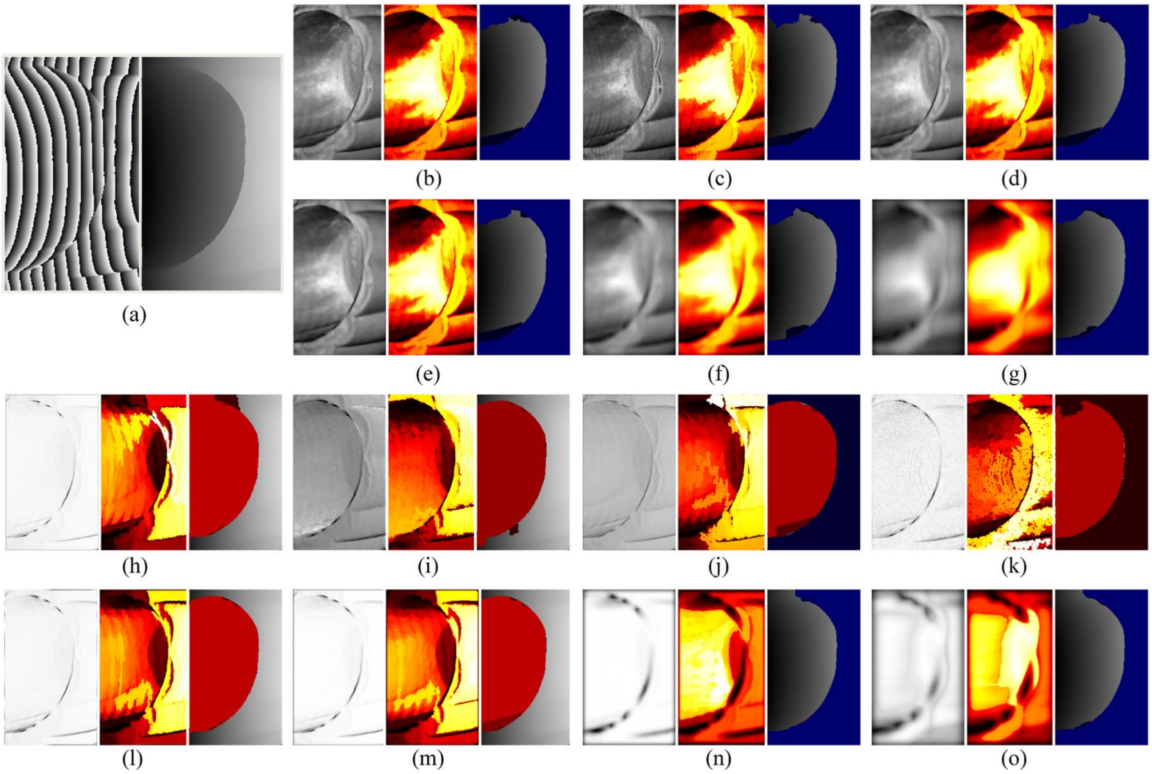

Fig. 11. (Color online) Failure of QGPU. (a) Wrapped phase and unwrapped phase with temporal phase unwrapping and unwrapping result under case A (b)–(g) and case B (h)–(o). The quality maps used are (b) Q_{MOD}, (c) Q_{REL}, (d) Q_{FT}, (e) Q_{WFF}, (f) Q_{WFR}, (g) Q_{WT}, (h) Q_{PCC}, (i) Q_{PDV}, (j) Q_{FPD}, (k) Q_{SPD}, (l) Q_{FT}, (m) Q_{WFF}, (n) Q_{WFR}, and (o) Q_{WT}, respectively. From left to right, each group shows quality maps, path maps, and unwrapping error maps.

sufficient memory can be preallocated for all pixels to reduce time consumption. However, the size of a mini-LL cannot be predetermined and in the worst cases its size would be the whole wrapped phase map. Fortunately, these mini-LLs can be interwoven in the preallocated array as illustrated in Fig. 12(d). Pointers can distinguish these mini-LLs without ambiguity. The array is one-dimensional (1D). The 1D index of an element of the array can be converted to the 2D location of a pixel in the wrapped phase map. The data structure is thus called interwoven indexed linked list (I2L2) and is shown in Fig. 12(d). From our practical experiences, section number in ILL and I2L2 is recommended to be the square root of pixel number of a phase map. To the best of our knowledge, the data structures ILL and I2L2 are proposed by the authors of this paper, with the hint from the data structure for stack-chain guiding strategy which is introduced in Section 4.A.3.

In addition, an algorithm called TRIM [28], which splits the original list into two sections if it is too long, is discussed and implemented. One section is still called the adjoin list and the other one is called the postponed list. Once the adjoin list is empty,

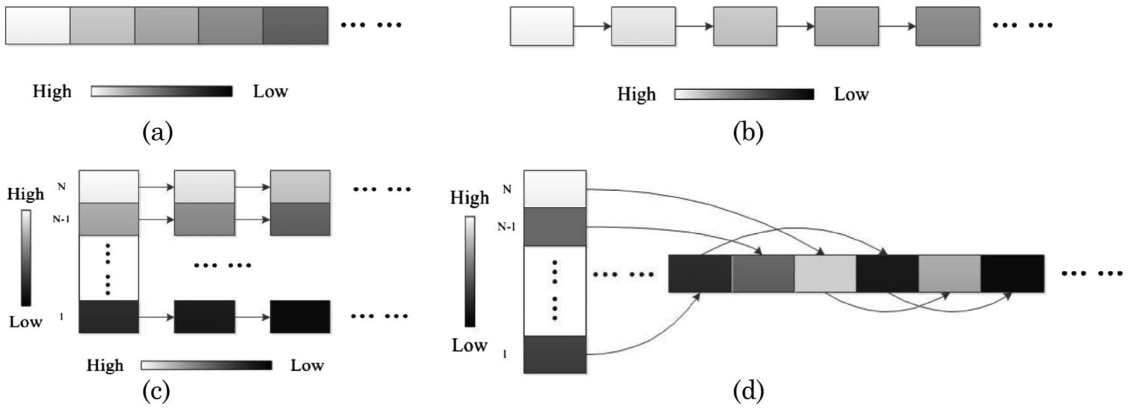

Fig. 12. Data structures for sorting in quality guiding. (a) Array, (b) linked list, (c) indexed linked list, and (d) indexed interwoven linked list.

Table 1. Speed Comparison for Different Data Structures in Classical Quality Guiding (C++)

	Data Structures	Array	LL	ILL	I2L2	TRIM
Time (s)	Box (960 × 1280) in Fig. 1	33.441	14.437	0.813	0.595	2.749
	Noisy Peaks (256 × 256) in Fig. 3	1.762	0.831	0.051	0.037	0.070
	Coffee Cup Cover (400 × 400) in Fig. 8	2.687	1.214	0.069	0.045	0.128

higher quality pixels from the postponed list will be inserted into it.

A comparison of the speed for different data structures is carried out with a result shown in Table 1. The performances from the worst to the best are given by an array, an LL, an ILL, and an I2L2, which is consistent with our previous analysis. TRIM performs better than an array and an LL, but is worse than the ILL and the I2L2.

2. *Two-Section Guiding Strategy*

In the classical quality guiding strategy, each pixel's quality value is compared with others before it is unwrapped. However, pixels with high-quality values usually give correct results even without quality comparison. A two-section guiding strategy is thus proposed: the high-quality pixels form a high-quality section without any comparison; the remaining pixels still need the quality comparison and form a low-quality section which can adopt the data structure of I2L2. Since only some of the pixels need to be ordered, higher processing speed is expected. However a threshold level has to be set to separate the two sections, which can be tricky. The classical quality guiding strategy introduced in Section 4.A.1 can be seen as a special case of the two-section guiding strategy when the threshold is high enough so that the high-quality section is empty. Experimental results with different thresholds are shown in Table 2. The improvement is not significant: to guarantee a zero-error rate, an improvement of only 0.1 s can be obtained. If a 1% error rate can be tolerated then the speed can be doubled.

3. *Stack-Chain Guiding Strategy*

Since the data structure in Fig. 12(c) has many sections and each section consists of pixels with similar quality values, the ordering of pixels in each section can be avoided to reduce the processing time. This idea was proposed in [15,40] and also formed through personal communication [41]. In each section, since no ordering is needed, the concept of a stack with simple “first in, last out” data management can be adopted. Similar to Fig. 12(d), many stacks can be interwoven and the technique is called stack-chain guiding strategy [41]. Empirically the section number is set as the square root of pixel number of a phase map. For the Box example in Table 2, which has 960 rows and 1,280 columns, the section number is $(960 \times 1280)^{1/2} \approx 1109$.

B. Comparison and Discussion

To evaluate the performance of all the strategies discussed above, three wrapped phase maps, Box in Fig. 1, Noisy Peaks in Fig. 3, and Coffee Cup Cover in Fig. 8, are used. Four methods, TRIM, I2L2, two-section, and stack-chain, are implemented in two programming languages, MATLAB and C++. These two languages are selected due to their popularity. MATLAB is preferred by many researchers for algorithm design and C++ is widely used in industry. MATLAB is an interpreted language and C++ is a compiled language. The former is usually slower than the latter. For the two-section strategy, three thresholds, 25%, 50%, and 75% of the largest quality value, are used to separate two sections. The results are shown in Tables 3–5.

From these results, it can be summarized that,

i. I2L2 provides the most reasonable and acceptable result as all the pixels are ordered. The computation time is also short due to the effective data structure. TRIM gives the same unwrapping results but it is slower.

ii. Stack-chain strategy is fastest among all strategies. However, it is not error-free, although the error is often small. This algorithm is most useful when the speed is an essential criterion.

Table 2. Threshold Level in Two-Section Guiding of Box (960 × 1280) in Fig. 1 (C++)

Threshold (Percentage of the highest quality)	Time (s)	Error Rate (%)
10%	0.202	0.0994
20%	0.281	0.1085
30%	0.340	0.4278
40%	0.399	0.0139
50%	0.445	0.0008
60%	0.487	0
70%	0.527	0
80%	0.566	0
90%	0.593	0
100%	0.595	0

Table 3. Comparison Results of Box (960 × 1280) in Fig. 1

		Time (s)		
		MATLAB	C++	Error Rate (%)
CQ	TRIM	65.226	2.749	0
	I2L2	19.406	0.595	0
Two-section	75%	18.323	0.548	0
	50%	16.462	0.445	0.0008
	25%	11.914	0.305	0.1152
Stack-chain		1.007	0.162	0.003

Table 4. Comparison Results of Noisy Peaks (256 × 256) in Fig. 3

		Time (s)		
		MATLAB	C++	Error Rate (%)
CQ	TRIM	2.335	0.070	0
	I2L2	0.705	0.037	0
Two-section	75%	0.692	0.037	0
	50%	0.660	0.035	0
	25%	0.540	0.030	0.4486
Stack-chain		0.056	0.010	0.0656

Table 5. Comparison Results of Coffee Cup Cover (400 × 400) in Fig. 8

		Time (s)		
		MATLAB	C++	Error Rate (%)
CQ	TRIM	4.783	0.128	0
	I2L2	1.936	0.045	0
Two-section	75%	1.821	0.042	0
	50%	1.492	0.031	0.0006
	25%	1.431	0.029	22.5375
Stack-chain		0.102	0.025	0.2338

iii. The two-section strategy is less attractive with regard to both error rate and speed. However, if a batch of phase maps with similar properties is to be unwrapped, a proper threshold can usually be predetermined. In this situation, there is a possibility that the two-section strategy is able to unwrap the phase maps with zero-error rate but faster than I2L2. Another situation is when the majority of pixels in a wrapped phase map have high quality, and the two-section approach can also quickly unwrap the phase without error. However, it can be noticed that the gain of speed is not much.

5. Conclusions

The QGPU method is discussed with regard to two aspects—the quality map and the guiding strategy. A good quality map is important for successful phase unwrapping. For noisy phase maps, transform-based methods perform better but with longer processing time. The parameter selection in transform-based methods reduces automation of QGPU. When discontinuities are present, QGPU alone is not always successful. A guiding strategy is the other important aspect in the phase unwrapping process. Three strategies, classical, two-section and stack-chain, are discussed. The classical quality guiding strategy appears to be best and can be accelerated by a proper data structure such as the interwoven index linked list. The two-section method runs faster, but its success depends on a threshold. The stack-chain method is fastest but is not error-free.

We would like to acknowledge Prof. Yong Li in Zhejiang Normal University, China, and Dr. Zhiling Hou in Sichuan University, China, for their contributions. This work was partially supported by the Singapore Academic Research Fund Tier 1 (RG11/10). We thank the Optics and Photonics Society of Singapore and the Nanyang Technological University for their support.

References

1. H. A. Zebker and R. M. Goldstein, "Topographic mapping from interferometric synthetic aperture radar observations," J. Geophys. Res. **91**, 4993–4999 (1986).
2. M. Takeda and K. Mutoh, "Fourier transform profilometry for the automatic measurement of 3-D object shapes," Appl. Opt. **22**, 3977–3982 (1983).
3. V. Srinivasan, H. C. Liu, and M. Halioua, "Automated phase-measuring profilometry of 3-D diffuse objects," Appl. Opt. **23**, 3105–3108 (1984).
4. L. O. Heflinger, R. F. Wuerker, and R. E. Brooks, "Holographic interferometry," J. Appl. Phys. **37**, 642–649 (1966).
5. L. Ole, "Electronic speckle pattern interferometry," Phys. Technol. **11**, 16 (1980).
6. G. Pedrini, I. Alexeenko, W. Osten, and H. J. Tiziani, "Temporal phase unwrapping of digital hologram sequences," Appl. Opt. **42**, 5846–5854 (2003).
7. J. M. Huntley and H. Saldner, "Temporal phase-unwrapping algorithm for automated interferogram analysis," Appl. Opt. **32**, 3047–3052 (1993).
8. H. Zhao, W. Chen, and Y. Tan, "Phase-unwrapping algorithm for the measurement of three-dimensional object shapes," Appl. Opt. **33**, 4497–4500 (1994).
9. J. Tian, X. Peng, and X. Zhao, "A generalized temporal phase unwrapping algorithm for three-dimensional profilometry," Opt. Lasers Eng. **46**, 336–342 (2008).
10. C. Reich, R. Ritter, and J. Thesing, "White light heterodyne principle for 3D-measurement," Proc. SPIE **3100**, 236–244 (1997).
11. R. M. Goldstein, H. A. Zebker, and C. L. Werner, "Satellite radar interferometry: Two-dimensional phase unwrapping," Radio Sci. **23**, 713–720 (1988).
12. L. Hock, W. Xu, and X. Hu, "Two new practical methods for phase unwrapping," in *Geoscience and Remote Sensing Symposium* (IEEE, 1995), Vol. 191, pp. 196–198.
13. T. J. Flynn, "Two-dimensional phase unwrapping with minimum weighted discontinuity," J. Opt. Soc. Am. A **14**, 2692–2701 (1997).
14. D. C. Ghiglia and L. A. Romero, "Minimum Lp-norm two-dimensional phase unwrapping," J. Opt. Soc. Am. A **13**, 1999–2013 (1996).
15. A. Asundi and Z. Wensen, "Fast phase-unwrapping algorithm based on a gray-scale mask and flood fill," Appl. Opt. **37**, 5416–5420 (1998).
16. X. Su and W. Chen, "Reliability-guided phase unwrapping algorithm: A review," Opt. Lasers Eng. **42**, 245–261 (2004).
17. S. Zhang, X. Li, and S.-T. Yau, "Multilevel quality-guided phase unwrapping algorithm for real-time three-dimensional shape reconstruction," Appl. Opt. **46**, 50–57 (2007).
18. W.-S. Li and X.-Y. Su, "Phase unwrapping algorithm based on phase fitting reliability in structured light projection," Opt. Eng. **41**, 1365–1372 (2002).
19. M. A. Herráez, D. R. Burton, M. J. Lalor, and M. A. Gdeisat, "Fast two-dimensional phase-unwrapping algorithm based on sorting by reliability following a noncontinuous path," Appl. Opt. **41**, 7437–7444 (2002).
20. Y. Lu, X. Wang, and G. He, "Phase unwrapping based on branch cut placing and reliability ordering," Opt. Eng. **44**, 055601 (2005).
21. S. Li, W. Chen, and X. Su, "Reliability-guided phase unwrapping in wavelet-transform profilometry," Appl. Opt. **47**, 3369–3377 (2008).

22. Q. Kemao, W. Gao, and H. Wang, "Windowed Fourier-filtered and quality-guided phase-unwrapping algorithm," Appl. Opt. **47**, 5420–5428 (2008).
23. M. Ramji and K. Ramesh, "Adaptive quality guided phase unwrapping algorithm for whole-field digital photoelastic parameter estimation of complex models," Strain **46**, 184–194 (2010).
24. Q. Kemao, W. Gao, and H. Wang, "Windowed Fourier filtered and quality guided phase unwrapping algorithm: on locally high-order polynomial phase," Appl. Opt. **49**, 1075–1079 (2010).
25. H. Zhong, J. Tang, S. Zhang, and M. Chen, "An improved quality-guided phase-unwrapping algorithm based on priority queue," IEEE Geosci. Remote Sens. Lett. **8**, 364–368 (2011).
26. H. Wang, J. Weaver, I. Perreard, M. Doyley, and K. Paulsen, "A three-dimensional quality-guided phase unwrapping method for MR elastography," Phys. Med. Biol. **56**, 3935–3852 (2011).
27. C. V. Jakowatz, D. E. Wahl, P. H. Eichel, D. C. Ghiglia, and P. Thompson, *Spotlight-Mode Synthetic Aperture Radar: A Signal Processing Approach* (Kluwer Academic, 1996).
28. D. C. Ghiglia and M. D. Pritt, *Two-Dimensional Phase Unwrapping: Theory, Algorithms, and Software* (Wiley, 1998).
29. M. D. Pritt, "Phase unwrapping by means of multigrid techniques for interferometric SAR," IEEE Trans. Geosci. Remote Sens. **34**, 728–738 (1996).
30. D. J. Bone, "Fourier fringe analysis: the two-dimensional phase unwrapping problem," Appl. Opt. **30**, 3627–3632 (1991).
31. J. A. Quiroga, A. González-Cano, and E. Bernabeu, "Phase-unwrapping algorithm based on an adaptive criterion," Appl. Opt. **34**, 2560–2563 (1995).
32. Q. Kemao, "Two-dimensional windowed Fourier transform for fringe pattern analysis: Principles, applications and implementations," Opt. Lasers Eng. **45**, 304–317 (2007).
33. Y. Lu, X. Wang, X. Zhong, G. He, Y. Liu, and D. Zheng, "A new quality map for quality-guided phase unwrapping," Chin. Opt. Lett. **2**, 698–700 (2004).
34. D. J. Bone, H. A. Bachor, and R. J. Sandeman, "Fringe-pattern analysis using a 2-D Fourier transform," Appl. Opt. **25**, 1653–1660 (1986).
35. X. Su and W. Chen, "Fourier transform profilometry: A review," Opt. Lasers Eng. **35**, 263–284 (2001).
36. Q. Kemao, "Windowed Fourier transform for fringe pattern analysis," Appl. Opt. **43**, 2695–2702 (2004).
37. Q. Kemao, H. Wang, and W. Gao, "Windowed Fourier transform for fringe pattern analysis: theoretical analyses," Appl. Opt. **47**, 5408–5419 (2008).
38. Z. Wang and H. Ma, "Advanced continuous wavelet transform algorithm for digital interferogram analysis and processing," Opt. Eng. **45**, 045601 (2006).
39. R. Sedgewick, *Algorithms in C: Parts 1–4, Fundamentals, Data Structures, Sorting, and Searching* (Addison-Wesley, 1997), p. 702.
40. Y. Li and X.-Y. Su, "Fast algorithm for reliability-guided phase unwrapping," Guangdian Gongcheng/Opto-Electronic Engineering **32**, 76–79 (2005).
41. Y. Li and Z. Hou (personal communications, 2011).