Phase Unwrapping by Accumulation of Residual Maps

Mariano Rivera∗, Francisco J. Hernandez-Lopez and Adonai Gonzalez∗
Centro de Investigacion en Matematicas AC; Guanajuato, Mexico 36000

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.

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

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][2][3][4][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

\begin{align}
\mathcal{L} & \text{ is the non–linear wrapping operator, with } n \text{ an integer such that } Wz & \in (−\pi, \pi]. \text{ The } W \text{ operator has the following properties:}
\end{align}

\begin{align}
f & = W^{-1} Wf, \\
Wf & = W Wf, \\
W^{-1} f & = W^{-1} W^{-1} f, \\
\sin f & = \sin(Wf); \\
g_r & = W\phi_r; \\
Wz & \overset{\text{def}}{=} z + 2n\pi,
\end{align}

one (wrapping) can be stated as follows: let $\phi$ be a phase map defined in a regular lattice $\mathcal{L}$ and let $g$ be its wrapped version such that

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{fig1}
\caption{Test wrapped phases. (a) Phase1 (synthetic), 512 × 512 pixels. (b) Phase2 (cornea), 1920 × 1440 pixels. (c) Phase3 (iron plate), 423 × 423 pixels.}
\end{figure}
where $W^{-1}$ denotes the unwrapping operator. We note that in general $f \neq W W^{-1} f$.

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

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

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 $\phi$, if

$$
|\phi_r - \phi_s| < \pi,
$$

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

$$
\phi_r - \phi_s = W(g_r - g_s). \quad (6)
$$

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

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

where $\lambda$ and $\mu$ are positive parameters of the algorithm, and we define

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

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 $\mu$ 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,

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

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 Figure 1(a). Figure 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 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 $\delta$ such that

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

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

$$
\delta_r - \delta_s = \rho_{rs}; \quad (11)
$$

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

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

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

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 $\phi$, if

$$
|\phi_r - \phi_s| < \pi,
$$

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

$$
\phi_r - \phi_s = W(g_r - g_s). \quad (6)
$$

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

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

where $\lambda$ and $\mu$ are positive parameters of the algorithm, and we define

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

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 $\mu$ 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,

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

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 Figure 1(a). Figure 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 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 $\delta$ such that

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

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

$$
\delta_r - \delta_s = \rho_{rs}; \quad (11)$$
\[
\tilde{\rho}_{rs} = \rho_{rs} - (f_r^{(0)} - f_s^{(0)}). \tag{12}
\]

with \(\rho_{rs}\) 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:

\[
[\delta, \omega] = \arg\min_{\delta, \omega} U(\delta, \omega; \tilde{\rho}). \tag{13}
\]

The solution to (13) can be computed by alternating minimisations with respect to \(\omega\) and \(\delta\). Thus, if \(\omega\) is fixed; the solution of the positive–definite diagonal–dominant linear system that results of \(\frac{\partial U}{\partial \omega} = 0\) can be computed with a Gauss–Seidel iterative scheme:

\[
\delta_r = \frac{\sum_{i \in N_r} \omega_i^2 (\delta_i + \tilde{\rho}_{rs}) + \lambda \delta_s}{\sum_{i \in N_r} \omega_i^2 [1 + \lambda]} \tag{14}
\]

Similarly, from \(\frac{\partial U}{\partial \delta} = 0\), we obtain the closed formula:

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

The process is resumed in Algorithm 1.

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

1: \textbf{function} \textsc{RobustUnwrap}(\(g, f, \lambda, \mu, T\))
2: \hspace{1cm} \triangleright An initial value \(f \neq 0\) can be passed
3: \hspace{1cm} Set \(\delta_r = 0, \forall r;\)
4: \hspace{1cm} Set \(\omega_{r+s} = 1, \forall (r, s);\)
5: \hspace{1cm} \triangleright An different initial value for \(\omega\) can be used
6: \hspace{1cm} Compute \(\hat{\rho}_{rs} = W(g_r - g_s) - (f_r - f_s), \forall (r, s);\)
7: \hspace{1cm} for \(t = 1, 2, \ldots, T\) do
8: \hspace{2cm} Update \(\delta_r\), keeping fixed \(\omega_r\), with (14);
9: \hspace{2cm} Update \(\omega_{rs}\), keeping fixed \(\delta_r\), with (15);
10: \hspace{1cm} end for
11: return \(f + \delta;\)
12: \textbf{end function}

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 \(f^{n} + \delta\) can, again, be corrected with the proposed scheme. This iterative upgrade of the partial solution is expressed by

\[
f_r^{(n+1)} = f_r^{(n)} + \delta_r^{(n)}. \tag{16}
\]

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

\[
[\delta^{(n)}, \omega^{(n)}] = \arg\min_{\delta, \omega} U(\delta, \omega; \tilde{\rho}^{(n)}). \tag{17}
\]

The convergence of this iterative scheme is guaranteed because

\[
U(\delta^{(n)}, \omega^{(n)}; \tilde{\rho}^{(n)}) \leq U(\mathbf{0}, 1; \tilde{\rho}^{(n)}). \tag{18}
\]

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.

![Multigrid strategy illustration](image)

Figure 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.

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 smooths 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 \(\lambda\) 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 Figure 3a 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
iteration (node in the multigrid scheme). The residual computing the residual wrapped phase at the end of each Algorithm 3. The multigrid strategy depicted in Figure 3c and detailed in error reduction. Our proposal implements a full–nested additional time, as benefit we improve the low frequency stages (top levels in the scale pyramid) have low resolution. Thus, a nested multigrid approach implements e

The simple multigrid scheme depicted in Figure 3a 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 sub-sampled (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 error. Thus, a nested multigrid approach implements efficient solvers [10]; the nested multigrid is depicted in Figure 3b. 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 Figure 3c 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

$$\text{residual}(g, f) \overset{\text{def}}{=} W \left( W^{-1} g - W^{-1} f \right),$$

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 $W^{-1}$:

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

with the identities:

$$\sin(g - f) = \sin g \cos f - \cos g \sin f,$$
$$\cos(g - f) = \cos g \cos f + \sin g \sin f.$$
The columns correspond to different λ values. First row depicts the absolute value of the level sets of the potentials and second row depicts the influence function. The x-axis corresponds to δ_ρ − δ_s values and y-axis to ˜ρ_s values.

Figure 4: Effect of the λ parameter of the proposed potential (25). The columns correspond to different λ values. First row shows the respective influence functions and second row shows the respective potentials.

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 need 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, θ = θ1, θ2 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:

\[ \theta_i = \text{mean}(W(g_r - g_{r-e_i}))_{(r,e) \in \mathcal{L}} \]  

for \( i = 1, 2 \); where the operator mean\( \mathcal{A} \) return the average of a set \( \mathcal{A} \) 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 \( M \subset \mathcal{L} \) (Mask), different from the whole image domain, can be achieved by redefining the neighbourhood (4) as

\[ N_r = \{ s \in M : ||r - s||_2 = 1 \} \]

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 \( \omega_{rs} \) [i.e., formula (15)] into the potential \( U(\delta, \omega, \rho) \). Then, we have that (13) is equivalent to

\[ \underset{\delta}{\text{arg min}} \ U(\delta; \rho) = \frac{1}{2} \sum_{r \in \mathcal{L}, s \in N_r} P(\delta_r - \delta_s, \tilde{\rho}_{rs}) \]

(24)

where

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

(25)

which can be seen as an extended version of the robust non-convex Geman-McClure potential [11]. The potential \( P \) can be analyzed 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 (\( \lambda \)). Figure 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 \( \lambda \) values, the white level indicates largest function values and black levels the lowest ones. The effect on the estimator of a phase differences \( \delta_r - \delta_s \) can be observed in the so called influence function, \( \psi \); second row in Figure 4.

This influence function is defined, in terms of the residual phase differences \( \delta_r - \delta_s \), as

\[ \psi(\delta_r - \delta_s, \tilde{\rho}_{rs}) = \frac{\partial P(\delta_r - \delta_s, \tilde{\rho}_{rs})}{\partial (\delta_r - \delta_s)} \]

(26)

Second row in Figure 4 shows the respective influence functions for the different \( \lambda \) 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 \( \lambda \) values, the low-influence region is composed by sites where \( \delta_r - \delta_s - \tilde{\rho}_{rs} \approx 0 \). Thus, as \( \lambda \) 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 \( \lambda \) 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 Figure 1. Figure 5 shows the unwrapped phase corresponding to the synthetic phase in Figure 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 \( \mu \) 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)

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

that can be understood as a quality map. The numerical experiment was performed in two conditions: noise-free (Figure 5) and noisy data (Figure 6). We compare the performance of the proposal (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 compute the unwrapped phase. The reconstruction error is shown in the second rows Table 1 and the computational times (for the noise–free case) are shown in the second rows Table 2, image size of $512 \times 512$ 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. Figure 1(b), shows a wrapped phase corresponding to a Placido’s Fringe Pattern (PFP) with large inconsistencies. A PFP are equally spaced monochromatic concentric rings placed in front of an 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-
Figure 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) with an image size of 423 × 423 pixels.

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

<table>
<thead>
<tr>
<th>Size of Image</th>
<th>Quality Guided</th>
<th>Goldstein</th>
<th>PUMA</th>
<th>Follower</th>
<th>WFF_QGPU</th>
<th>WFR_QGPU</th>
<th>ARM</th>
</tr>
</thead>
<tbody>
<tr>
<td>512 × 512</td>
<td>0.056</td>
<td>2.319</td>
<td>1.042</td>
<td>3.837</td>
<td>1.006</td>
<td>0.132</td>
<td>0.008</td>
</tr>
<tr>
<td>Noise-Free</td>
<td>0.583</td>
<td>2.092</td>
<td>1.009</td>
<td>3.835</td>
<td>1.070</td>
<td>0.710</td>
<td>0.011</td>
</tr>
</tbody>
</table>

Figure 1(b) shows phase inconsistencies produced by effect of eyelashes, noise and the hole at the PFP center. The image dimension are 423 × 423 pixels. On the other hand, Figure 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 are 1920 × 1440 pixels.

Figures 7 and 8 show the unwrapped phases with the compared algorithms. The phases in Figures 7 and 8 are smooth. As we do not expect phase discontinuities, the discontinuity sensitivity parameter $\mu$ 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 Figure 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. Figure 8 shows the unwrapped phases with the same algorithms than in Figure 7 (except 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 Figure 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 are 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 low–polynomial order [22], meanwhile multigrid algorithms could almost reach lineal order [10].

The experiments were conducted in a PC system: Windows 7 64bits, 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 compared with path dependent strategies as: Quality Guided [13], Goldstein’s [15], Follower [17] and the WFF(R)QGPU[19],[20] algorithms. As it can be note from Figure 5 and Figure 6, the proposed algorithm has a high performance in presence of noise, which is the most common problem with experimental images, more over, due to path-independent principle used here in, 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 x 512) is relatively large, 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 unwraps 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

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). (*) Goldstein’s algorithm does not converged after 5 days.

<table>
<thead>
<tr>
<th>Dimension in pixels</th>
<th>Quality Guided</th>
<th>Goldstein</th>
<th>PUMA</th>
<th>Follower</th>
<th>WFF QGPU</th>
<th>WFR QGPU</th>
<th>ARM</th>
</tr>
</thead>
<tbody>
<tr>
<td>512 x 512</td>
<td>838.773</td>
<td>52.696</td>
<td>11.698</td>
<td>1.006</td>
<td>27.525</td>
<td>19.780</td>
<td>56.007</td>
</tr>
<tr>
<td>1920 x 1440</td>
<td>7.1 x 10^1</td>
<td>(*)</td>
<td>192.373</td>
<td>33.001</td>
<td>254.394</td>
<td>224.897</td>
<td>46.443</td>
</tr>
</tbody>
</table>

Figure 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 x 1440 pixels. Results for Goldstein algorithm are not showed because it does not converge after 5 computation days.

Figure 9: Details of unwrapped phases (first row) and rewrapped phases (second row) computed with (a) Follower and (b) Proposal (ARM); region size of 494 x 494 pixels.
method is proper for being implemented in parallel hardware which could improve its computational efficiency.

Acknowledges

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 Figure 1(b) was kindly provided by Manuel H. De la Torre I.

Note for the reviewers: Once the paper would be accepted, we (the authors) plan to publish the final version of the code and the link will be included in the published version.