Optics and Lasers in Engineering 64 (2015) 51–58

Contents lists available at ScienceDirect

Optics and Lasers in Engineering

journal homepage: www.elsevier.com/locate/optlaseng

Phase unwrapping by accumulation of residual maps

Mariano Rivera*, Francisco J. Hernandez-Lopez, Adonai Gonzalez*

Centro de Investigacion en Matematicas AC; Guanajuato 36000, Mexico

ARTICLE INFO

Article history:
Received 15 March 2014
Received in revised form
17 June 2014
Accepted 10 July 2014
Available online 8 August 2014

Keywords:
Phase unwrapping
Optical metrology
Robust regularisation
Multigrid methods
Inverse problems

ABSTRACT

We present a path independent (global) algorithm for phase unwrapping based on the minimisation of a robust cost function. The algorithm incorporates an outlier rejection mechanism making it robust to large inconsistencies and discontinuities. The proposal consists on an iterative incremental scheme that unwraps a sub-estimation of the residual phase at each iteration. The sub-estimation degree is controlled by an algorithm's parameter. We present an efficiently computational multigrid implementation based on a nested strategy: the process is iterated by using multiple resolutions. The proposal's performance is demonstrated by experiments with synthetic and real data, and successfully compared with algorithms of the state of the art.

1. Introduction

Fringe analysis consists on the computation of the phase map from an image, or image set, with a sinusoidal fringe pattern. Such a phase is associated with a physical quantity to measure; for example, the three-dimensional topography of an optical surface. Phase unwrapping is an important process in fringe analysis and therefore an active research topic. Energy minimisation based approaches are among the most robust and efficiently computational techniques for this purpose [1–5]. Algorithms based on the minimisation of robust potentials have shown to be capable of unwrapping discontinuous phase maps [6,7]. Despite this capability, there are two major drawbacks of the energy minimisation based algorithms: their computational efficiency depends on the problem size and the differentiable energies lead one to algorithms with a slow reduction of the low-frequency error's components. In this work we propose a new algorithm for phase unwrapping constructed on the robust method proposed in [6]. Our method preserved the advantages of the original formulation, improves its slow convergence and avoids a reduction of the dynamic range of the solution.

Before to presents a brief review of the method in [6], we introduce our notation and important definitions. Phase unwrapping is an inverse problem which direct one (wrapping) can be stated as follows: let ϕ be a phase map defined in a regular lattice $\mathcal{L}$ and let g be its wrapped version such that

$$g_r = W\phi_r; \tag{1}$$

where r indicates a pixel position in $\mathcal{L}$ and

$$Wz \stackrel{\text{def}}{=} z + 2n\pi \tag{2}$$

is the non-linear wrapping operator, with n an integer such that $Wz \in (-\pi, \pi]$. The W operator has the following properties:

$$\begin{aligned} f &= W^{-1}Wf, \\ Wf &= WWf, \\ W^{-1}f &= W^{-1}W^{-1}f, \\ \sin f &= \sin(Wf); \end{aligned} \tag{3}$$

where W^{-1} denotes the unwrapping operator. We note that in general $f \neq WW^{-1}f$.

The aim of this work is to propose a procedure for implementing the unwrapping operator. Let

$$\mathcal{N}_r = \{s \in \mathcal{L} : \|r - s\|_2 = 1\} \tag{4}$$

be the set of first neighbour pixels to the pixel r. Hence, let r and $s \in \mathcal{N}_r$ be two neighbour pixels in the phase map ϕ, if

$$|\phi_r - \phi_s| < \pi, \tag{5}$$

then the first differences of the true phase can be computed by wrapping the first differences of the wrapped phase [8]:

$$\phi_r - \phi_s = W(g_r - g_s). \tag{6}$$

Since noise and phase jumps may introduce large phase differences such that $|\phi_r - \phi_s| > \pi$, then in [6] is proposed a robust phase unwrapping method based on the minimisation of

$$U(f, \omega; \rho) = \frac{1}{2}\sum_{r \in \mathcal{L}}\sum_{s \in \mathcal{N}_r}\left\{\omega_{rs}^2[(f_r - f_s - \rho_{rs})^2 + \lambda(f_r - f_s)^2] + \mu(1 - \omega_{rs})^2\right\}, \tag{7}$$

* Corresponding authors.
E-mail address: mrivera@cimat.mx (M. Rivera).

http://dx.doi.org/10.1016/j.optlaseng.2014.07.005
0143-8166/

where λ and μ are positive parameters of the algorithm, and we define

$$\rho_{rs} \stackrel{\text{def}}{=} W(g_r - g_s). \tag{8}$$

The data term (first one) in (7) is a weighted version of the regularised least square potential [1,2]. The second term [membrane potential: $(f_r - f_s)^2$] penalises large local variations on the unwrapped phase [3,4]. In the minimisation of (7), the weights $\omega_{rs} \in [0,1]$ are computed by a balance between the data and the membrane terms *versus* the value of the μ parameter; *i.e.*, if the magnitude of, either, the residual $f_r - f_s - \rho_{rs}$ or the phase difference $f_r - f_s$ is too large,

$$(f_r - f_s - \rho_{rs})^2 + \lambda(f_r - f_s)^2 > \mu, \tag{9}$$

then, for reducing the cost function value, the data and membrane terms are discarded by setting $\omega_{rs} = 0$. One can note that the membrane term introduces a smoothing effect on the final unwrapped phase product of the first differences penalisation. However, as was pointed in [3], the use of the membrane potential for noise reduction purposes also reduces the dynamical range of the solution. This effect can be noted if we try to unwrap the phase in Fig. 1(a). Fig. 2 shows the computed unwrapped phase (first row) and the rewrapped phase (second row) by minimising the energy (7) with a Gauss–Seidel scheme after 25,000 and 500,000 iterations; columns (a) and (b), respectively. It is evident the computational inefficiency of the Gauss–Seidel solver for reducing low frequency errors (residuals).

In this work we consider that denoising can be achieved in a pre-processing stage (using a narrowband quadrature filter-bank) or in a post-processing stage (filtering the unwrapped phase). This we focus on the development of an efficiently phase unwrapping method that preserves the dynamic range of the solution. Our proposal is based on a the robust energy (7) and consists on an incremental scheme where we compute an unwrapped approximation of the current residual at each iteration. The computational performance is improved by using an efficient multigrid strategy.

2. Phase unwrapping method by accumulation of residual maps (ARM)

Initially, we assume that an initial unwrapped phase $f^{(0)}$ is available, then our aim is to estimate an unwrapped residual phase δ such that

$$\phi_r = f_r^{(0)} + \delta_r. \tag{10}$$

Hence, if (5) holds, we have from (6) and (10):

$$\delta_r - \delta_s = \tilde{\rho}_{rs}; \tag{11}$$

where we set

$$\tilde{\rho}_{rs} = \rho_{rs} - (f_r^{(0)} - f_s^{(0)}) \tag{12}$$

with ρ_{rs} defined in (8). Since noise and phase jumps may introduce large phase differences such that $\exists(r,s) : |\phi_r - \phi_s| > \pi$, then (6) can be contravened by some first neighbour pixel pairs. Therefore, we propose to compute the residual phase by minimising the robust cost function for the residual phase map:

$$\{\delta, \omega\} = \underset{\delta,\omega}{\operatorname{argmin}}\, U(\delta, \omega; \tilde{\rho}). \tag{13}$$

The solution to (13) can be computed by alternating minimisations with respect to δ and ω [6,9,12]. Thus, if ω is fixed; the solution of the positive-definite diagonal-dominant linear system that results of $\partial U/\partial\delta_r = 0$ can be computed with a Gauss–Seidel iterative scheme:

$$\delta_r = \frac{\sum_{s \in \mathcal{N}_r} \omega_{rs}^2[(\delta_s + \tilde{\rho}_{rs}) + \lambda\delta_s]}{\sum_{s \in \mathcal{N}_r} \omega_{rs}^2[1+\lambda]}. \tag{14}$$

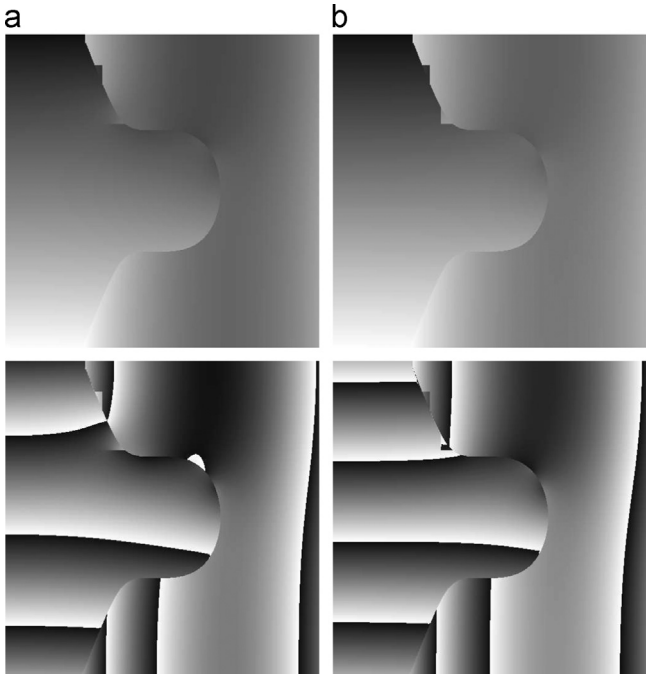

Fig. 2. Unwrapped phases (first row) and rewrapped phases (second row) computed by minimising (7) with a Gauss–Seidel scheme after: (a) 25,000 iterations, computational time = 127.2 s and MSE = 0.669. (b) 500,000 iterations, computational time = 2481.1 s and MSE = 0.241.

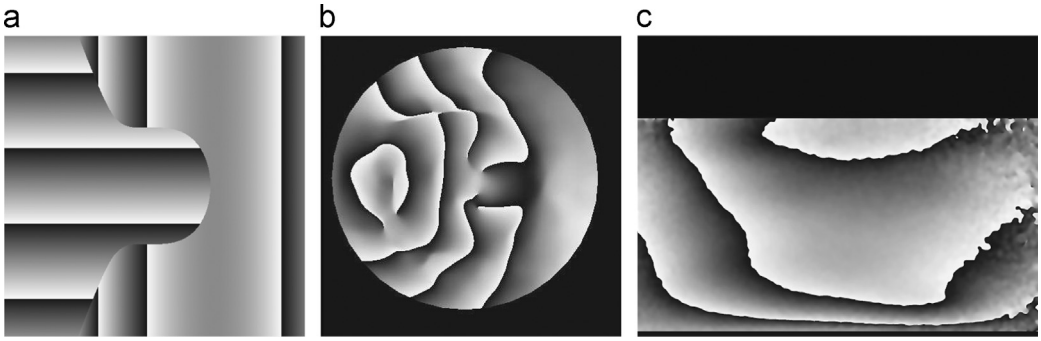

Fig. 1. Test wrapped phases. (a) Phase1 (synthetic), 512 × 512 pixels. (b) Phase2 (cornea), 1920 × 1440 pixels. (c) Phase3 (iron plate), 423 × 423 pixels.

Similarly, from $\partial U/\partial \omega_{rs}=0$, we obtain the closed formula:

$$\omega_{rs}=\frac{\mu}{\mu+(\delta_r-\delta_s-\rho_{rs})^2+\lambda(\delta_r-\delta_s)^2}. \quad (15)$$

The process is resumed in Algorithm 1.

Algorithm 1. Robust unwrapping. Unwraps g with initial condition f; T is the maximum iteration number.

function RobustUnwrap (g,f,λ,μ,T)
  ▷ An initial value $f\neq 0$ can be passed
  Set $\delta_r=0,\forall r$;
  Set $\omega_{rs}=1,\forall(r,s)$;
  ▷ A different initial value for ω can be used
  Compute $\tilde{\rho}_{rs}=W(g_r-g_s)-(f_r-f_s),\forall(r,s)$;
  for $t=1,2,\ldots,T$ **do**
    Update δ, keeping fixed ω, with (14);
    Update ω, keeping fixed δ, with (15);
  end for
  return $f+\delta$;
end function

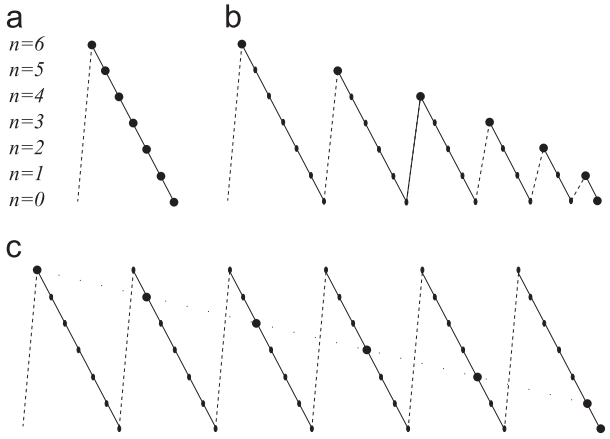

Fig. 3. Multigrid strategy illustration. (a) Simple multigrid, the initial guess is recursively computed by solving a reduced size problem (subsampled), it accelerates the reduction of low frequency residuals. (b) Nested multigrid, the initial guess is recursively computed with a reduction of all the frequency-components of the residual. (c) Full nested multigrid, all the frequency components of the residuals are reduces in the initial guess with emphasis in the low frequency components.

Now, we give a forward step based on the following analysis: suppose that we have approximately solved (13) by, for example, stopping the algorithm iterations before convergence, then the updated phase $f^0+\delta$ can, again, be corrected with the proposed scheme. This iterative upgrade of the partial solution is expressed by

$$f_r^{(n+1)}=f_r^{(n)}+\delta_r^{(n)}; \quad (16)$$

where $\delta^{(n)}$ and $\omega^{(n)}$ are the approximated solution of

$$\{\delta^{(n)},\omega^{(n)}\}=H\{\underset{\delta,\omega}{\operatorname{argmin}}U(\delta,\omega;L\{\tilde{\rho}^{(n)}\})\}. \quad (17)$$

The convergence of this iterative scheme is guaranteed because

$$U(\delta^{(n)},\omega^{(n)};\tilde{\rho}^{(n)})\leq U(\mathbf{0},\mathbf{1};\tilde{\rho}^{(n)}), \quad (18)$$

where U is defined in (7); $\mathbf{0}$ and $\mathbf{1}$ denote phase fields with all the pixels equal zero and one, respectively, with dimensions defined by the context. The operators H and L in (17) are discussed below, for the moment the reader can assume that they are the identity.

Note that minimise $U(\delta,\omega;\tilde{\rho})$ is different from minimise $U(f+\delta,\omega;\rho)$: the first one penalises large differences in the residual phase (smoothing it), meanwhile the second one corresponds to the method in [6] and smoothes the accumulated phase $f+\delta$ with the mentioned dynamic range reduction effect. Conversely, the proposed algorithm unwraps discontinuous phases without a reduction of the phase's dynamic range; the effect of the λ parameter is to keep smooth the computed residual at each iteration and therefore a smooth algorithm convergence. Moreover, as it is well known, Gauss–Seidel is prone to have a slow reduction of low-frequency residuals. Then, we accelerate such a convergence by using a multigrid strategy [10]; this strategy is depicted in Fig. 3(a) and resumed in Algorithm 2. It shows a simple multigrid scheme where the solution at the level n is used as initial guess for the level $n-1$. The nodes represent the partial computation of a robust unwrapped phase given a scaled version of ρ and an initial guess for the unwrapped phase ϕ. Solid lines illustrated a resizing process. Each resizing implies a bicubic interpolation of the current unwrapped phase that will be used as initial guess for the new unwrapping stage at a refined scale. This up-scaling (interpolation) is denoted by the $H\{\cdot\}$ operator in (17). On the other hand, the dashed lines illustrate a subsampling process (up moving in the scale pyramid) of the current unwrapped phase used an initial point for the top level of the unwrapping. The down-scaling operator, denoted by $L\{\cdot\}$, is implemented with a subsampling.

Algorithm 2. Multigrid unwrapping. Multigrid strategy for Algorithm 1 with N scale levels.

function MultigridUnwrap (g,f,λ,μ,N,T)
  if $N>0$ **then**
    $\hat{g}=$ DownSample(g); ▷ Down sampling
    $\hat{f}=$ DownSample(f);
    ▷ Change level
    $\hat{f}=$ MultigridUnwrap$(\hat{g},\hat{f},\lambda,\mu,N-1,T)$
  $f=$ UpSample$(\hat{f})$; ▷ Up sampling
  end if
  ▷ Unwrap current level
  $f=$ RobustUnwrap(g,f,λ,μ,T)
  return f;
end function

Algorithm 3. Nested multigrid unwrapping I. Nested multigrid strategy for Algorithm 2 with N iterations of N scale levels.

function NestMultigridUnwrap (g,λ,μ,N,T)
  $f=0;N_0=N$
  while $N>0$ **do**
    ▷ simple multigrid
    $f'=$ MultigridUnwrap(g,λ,μ,N_0,T);
    ▷ residual wrapped phase
    $f=f+f'$;
    $g=$ residual(g,f);
    $N=N-1$;
  end while
  return $f+g$;
end function

The simple multigrid scheme depicted in Fig. 3(a) is a cost-effective technique for providing an initial guess for the subsequent level in the scale pyramid, given that low frequency error is more efficiently reduced in subsampled (smaller in size) data.

Since we are also interested in unwrapping discontinuous and inconsistent phases, we need to estimate a good initial guess that properly reduces both low and high frequency errors. Thus, a nested multigrid approach implements efficient solvers [10]; the nested multigrid is depicted in Fig. 3(b). In that illustration, we can see that the initial guess for a new level (a peak in the saw-teeth) is estimated by using an approximated solution at all the levels (a complete simple multigrid). Moreover, since the additional stages (top levels in the scale pyramid) have low resolution, their computations consume a small amount of additional time, as benefit we improve the low frequency error reduction. Our proposal implements a full-nested multigrid strategy depicted in Fig. 3(c) and detailed in Algorithm 3.

In the implementation of Algorithm 3 we require of computing the residual wrapped phase at the end of each iteration (node in the multigrid scheme). The residual wrapped phase between two given phases g and f is denoted by

$$\text{residual}(g,f) \stackrel{\text{def}}{=} W(W^{-1}g - W^{-1}f), \tag{19}$$

where W and W^{-1} are the wrapping and unwrapping operators, see (3). Note, however, that (19) is a definition but does not a computational formula: it implies the use of the unwrapping operator, W^{-1}. The definition of an unwrapping procedure is the purpose of the present work. To compute the wrapped residual between two phases we use a formula that does not explicitly require of W^{-1}:

$$\text{residual}(g,f) = \text{atan2}(\sin(g-f), \cos(g-f)), \tag{20}$$

with the identities:

$$\begin{aligned} \sin(g-f) &= \sin g \cos f - \cos g \sin f, \\ \cos(g-f) &= \cos g \cos f + \sin g \sin f. \end{aligned} \tag{21}$$

Algorithm 4. Nested multigrid unwrapping II. Subtracts the mean-plane phase. It is proper for fringe patterns with lineal carriers: as projection setups.

1: **function** NestMultigridUnwrapII (g, λ, μ, N, T)
2: ▷ Substrate dominant plane phase
3: Compute $\theta = [\theta_1, \theta_2]^T$ with (22);
4: $g_r = \text{residual}(g_r, \theta^T r), \ \forall r$;
5: $f = \text{NestMultigridUnwrap}(g, \lambda, \mu, N, T)$;
6: **return** $f + \theta^T r$;
7: **end function**

Finally, we noted that the number of levels in the multigrid algorithm is limited by the frequency of the fringe pattern; *i.e.*, the limit for subsampling a fringe pattern is Nyquist's limit. That lead us to the last algorithm improvement following discussed. Suppose that the wrapped phase corresponds to one obtained from a fringe pattern projection setup, then the projected fringe frequency constrains the number of levels in the multigrid strategy: one needs at least two samples (pixels) per projected period. The solution in this case is to subtract the carrier frequency that corresponds to a constant slope plane, $\theta^T r$; where $\theta = [\theta_1, \theta_2]$ components are the plane slopes in the first and second coordinates, respectively. Such slopes can be estimated as the average of the wrapped first differences along each coordinate:

$$\theta_i = \text{mean}\{W(g_r - g_{r-e_i})\}_{(r, r-e_i) \in \mathcal{L}} \tag{22}$$

for $i = 1,2$; where the operator mean$\{A\}$ return the average of a set A of real numbers, e_i denotes the i-th canonical base vector (*e.g.*, $e_1 = [1,0]^T$) and W is the wrapping operator defined in (2). Algorithm 4 includes the subtraction of a plane phase that could be introduced by a frequency carrier in a fringe projection setup.

Remark. The modification of the algorithms for considering a region of interests $\mathcal{M} \subset \mathcal{L}$ (Mask), different from the whole image domain, can be achieved by redefining the neighbourhood (4) as

$$\mathcal{N}_r = \{s \in \mathcal{M} : \|r - s\|_2 = 1\} \tag{23}$$

and taking it into account at the different levels of the multigrid strategy.

3. Analysis of the proposed robust potential

In order to understood the proposed algorithm behaviour [the potential (2) for the problem (13)], we substitute ω_{rs} [*i.e.*, formula (15)] into the potential $U(\delta, \omega; \rho)$. Then, we have that (13) is equivalent to

$$\underset{\delta}{\text{argmin}} U(\delta; \rho) = \frac{1}{2} \sum_{r \in \mathcal{L}} \sum_{s \in \mathcal{N}_r} P(\delta_r - \delta_s, \tilde{\rho}_{rs}) \tag{24}$$

where

$$P(\delta_r - \delta_s, \tilde{\rho}_{rs}) = \frac{\mu[(\delta_r - \delta_s - \tilde{\rho})^2 + \lambda(\delta_r - \delta_s)^2]}{\mu + (\delta_r - \delta_s - \tilde{\rho})^2 + \lambda(\delta_r - \delta_s)^2}, \tag{25}$$

which can be seen as an extended version of the robust non-convex Geman–McClure potential [11]. The potential P can be analysed in the framework of the theory for M-estimators [9,12] but taking into account the contributions of: the gradient residuals $(\delta_r - \delta_s)$, the data $(\tilde{\rho}_{rs})$ and the regularisation parameter (λ). Fig. 4 depicts the graphs of P as a function of the phase differences $\delta_r - \delta_s$ (x-axis) and data $\tilde{\rho}_{rs}$ (y-axis). The columns correspond to different λ values, the white level indicates largest function values and black levels the lowest ones. The effect on the estimator of a phase difference $\delta_r - \delta_s$ can be observed in the so-called *influence function*, ψ; second row in Fig. 4. This influence function is defined, in terms of the residual phase differences $\delta_r - \delta_s - \tilde{\rho}_{rs}$, as

$$\psi(\delta_r - \delta_s, \tilde{\rho}_{rs}) = \frac{\partial P(\delta_r - \delta_s, \tilde{\rho}_{rs})}{\partial(\delta_r - \delta_s)}. \tag{26}$$

Second row in Fig. 4 shows the respective influence functions for the different λ values. We note that combinations of $\delta_r - \delta_s$ and $\tilde{\rho}_{rs}$ values located at darker regions, of the influence function, have a small effect on the potential and therefore are neglected. For small λ values, the low-influence region is composed by sites where $\delta_r - \delta_s - \tilde{\rho}_{rs} \approx 0$. Thus, as λ increases more smoothness is imposed to residual with small derivatives (indeed, first differences) but residuals with large derivatives have a small effect on the robust potential. As a consequence, if λ takes a relatively large value the ARM method preserves large phase steps.

4. Experiments

We present experimental results using real wrapped phases in order to demonstrate the proposal performance. The tested wrapped phases are shown in Fig. 1. Fig. 5 shows the unwrapped phase corresponding to the synthetic phase in Fig. 1(a). The difficulty of unwrapping such a phase is due to it has regions with smooth phases (low bandwidth) delimited by large phase discontinuities. In order to preserve phase discontinuities, the μ parameter is tuned for such purpose, we used $\lambda = 0.1$ and $\mu = \pi/10$. Moreover, we use as initial condition for the weights (step 4 in Algorithm 1)

$$\omega_{rs} = \frac{\mu}{\mu + [W(g_r - g_s)]^2}, \tag{27}$$

that can be understood as a quality map. The numerical experiment was performed in two conditions: noise-free (Fig. 5) and noisy data (Fig. 6). We compare the performance of the proposal

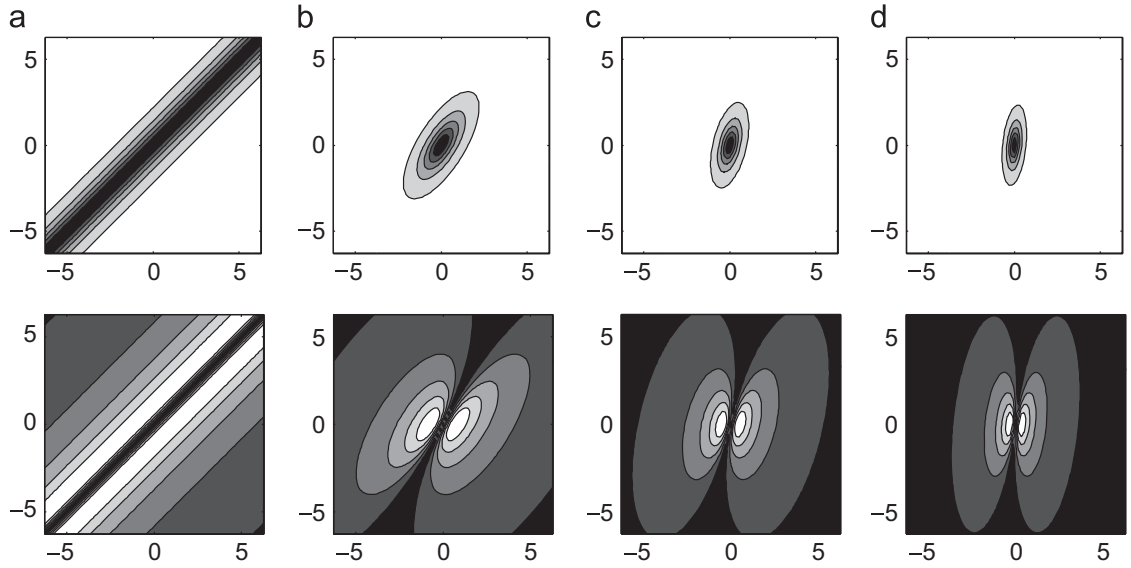

Fig. 4. Effect of the λ parameter of the proposed potential (25). The columns correspond to different λ values. First row depicts the level sets of the potentials and second row depicts the absolute value of the influence function. The x-axis corresponds to $\delta_r - \delta_s$ values and y-axis to $\tilde{\rho}_{rs}$ values. (a) $\lambda = 0$, (b) $\lambda = 1$, (c) $\lambda = 4$, (d) $\lambda = 10$.

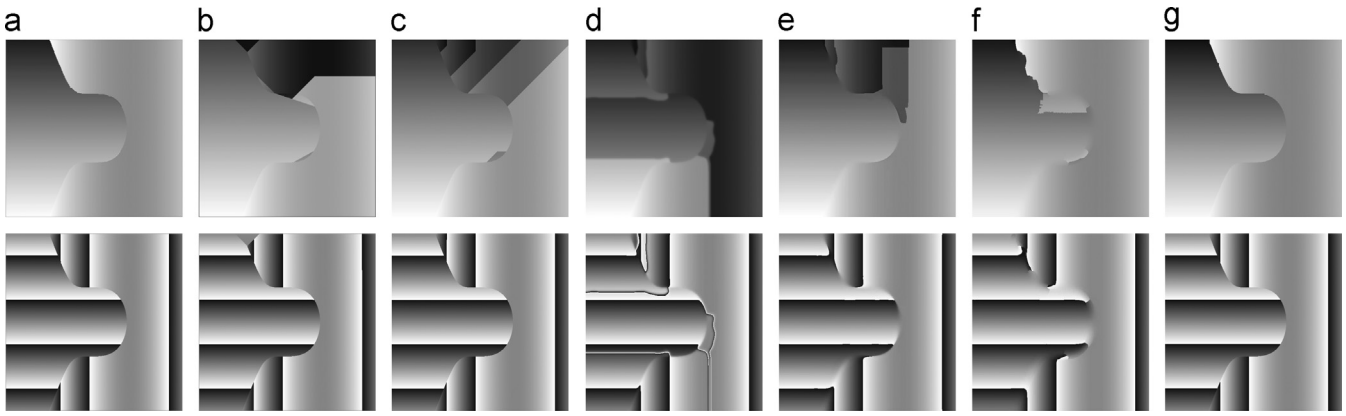

Fig. 5. Unwrapped phases (first row) and rewrapped phases (second row) computed with the evaluated methods. The test phase corresponds to the one in Fig. 1(a) and image size of 512×512 pixels. (a) Quality guided, (b) Goldstein, (c) PUMA, (d) Follower, (e) WFF_QGPU, (f) WFR_QGPU, (g) ARM.

(ARM phase unwrapping method) with algorithms of the state of the art: the Quality Guided [13] (implemented in [14]), the branch-cut based Goldstein's method [15] (implemented in [14]), the unwrapping algorithm based on graph-cut, PUMA [7] (implemented in [16]), the iso-phase follower unwrapping algorithm (named here "Follower") reported in [17] which code is available in [18], WFF-QGPU [19,20] and WFR-QGPU [19,20] (both implemented in [21]). The evaluated algorithms are implemented in Matlab with mexfiles (C-code) except for the: Follower algorithm that is implemented in pure C-code, and WFF-QGPU and WFR-QGPU implemented in pure Matlab with build in functions. In spite of noise and discontinuities induce fails of the phase unwrapping algorithms, the proposed ARM method effectively computes the unwrapped phase. The reconstruction error is shown in Table 1 and the computational times (for the noise-free case) are shown in the second row of Table 2, image size of 512×512 pixels.

Last synthetic experiment allows us to compare in both qualitative and quantitative aspects the performance of ARM. Now we use real data for conducting a qualitative comparison of our proposal performance: the true unwrapped phases are unknown but it is know that they are smooth. Fig. 1(b) shows a wrapped phase corresponding to a Placido's Fringe Pattern (PFP) with large inconsistencies. A PFP is equally spaced monochromatic concentric rings placed in front of a human eye. Then, a camera is placed in a hole at the PFP centre and captures the reflected pattern modulated by the corneal irregularities. The wrapped phase in Fig. 1(b) shows phase inconsistencies produced by an effect of eyelashes, noise and the hole at the PFP centre. The image dimension are 423×423 pixels. On the other hand, Fig. 1(c) shows a wrapped phase corresponding to an Electronic Speckle Pattern Interferometry (ESPI) setup. The purpose is to estimate the deformation of a metallic plate under thermal stress. The wrapped phase has inconsistencies produced by speckle noise; see upper-right section of the plate. The image dimension is 1920×1440 pixels.

Figs. 7 and 8 show the unwrapped phases with the compared algorithms. The phases in Figs. 7 and 8 are smooth. As we do not expect phase discontinuities, the discontinuity sensitivity parameter μ was set large enough: we use $\mu = 1 \times 10^8$ and we keep $\lambda = 0.1$ as in the experiment with discontinuities.

The results in Fig. 7 demonstrate that Quality Guided, Goldstein, PUMA, Follower, WFF_QGPU and WFR_QGPU algorithms [columns (a)–(f)] fail unwrapping phases with large inconsistencies and introduce false phase discontinuities. On the other hand, our ARM method reconstructs the phase correctly. Fig. 8 shows the unwrapped phases with the same algorithms than in Fig. 7 (except

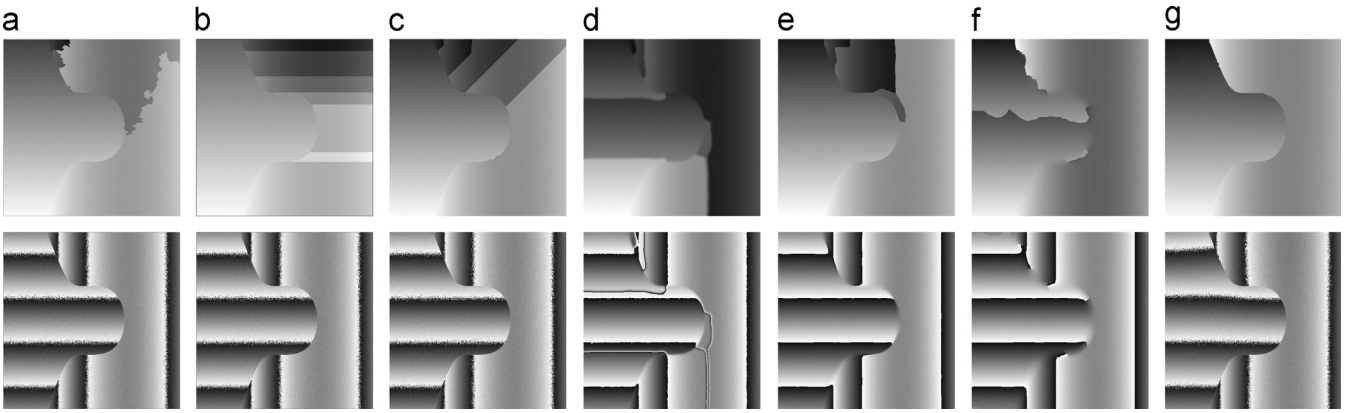

Fig. 6. Unwrapped phases (first row) and rewrapped phases (second row) computed with the evaluated methods. The test phase corresponds to the one in Fig. 1(a) with noise and image size of 512 × 512 pixels. (a) Quality guided, (b) Goldstein, (c) PUMA, (d) Follower, (e) WFF_QGPU, (f) WFR_QGPU, (g) ARM.

Table 1
Comparative performance of the Quality Guided, Goldstein's, PUMA, Follower, WFF_QGPU, WFR_QGPU and Proposed method (ARM). Unwrapped phases in Fig. 5.

Size of image	**Mean square error**						
512 × 512	Quality Guided	Goldstein	PUMA	Follower	WFF_QGPU	WFR_QGPU	ARM
Noise-free	0.056	2.319	1.042	3.837	1.006	0.132	0.008
Noisy	0.583	2.092	1.009	3.835	1.070	0.710	0.011

Table 2
Comparative performance of the Quality Guided, Goldstein's, PUMA, Follower, WFF_QGPU, WFR_QGPU and Proposed method (ARM). Computational time in seconds for unwrapping test phases. Quality Guided and Goldstein algorithms are implemented in Matlab; Follower is implemented in language programming C; PUMA and the proposed algorithms are implemented using Matlab and mexfiles (C-code).

Dimension in pixels	**Computational time (s)**						
	Quality Guided	Goldstein	PUMA	Follower	WFF_QGPU	WFR_QGPU	ARM
423 × 423	229.295	17.160	3.777	1.002	24.378	14.706	2.071
512 × 512	838.773	52.696	11.698	1.006	27.525	19.780	56.007
1920 × 1440	7.1×10^4	[a]	192.373	33.001	254.394	224.897	46.443

[a] Goldstein's algorithm does not converged after 5 days.

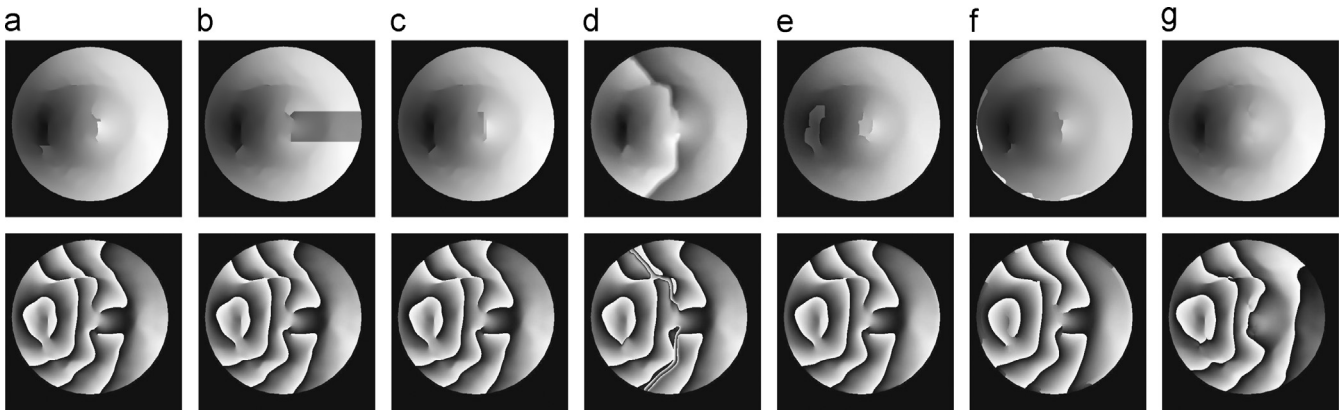

Fig. 7. Unwrapped phases (first row) and rewrapped phases (second row) computed with the evaluated methods. The test phase corresponds to the one in Fig. 1(b) and image size of 423 × 423 pixels. (a) Quality guided, (b) Goldstein, (c) PUMA, (d) Follower, (e) WFF_QGPU, (f) WFR_QGPU, (g) ARM.

with Goldstein because it does not converge after 5 days of computation). We can note that the proposed algorithm effectively unwraps the region with large inconsistencies: upper-right phase region. For comparison purposes, the results of Follower and our method for such a region are showed in Fig. 9, see figure caption. Additionally, we remark that in large phase maps the computational efficiency of the proposal overcomes the competitors. The computational time (in seconds) for the evaluated algorithms is showed in Table 2. As can be observed, PUMA, Follower, WFF_QGPU, WFR_QGPU and the proposed ARM are computationally efficient for, relatively, small images sizes. However, the computational efficiency of our competitors degrades as the images size is increased. The PUMA low performance for large images is explained by the fact that graph-cut algorithms are of

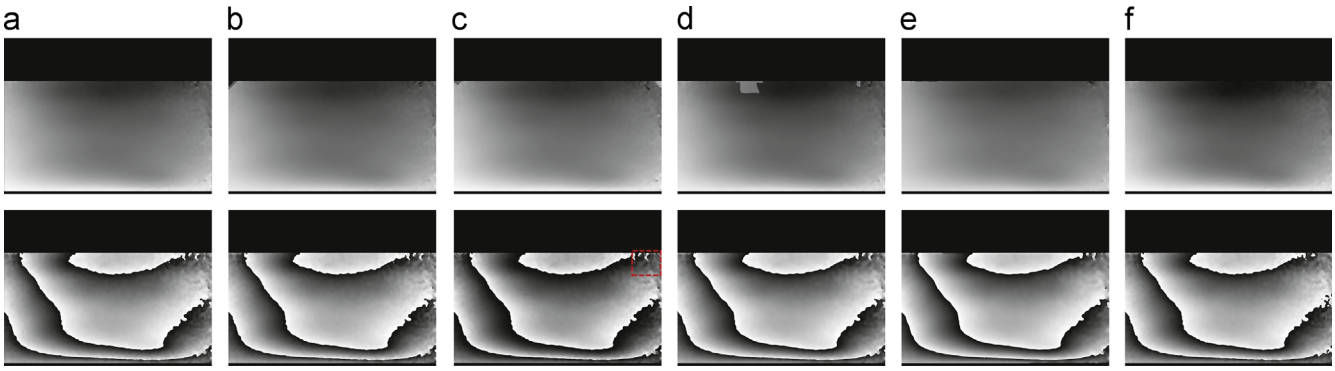

Fig. 8. Unwrapped phases (first row) and rewrapped phases (second row) computed with the evaluated methods. The test phase corresponds to the one in Fig. 1(c) with size of 1920×1440 pixels. Results for Goldstein algorithm are not showed because it does not converge after 5 computation days. (a) Quality guided, (b) PUMA, (c) Follower, (d) WFF_QGPU, (e) WFR_QGPU, (f) ARM. (For interpretation of the references to colour in this figure caption, the reader is referred to the web version of this paper.)

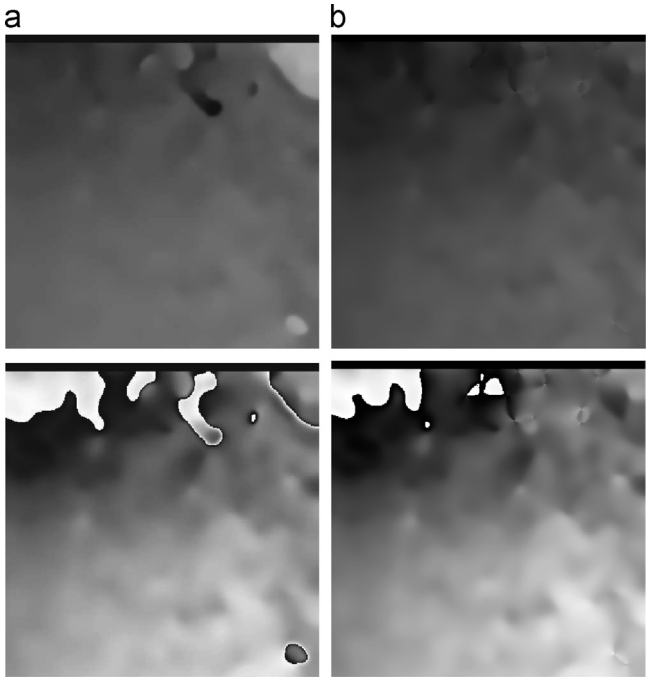

Fig. 9. Details of unwrapped phases (first row) and rewrapped phases (second row) computed with (a) Follower and (b) Proposal (ARM); region size of 494×494 pixels.

low-polynomial order [22], meanwhile multigrid algorithms could almost reach lineal order [10].

The experiments were conducted in a PC system: Windows 7 64 bits, Intel Core i7 3.4 MHz, 8GB RAM using a single thread.

5. Conclusion

We presented an iterative phase unwrapping algorithm that unwraps the residual (error) phase at each iteration. Our proposal is based on the minimisation of a robust energy for computing a real-valued residual phase map. This strategy is proper for being implemented in an efficient nested multigrid Gauss–Seidel algorithm and is computationally efficient for unwrapping large phase maps.

We demonstrate by experiments that our global (path independent) algorithm is effective for unwrapping noisy, discontinuous, inconsistent and large phases when is compared with path dependent strategies as: Quality Guided [13], Goldstein's [15], Follower [17] and the WF(F/R)_QGPU [19,20] algorithms. As it can be noted from Figs. 5 and 6, the proposed algorithm has a high performance in the presence of noise, which is the most common problem with experimental images, moreover, due to the path-independent principle used herein, large discontinuities can be identified and processed, resulting in no error propagation. Since this method is based on an iterative scheme, the computational time for small size images (512×512) is relatively long. However, we obtain a final phase unwrapping map almost error free.

Although multigrid phase unwrapping algorithms were early reported [5,23], our proposal uses a robust cost function adapted to an incremental formulation in combination with a nested multigrid strategy that can effectively unwrap discontinuous, inconsistent and large phase maps. We also noted that propagation based algorithms are a good option for relative small phase sizes and with good quality phase maps.

The presented implementation, based on a multigrid Gauss–Seidel scheme, uses a single threat. Our ARM method is proper for being implemented in parallel hardware which could improve its computational efficiency.

Acknowledgements

The authors thank to the anonymous reviewers for their comments that help to improve the quality of the paper. This work was supported in part by CONACYT, Mexico: research grants (131369 and 169178-F) and A. Gonzalez with a postdoctoral grant. The wrapped phase in Fig. 1(b) was kindly provided by Manuel H. De la Torre I.

References

[1] Hunt BR. Matrix formulation of the reconstruction of phase values from phase differences. J Opt Soc Am 1979;69(3):393–9.
[2] Ghiglia DC, Romero LA. Robust two-dimensional weighted and unweighted phase unwrapping that uses fast transforms and iterative methods. J Opt Soc Am A 1994;11(1):107–17.
[3] Marroquin JL, Rivera M. Quadratic regularization functionals for phase unwrapping. J Opt Soc Am A 1995;12(11):2393–400.
[4] Rivera M, Marroquin JL, Servin M, Rodriguez-Vera R. Fast algorithm for integrating inconsistent gradient fields. Appl Opt 1997;36(32):8381–90.
[5] Botello S, Marroquin JL, Rivera M. Multigrid algorithms for processing fringe-pattern images. Appl Opt 1998;37(32):7587–95.
[6] Rivera M, Marroquin JL. Half-quadratic cost functions for phase unwrapping. Opt Lett 2004;29(5):504–6.
[7] Bioucas-dias J, Member S, Valado G. Phase unwrapping via graph cuts. IEEE Trans Image Process 2007;16(3):698–709.
[8] Itoh K. Analysis of the phase unwrapping algorithm. Appl Opt 1982;21 (14):2470.
[9] Black M, Rangarajan A. On the unification of line processes, and robust statistics with applications in early vision. Int J Comput Vis 1996;19(1):57–92.

[10] Briggs WL, McCormick S, Henson V. A multigrid tutorial. 2nd ed., Philadelphia, Pa.: SIAM Publications; 2000.
[11] Geman S, McClure DE. Statistical methods for tomographic image reconstruction. Bull Intl Statist Inst 1987;LII(4):5–21.
[12] Charbonnier P, Blanc-Feraud L, Aubert G, Barluad M. Deterministic edge-preserving regularization in computed imaging. IEEE Trans Image Process 1997;6:298–311.
[13] Ghiglia DC, Pritt MD. Two-dimensional phase unwrapping: theory, algorithms, and software. New York: Wiley; 1998.
[14] MathWorks. 2D phase unwrapping algorithms. Website: ⟨http://www.mathworks.com/matlabcentral/fileexchange/22504-2d-phase-unwrapping-algorithms/content/GoldsteinUnwrap2D.m⟩, 2014.
[15] Goldstein RM, Zebken HA, Werner CL. Satellite radar interferometry: two-dimensional phase unwrapping. Radio Sci 1988;23(4):713–20.
[16] Bioucas-Dias J. Code. Website: ⟨http://www.lx.it.pt/~bioucas/code.htm⟩, 2014.
[17] Estrada JC, Vargas J, Flores-Moreno JM, Quiroga JA. Windowed phase unwrapping using a first-order dynamic system following iso-phase contours. Appl Opt 2012;51(31):7549–53.
[18] Estrada JC. Phase unwrapping software: ⟨http://goo.gl/RW2nQ⟩, 2012.
[19] Kemao Q. Two-dimensional windowed Fourier transform for fringe pattern analysis: principles, applications and implementations. Opt Lasers Eng 2007;45(2):304–17.
[20] Kemao Q, Gao W, Wang H. Windowed Fourier-filtered and quality-guided phase-unwrapping algorithm. Appl Opt 2008;47(29):5420–8.
[21] Kemao Q. Fourier transform windowed phase unwrapping software: ⟨http://www.mathworks.com/matlabcentral/fileexchange/authors/64472⟩, 2009.
[22] Kolmogorov V, Zabih R. What energy functions can be minimized via graph cuts. IEEE Trans Pattern Anal Mach Intell 2004;26:65–81.
[23] Pritt MD. Phase unwrapping by means of multigrid techniques for interferometric SAR. IEEE Trans Geosci Remote Sens 1996;34(3):728–38.