A partition strategy to speedup Goldstein's phase unwrapping algorithm on a multi-core architecture

Una estrategia de partición para acelerar el algoritmo de desenvolvimiento de fase de Goldstein sobre una arquitectura multi-núcleo

Abel López-Ocaña[1], William Cruz-Santos[1], Anmi García-Arellano[2], and Juvenal Rueda-Paz Abe_lo_pc@hotmail.com, wdelacruzd@uaemex.mx, anmigarcia@ecosur.mx, jruedap@uaemex.mx
[1] CU-UAEM Valle de Chalco, Hermenegildo Galeana 3, Valle de Chalco, Estado de México 56615, México
[2] CONACYT-ECOSUR, Unidad Chetumal, Chetumal, Quintana Roo, México

KEYWORDS:

parallel computing, phase unwrapping, OpenMP

ABSTRACT

The two-dimensional phase unwrapping is an important and demanding task in measuring methods where a wrapped phase is retrieved such as in remote sensing applications and interferometry techniques. Among phase unwrapping tech-niques, Goldstein's algorithm is one of the most robust and efficient. In this arti-cle, a partition strategy to parallelize Goldstein's algorithm on a multi-core archi-tecture using the programming languages C and OpenMP is proposed. Experi-mental results, using simulated and real data, show that our proposal can be used for real time applications.

PALABRAS CLAVE:

cómputo paralelo, desenvolvimiento de fase, OpenMP

RESUMEN

El desenvolvimiento de fase en dos dimensiones es una tarea importante y de-mandante en los métodos donde se obtienen mapas de fase envueltos tales como en aplicaciones de percepción remota y técnicas de interferometría. Entre las téc-nicas de desenvolvimiento de fase, el algoritmo de Goldstein es uno de los más robustos y eficientes. En este artículo, se propone una estrategia de partición para obtener una versión paralela del algoritmo de Goldstein sobre una arquitectura multi-núcleo usando los lenguajes de programación C y OpenMP. Resultados experimentales obtenidos con datos simulados y reales muestran que nuestra pro-puesta se puede usar en aplicaciones en tiempo real.

Recibido: 24 Octubre del 2017 • Aceptado: 26 de enero del 2017 • Publicado en línea: 28 de febrero del 2018

1. Introduction

The In measuring techniques such as synthetic aperture radar (SAR), interferometric synthetic aperture radar (INSAR), magnetic resonance imaging (MRI), phase-shifting digital holography (PSDH) and digital fringe projection (DFP), among others, a wrapped phase map is retrieved $\phi_w(x,y)$ which is characterized by its 2π disconti-nuities [1, 2]. The phase unwrapping (PU) problem consists in eliminating these 2π ``jumps" to obtain a continuous or unwrapped phase map $\phi(x,y)$. The wrapped and unwrapped maps are related by the equation $\phi_w(x,y)=\phi(x,y)+2\pi m(x,y)$ where $m(x,y)$ is an integer function to be determined. The PU becomes a challenging prob-lem when the absolute phase differences between adjacent pixels at points other than discontinuities are greater than π. These discontinuities can appear as high-frequency, high amplitude noise, discontinuous phase jumps and regional under-sampling in the wrapped phase [3, 4].

Several PU algorithms exist that can be classified in spatial and temporal: in the first, we find 2π discontinuities in the spatial domain and in the last, it is used a sequence of fringe patterns with different fringe pitches or frequencies in order to determine the fringe orders. Among the most used spatial PU algorithms there is Gold-stein's method [5], quality-guide method [6], Flynn's method [7] and minimum -norm method [8]. On the other hand, there are temporal PU algorithms such as the Huntley-Romero method [9], gray-code method [10] and multi-frequency method [11].

With the demand of real-time measuring applications, an efficient PU algo-rithm is required. A number of authors have proposed to use multi-core and graphics processing unit (GPU) architectures to speed up the unwrapping process. In [12], a parallel version of Goldstein's algorithm is proposed using a GPU architecture for wrapped phase maps of sizes up to 1024x1024 pixels. In [13], a parallel approach of the -norm algorithm is proposed, based on the simulated annealing; it is used to process wrapped phase maps of size up to 3000x3000 pixels. A similar approach was presented in [14], based on the discrete cosine transform and applied to process off-axis holograms.

In this article, a new parallelization of Goldstein's algorithm is proposed on a multi-core architecture. We take advantage of a partition strategy to speed up the unwrapping process for large wrapped phase maps. Experimental evaluation using simulated and real data show that the execution time of our proposal is competitive to the state of the art. The organization of this article is as follows: in Section 2, our partition strategy to parallelize Goldstein's algorithm is presented; in Section 3, the results; and in Section 4, the conclusions.

2. The partition strategy

The Goldstein algorithm consists of three main steps (see Fig. 1): (a) identification of residues, (b) branch-cut placement and (c) integration. The identification of residues finds inconsistencies pixel by pixel in a wrapped phase map; a pixel $p=(x,y)$ is a residue if the sum $q=\Delta_1+\Delta_2+\Delta_3+\Delta_4$ is different from zero, where

$$\Delta_1=\phi_w(x,y+1)-\phi_w(x,y), \quad (1)$$

$$\Delta_2=\phi_w(x+1,y+1)-\phi_w(x,y+1), \quad (2)$$

$$\Delta_3=\phi_w(x+1,y)-\phi_w(x+1,y+1) \text{ and} \quad (3)$$

$$\Delta_4=\phi_w(x,y)-\phi_w(x+1,y). \quad (4)$$

If $q>0$, then p is a residue of positive charge, and it is of negative charge if $q<0$. The objective of the branch cut placement is to place lines between positive and negative residues and if there is no residue of an opposite charge, then the cut can be placed with the image border; the cuts serve as barriers to avoid residues in any route of integration. Given a residue, we look for the closer residue of the oppo-site charge within a search window of radius r, so that, the maximum length of a cut is r. Here, a radius of r=25 pixels is used. The integration adds multiples of 2π to the wrapped phase values to get an approximation of the unwrapped phase $\phi(x,y)$. It is started by selecting an initial pixel, whose unwrapped value corresponds to its

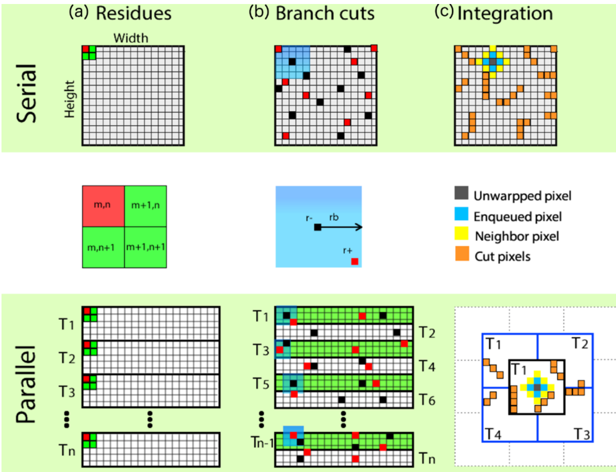

Fig. 1. Serial and parallel process of Goldstein's algorithm: (top) serial process for steps (a) iden-tification of residues, (b) branch-cut placement and (c) integration; (middle) a pixel's neighbor-hood, search region to find an opposite's charge residue, and color notations used in this diagram, respectively; (bottom) partition strategy to parallelize Goldstein's algorithm.

wrapped value; their neighbor pixels are resolved and are inserted in a queue data structure, then a pixel is removed from the queue, their neighbors are resolved and added again to the data structure; this process is repeated until the queue is empty. The pixels that are inserted in the data structure do not have to be residues nor do they have to belong to a cut. Finally, the pixels that are marked ``as cut" are resolved.

We propose a partition strategy of the steps (a)-(b) to parallelize Goldstein's algo-rithm as follows: (a') in the identification of residues, it is proposed to divide the im-age into stripes $T_1, T_2, \ldots, T_n$, the number of partitions will depend on the cores that are available in the processor, creating a thread for each strip that will seek for resi-dues as in the serial version of Goldstein's algorithm; (b') in the branch cut placement, the image is partitioned into stripes $T_1, T_2, \ldots, T_{2n}$ where n is the number of threads that can be created. To balance the largest number of residues, an n number of threads is firstly created to process the even stripes to find branch cuts, last, the odd stripes are processed. In both cases, it is done as in the serial version; and (c') the inte-gration process starts by selecting a square central region T_1 in the image, an initial pixel is selected as starting point and a thread is created to process this region. When all pixels in region T_1 are unwrapped, then a second centered region is selected and divided into sub-regions T_1, T_2, T_3, T_4 of the same size as the initial region T_1. There-fore, a thread is created to process each region; the initial pixels used to start the un-wrapping process for each sub-region, are chosen in the overlapping area with the previous region T_1 . This process continues until the image is fully resolved.

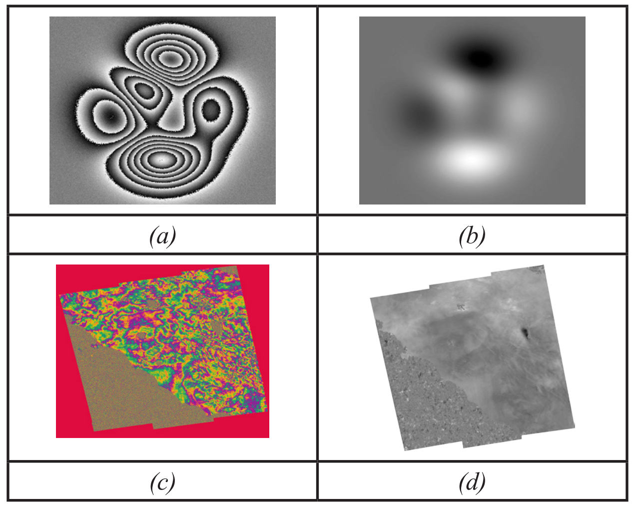

Fig. 2. Simulated and real data: (a) simulated wrapped phase of the peaks distribution with added white noise, (b) unwrapped phase of the peaks distribution, (c) ascending interferogram retrieved by Sentinel-1A/B satellite over the affected area after the 6.2M earthquake in Central Italy on August 24th, 2016 [15], and (d) unwrapped phase of the earthquake interferogram.

3. Experimental evaluation

The experiments were achieved on a desktop computer with the following character-istics: Intel core i7 processor at 3.4 GHz working frequency, 11.7 Gb in RAM, and Linux Mint 17.3 Cinnamon 64 bit operating system. The physical number of cores in this processor are 4 where it is possible to create two threads per core for a total of 8 threads. The programming language that were used are C and OpenMP. Two wrapped phase maps were used for efficient tests: the first one consisted of a simu-lated phase map with white noise added to the peaks distribution. The second one consists of a real phase map obtained by Sentinel-1A/B which was used for INSAR analysis. Figs. 2(a) and 2(c) show the wrapped phase maps used in this research. Table 1 summarizes the results obtained using our parallel implementation of Gold-stein's algorithm. First, the performance of the serial and parallel Goldstein algorithms was evaluated using the peaks phase map for several sizes. In Fig. 2(b), an example of the unwrapped phase for the peaks distribution is shown. Last, the wrapped phase map that was used is shown in Fig. 2(c) of size 7202x5854 pixels. In Fig. 2(d), the unwrapped phase obtained using our parallel algorithm can be seen.

As it can be seen in Table 1, the speedup factor of the simulated data increases with the size of the wrapped phase map. However, for the earthquake real data, the speedup factor is not the expected according to its size since a lot of residues were found in the wrapped phase map. In the last two columns in Table 1, the number of detected residues in the wrapped phase maps are shown for the simulated and real data. Also, the required memory resources of the serial and parallel Goldstein's algo-rithms are shown. In both cases, two dimensional arrays of single-precision floating-point type are used. Fig. 3 shows the branch cuts for the earthquake wrapped phase map, which explains why the speedup factor of the parallel Goldstein algorithm is less than when simulated data is used.

Table 1. Comparison of the execution time between the serial and parallel implementations of Goldstein's phase unwrapping algorithm for simulated and real data. The execution times are given in mi-lliseconds and are averaged over 20 repetitions. The number of threads created are 8 for all tests. The speedup factor corresponds to the ratio between the execution time of the serial and parallel algorithms.

	Image size (pixels)	*Time serial*	*Time parallel*	*speedup*	*Number of residous*	*Used memory (Mbits)*
	128x128	2.07	0.6	3.54	97	0.125
	256x256	5.99	1.78	3.37	22	0.5
Peaks	512x512	23.28	4.95	4.7	42	2
	1024x1024	106.18	13.15	8.07	162	8
	2048x2048	512.36	48.89	10.48	568	32
	4096x4096	2600.51	159.03	16.35	2370	128
Earth-quake	7202x7202	4226.88	359.5	11.76	369381	322

In Table 2, a comparison of the obtained execution times of our proposal are shown with regard to other authors. In Pham et al. [12] and in Backoach et al. [14], a parallel implementation of Goldstein's algorithm and a parallel unweighted least squares phase unwrapping algorithm are proposed, respectively. These proposals use CUDA programming on a GPU architecture, while in our proposal, we use a multi-core architecture.

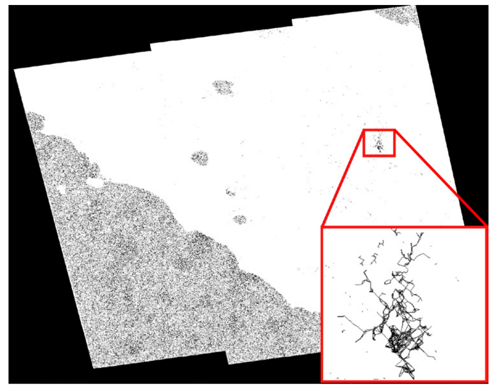

Fig. 3. Branch cuts of the earthquake wrapped phase map shown in Fig. 2. An amplification of a conflicting region is shown in which isolated pixels that will produce a discontinuous unwrapped phase can be observed.

Table 2. Comparison of our proposal with reported results of other authors. The only common characteristic in this comparison among two parallel proposals of unwrapping algorithms is the used image size.

Image size	*Pham [12]*	*Backoach [14]*	*proposed*
256x256	-	0.51	1.78
512x512	2.072	1.7	4.95
1024x1024	11.719	7.78	13.15
2048x2048	-	28.7	48.89

4. Conclusions

A parallel version of Goldstein's algorithm has been proposed on a multi-core archi-tecture that takes advantage of a partition strategy to speedup the unwrapping pro-cess. Our parallel algorithm im-plementation has been validated using simulated phase data of the peaks distribution and real data of an earthquake interferogram obtained of INSAR applications. The obtained experimental results show that the running time of the parallel algorithm is competitive with the state of the art. Alt-hough, it was not possible to compare with other authors since the used computer systems to perform the validation do not have the same characteristics.

Using linear interpolation, one can notice that the execution time of the par-allel algorithm for VGA resolution (640x480 pixels) is of 4.6645 ms which gives us a frame rate of 214.38 images per second. This frame rate is superior for most of the real-time visualization applications using VGA resolution. It is expected that the effi-ciency of our proposal will improve if more computational resources are used, name-ly, more processing cores. Future research will be to apply our parallel algorithm in remote sensing application, namely, to process the wrapped phase retrieved by SAR and INSAR systems which have important use in the analysis and construction of digital elevation models.

Acknowledgements

The first author acknowledges support from Conacyt through a scholarship to pursue master studies at the CU-UAEM Valle de Chalco.

References

1. J. R. Fienup. Phase retrieval algorithms: a comparison. Appl. Opt., 21(15):2758–2769, Aug1982.

2. Q. Kemao. Windowed Fringe Pattern Analysis. EBL-Schweitzer. SPIE, 2013.

3. D.C. Ghiglia and M.D. Pritt. Two-dimensional phase unwrapping: theory, algorithms, andsoftware. Wiley-Interscience publication. Wiley, 1998.

4. Z. Malacara and M. Servín. Interferogram Analysis For Optical Testing, Second Edition. Optical engineering. CRC Press, 2016.

5. Richard M. Goldstein, Howard A. Zebker, and Charles L. Werner. Satellite radar interferome-try: Two-dimensional phase unwrapping. Radio Science, 23(4):713–720, 1988.

6. Hock Lim, Wei Xu, and Xiaojing Huang. Two new practical methods for phase unwrapping. In Geoscience and Remote Sensing Symposium, 1995. IGARSS '95. 'Quantitative Remote Sensing for Science and Applications', International, volume 1, pages 196–198 vol.1, Jul 1995.

7. T. J. Flynn. Two-dimensional phase unwrapping with minimum weighted discontinuity. J. Opt. Soc. Am. A, 14(10):2692–2701, 1997.

8. Dennis C. Ghiglia and Louis A. Romero. Minimum lp-norm two-dimensional phase unwrap-ping. J. Opt. Soc. Am. A, 13(10):1999–2013, Oct 1996.

9. J. M. Huntley and H. Saldner. Temporal phase-unwrapping algorithm for automated interfero-gram analysis. Appl. Opt., 32(17):3047–3052, Jun 1993.

10. Giovanna Sansoni, Matteo Carocci, and Roberto Rodella. Three-dimensional vision based on a combination of gray-code and phase-shift light projection: analysis and compensation of the systematic errors. Appl. Opt., 38(31):6565–6573, Nov 1999.

11. Yajun Wang and Song Zhang. Superfast multifrequency phase-shifting technique with opti-mal pulse width modulation. Opt. Express, 19(6):5149–5155, Mar 2011.

12. Hoa Pham, Huafeng Ding, Nahil Sobh, Minh Do, Sanjay Patel, and Gabriel Popescu. Off-axis quantitative phase imaging processing using cuda: toward real-time applications. Biomed. Opt. Express, 2(7):1781–1793, Jul 2011.

13. Q. Huang, H. Zhou, S. Dong, and S. Xu. Parallel branch-cut algorithm based on simulated annealing for large-scale phase unwrapping. IEEE Transactions on Geoscience and Remote Sensing, 53(7):3833–3846, July 2015.

14. Ohad Backoach, Saar Kariv, Pinhas Girshovitz, and Natan T. Shaked. Fast phase processing in off-axis holography by cuda including parallel phase unwrapping. Opt. Express, 24(4):3177–3188, Feb 2016.

15. Petar Marinkovic and Yngvar Larsen. Mapping and analysis of the Central Italy Earthquake (2016) with Sentinel-1 A/B interferometry, August 2016.

About the authors

Abel López-Ocaña obtained his Bs in computer engineering from the Universidad Autónoma del Estado de México (UAEMex) and currently is a master student in computer science at the UAEMex.

Anmi García-Arellano obtained a Phd from the Instituto Nacional de Astrofísica, Óptica y Elec-trónica (INAOE). His research interests are digital image processing and optical instrumenta-tion.

William Cruz-Santos obtained a Ms and Phd from the Centro de Investigación y de Estudios Avanzados del IPN (Cinvestav). His research interests are quantum computing, computer vi-sion, computational complexity and applied optics.

Juvenal Rueda-Paz obtained a Phd in mathematics from Universidad Autónoma de Morelos. His research interests are in theoretical physics, special functions, mathematical optics and quan-tum computing.