

Robust phase unwrapping algorithm based on enhanced denoising and fringe quality improvement routines

PEDRO ENRIQUE ALCARAZ, REMINGTON SPENCER KETCHUM, AND
PIERRE-ALEXANDRE BLANCHE* 

College of Optical Sciences, University of Arizona, 1630 E University Blvd., Tucson, AZ 85721, USA

**Corresponding author: pblanche@optics.arizona.edu*

Abstract: Phase unwrapping algorithms have widely been studied and implemented with efforts aimed at unwrapping wrapped phase signals. However, the presence of noise and unreliable fringe quality poses a major obstacle for the retrieval of reliable phase signals. While many techniques have been implemented to deal with the aforementioned issues, most algorithms are application dependent or difficult to implement. Here we present a simple yet effective global phase unwrapping algorithm, that does not resort to Least-Squares Minimization, making use of Fast-Fourier Transform (FFT) based spectral differentiation, Signal Dependent Rank Ordered Mean (SD-ROM) filtering, and Fuzzy Logic Edge Detection (FLED). The proposed algorithm was tested using simulated, noisy, wrapped phaseograms and has shown to improve image and fringe quality, as well as overall retrieved phase reliability.

© 2021 Optical Society of America under the terms of the [OSA Open Access Publishing Agreement](#)

1. Introduction

Phase unwrapping is a technique that has widely been explored in literature for the post-processing of recovered wrapped interferograms, phaseograms, and signals. This technique is widely used in a variety of fields including medical imaging with - magnetic resonance imaging, [1] and interference-based, label-free, digital holographic microscopy, [2] topological surveying - with interferometry synthetic aperture radar (InSAR) images, [3] image reconstruction with fringe projection profilometry, [4,5] and more recently, an interferometry based radar emulation system, [6] where the imaging modalities ultimately produce wrapped signals. The wrapped signals of interest typically lay in the interval of $[-\pi, \pi]$. This property has been exploited by region, path following, or global phase unwrapping algorithms with efforts aimed at recovering the unwrapped signals of interest. However, unwrapping phaseogram that are corrupted by noise or phase-based residues, due to faulty image acquisition techniques, sensors, or noisy media, has proven to be a major obstacle for reliable phase retrieval and been the subject of many recent publications [7–11]. Presently, a search for robust, efficient, and application-independent phase unwrapping algorithms that handle a high degree of noise and defective fringe patterns is necessary for reliable phase retrieval. Discussed below is the implementation of a global phase unwrapping algorithm whose efforts are aimed at minimizing the difference between the true phase and unwrapped phase gradients using spectral differentiation techniques. This article presents modifications to the techniques introduced by Schofield et al., 2003, and subsequently Shi et al., 2010, that improve phase unwrapping of both simulated and experimental wrapped phases. Furthermore, the proposed phase unwrapping algorithm utilizes Signal Dependent Rank Ordered Mean (SD-ROM) filtering and Fuzzy Logic Edge Detection (FLED) routines that can be easily implemented and are robust enough to handle images with very low SNRs and high noise variation.

2. Methods

2.1. Problem formulation

The aim of phase unwrapping algorithms is to efficiently and accurately unwrap wrapped phaseogram. Consider the true phase of an image $\phi(x, y)$, and its wrapped phase $\phi_w(x, y)$, both dependent on the spatial domain, where pixels $(x, y) \in \mathbb{N}$. The relationship between the true phase and the wrapped phase of an image can be described via [Eq. (1)]:

$$\phi(x, y) = \phi_w(x, y) + 2\pi k(x, y); \quad k \in \mathbb{Z}; \tag{1}$$

where it is assumed that $\phi_w(x, y)$ is wrapped in the following manner $\phi_w(x, y) = \hat{W}[\phi(x, y)]$, \hat{W} being the wrapping operator: $\hat{W} = \text{mod}\{[\pi + \phi(x, y)], 2\pi\} - \pi$.

The proposed phase unwrapping algorithm relies on the assumption that the desired unwrapped and the wrapped phaseograms have the same, smooth, local phase differences; i.e. the phase gradients in the intervals $[-\pi, \pi]$ are equal. Therefore, the goal of phase unwrapping is to determine the integer values of $k(x, y)$. This can be accomplished by evaluating the change in the local phase gradients of both the wrapped and unwrapped phaseograms using the forward ∇_{\perp}^2 and inverse ∇_{\perp}^{-2} two-dimensional Laplacian operators:

$$k(x, y) = \frac{1}{2\pi} \nabla_{\perp}^{-2} [\nabla_{\perp}^2 \phi(x, y) - \nabla_{\perp}^2 \phi_w(x, y)] \tag{2}$$

2.2. Laplacian solution

The use of spectral differentiation methods, such as Fast-Fourier Transform (FFT) differentiation, can reduce the computational complexity associated with solving [Eq. (2)]. Recall that $f(x) = \int_{-\infty}^{\infty} F(\xi)e^{2\pi i\xi x} dx$ and in turn $f'(x) = \int_{-\infty}^{\infty} 2\pi i\xi F(\xi)e^{2\pi i\xi x} dx$, via the shifting theorem. Using this identity one can rewrite the Laplacian operators in the following manner:

$$\nabla_{\perp}^2 f(x, y) = -\frac{4\pi^2}{N \times M} \mathcal{F}^{-1} \{ (p^2 + q^2) \mathcal{F}[f(x, y)] \} \tag{3}$$

$$\nabla_{\perp}^{-2} g(x, y) = -\frac{N \times M}{4\pi^2} \mathcal{F}^{-1} \left\{ \frac{\mathcal{F}[g(x, y)]}{(p^2 + q^2)} \right\} \tag{4}$$

where $(x, y) \in N$ and $(p, q) \in N$ are the real and the Fourier space pixel coordinates, and $N \times M$ is the interferogram's area [12]. Note that to obtain $k(x, y)$ the unwrapped phase must be known a priori, see [Eq. (2)]. However, by defining the complex quantity $e^{i\phi_w(x, y)}$ and considering that $e^{i\phi_w(x, y)} = e^{i\phi(x, y) + 2\pi k(x, y)} = e^{i\phi(x, y)}$, one can define the following:

$$P(x, y) = e^{i\phi_w(x, y)} \tag{5}$$

$$\nabla_{\perp}^{-2} \phi(x, y) = \text{Im} \left\{ \frac{1}{P(x, y)} \nabla_{\perp}^2 P(x, y) \right\} \tag{6}$$

Combining [Eqs. (3)–(6)], one arrives at the following solution to for the phase unwrapping problem:

$$\phi(x, y) = \nabla_{\perp}^{-2} \left[\text{Im} \left\{ \frac{1}{P(x, y)} \nabla_{\perp}^2 P(x, y) \right\} - \nabla_{\perp}^2 \phi_w(x, y) \right] + \phi_w(x, y) \tag{7}$$

$$\text{Where, } k(x, y) = \frac{1}{2\pi} \nabla_{\perp}^{-2} \left[\text{Im} \left\{ \frac{1}{P(x, y)} \nabla_{\perp}^2 P(x, y) \right\} - \nabla_{\perp}^2 \phi_w(x, y) \right] \tag{8}$$

2.3. Phase unwrapping - prior art

Schofield et al., 2003, propose combining [Eqs. (2)–(4) and (6)] to generate an unwrapped phase estimate:

$$\begin{aligned} \phi_{est}(x, y) = \mathcal{F}^{-1} & \left(\frac{\mathcal{F}\{ \cos\phi_w(x, y) \mathcal{F}^{-1}[(p^2 + q^2)\mathcal{F}(\sin\phi_w(x, y))] \}}{(p^2 + q^2)} \right) \\ & - \mathcal{F}^{-1} \left(\frac{\mathcal{F}\{ \sin\phi_w(x, y) \mathcal{F}^{-1}[(p^2 + q^2)\mathcal{F}(\cos\phi_w(x, y))] \}}{(p^2 + q^2)} \right) \end{aligned} \quad (9)$$

Where, (p, q) are the Fourier space pixel coordinates and [Eq. (6)] is rewritten as:

$$\nabla_{\perp}^2 \phi(x, y) = \cos(\phi_w(x, y)) \nabla_{\perp}^2 \sin(\phi_w(x, y)) - \sin(\phi_w(x, y)) \nabla_{\perp}^2 \cos(\phi_w(x, y)) \quad (10)$$

Schofield et al., 2003 suggests that phase unwrapping is an iterative process; where

$$\phi'_{est}(x, y) = \phi_w(x, y) + 2\pi \text{round} \left(\frac{\phi_{est}(x, y) - \phi_w(x, y)}{2\pi} \right) \quad (11)$$

Note that when using FFT-based spectral differentiation techniques such as [Eqs. (3) and (6)–(8)], it is important that the wrapped phaseogram be periodic in-order to be differentiable within Fourier space. Several boundary conditions have been studied and implemented in literature. Schofield et al., 2003, suggest adding symmetric extensions of the $N \times M$ wrapped phaseogram to meet the Neumann boundary condition necessary for phase retrieval, generating a $2N \times 2M$ matrix, seen in Fig. 1; where the unwrapped retrieved phase solution should be derived from the third quadrant [13].

Shi et al., 2010, propose a technique similar to that of Schofield et al., 2003, combining [Eqs. (2)-(4) and (10)] to generate an unwrapped phase estimate, however, they use the Discrete-Cosine Transform (DCT) to compute the forward and reverse Laplacian operators. Their estimated phase is presented as follows:

$$\begin{aligned} \phi_{est}(x, y) = \text{DCT}^{-1} & \left(\frac{\text{DCT}\{ \cos\phi_w(x, y) \text{DCT}^{-1}[(p^2 + q^2)\text{DCT}(\sin\phi_w(x, y))] \}}{(p^2 + q^2)} \right) \\ & - \text{DCT}^{-1} \left(\frac{\text{DCT}\{ \sin\phi_w(x, y) \text{DCT}^{-1}[(p^2 + q^2)\text{DCT}(\cos\phi_w(x, y))] \}}{(p^2 + q^2)} \right) \end{aligned} \quad (12)$$

Shi et al., 2010, also suggest that phase unwrapping is an iterative process; where

$$\phi'_{est}(x, y) = \phi_w(x, y) + 2\pi \text{round} \left(\frac{\phi_{est}(x, y) - \phi_w(x, y)}{2\pi} \right) \quad (13)$$

However, they do not add symmetric extensions of the $N \times M$ wrapped phaseogram to solve combining [Eq. (3), Eqs. (4), and (6)], claiming that the Neumann boundary conditions are satisfied by virtue of using the DCT technique [14].

2.4. Phase unwrapping modifications

As mentioned previously, the current spectral differentiation-based phase unwrapping techniques are assumed to provide estimates of the true unwrapped phase $\phi(x, y)$. Both [Eqs. (9) and (12)] run through an iterative rounding process described in [Eqs. (11) and (13)] in an effort to converge upon $\phi(x, y)$. However, the implementation of these iterative algorithms tend to fail after the first iteration [15]. Here we propose a novel approach to accurately perform Laplacian phase unwrapping, keeping in line with most of the work presented by Schofield et al., 2003. The first

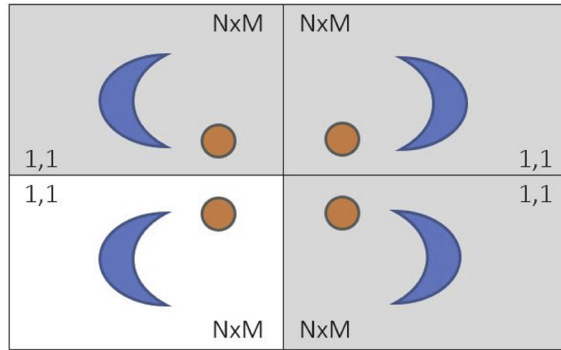


Fig. 1. Illustration of $2N \times 2M$ matrix configuration used by Schofield et al., 2003, to meet the Neuman boundary condition. The original wrapped and unwrapped phaseograms, denoted by the white background, are in the third quadrant.

amendment we propose is aimed at the (p, q) matrix indices in the Fourier domain; instead of using pixel coordinates we propose using the corresponding Fourier space values:

$$x_m = \left\{ \frac{-N}{2}, \frac{N}{2}, N \right\}; y_m = \left\{ \frac{-M}{2}, \frac{M}{2}, M \right\} \tag{14}$$

$$\xi_m = \left\{ \frac{-1}{2\Delta x_n}, \frac{1}{2\Delta x_n}, N \right\}; \eta_m = \left\{ \frac{-1}{2\Delta y_m}, \frac{1}{2\Delta y_m}, M \right\} \tag{15}$$

The modified Fourier space coordinates are denoted as (ξ_n, η_m) , bounded on the interval $(\frac{-1}{2\Delta x_n}, \frac{1}{2\Delta x_n})$, and $(\frac{-1}{2\Delta y_m}, \frac{1}{2\Delta y_m})$, with $N \times M$ number of indices, hence the notation (ξ_n, η_m) . Thus, generating the following matrix (ξ_n^2, η_m^2) to replace $(p^2 + q^2)$ in [Eqs. (3), Eq. (4)].

$$\nabla_{\perp}^2 f(x, y) = -\frac{4\pi^2}{N \times M} \mathcal{F}^{-1} \{ (\xi_n^2, \eta_m^2) \mathcal{F} [f(x, y)] \} \tag{16}$$

$$\nabla_{\perp}^{-2} g(x, y) = -\frac{N \times M}{4\pi^2} \mathcal{F}^{-1} \left\{ \left(\frac{\mathcal{F} [g(x, y)]}{(\xi_n^2, \eta_m^2)} \right) \right\} \tag{17}$$

$$k(x, y) = \frac{1}{2\pi} \nabla_{\perp}^{-2} \left[\text{Im} \left\{ \frac{1}{P(x, y)} \nabla_{\perp}^2 P(x, y) \right\} - \nabla_{\perp}^2 \phi_w(x, y) \right] \tag{18}$$

These modifications are more consistent with the spectral differentiation techniques, where $f'(x) = \int_{-\infty}^{\infty} 2\pi i \xi F(\xi) e^{2\pi i \xi x} dx$.

Using the spectral differentiation techniques presented in [Eqs. (16) and (17)] one can approximate the $\tilde{k}(x, y)$ wrapping integers via [Eq. (18)]. These values are considered equal, if not close to, the expected $k(x, y)$ values required to solve [Eq. (1)]. However, the presence of discontinuities that cannot be accounted for via the applied Neumann boundary conditions tends to generate minor scaling effects in $\tilde{k}(x, y)$ motivating the use of the rounding operators found in [Eqs. (11) and (13)]. However, use of either [Eqs. (11) or (13)] can produce unwanted rounding errors in the recovered phaseogram when $\tilde{k}(x, y)$ is sufficiently close to $n \pm 0.5, n \in \mathbb{Z}$. The scaling issue can be properly accounted for by exploiting the relationship between the wrapped and unwrapped phaseogram denoted in [Eq. (1)]. Being that $\tilde{k}(x, y)$ must be an integer value, note that $\sin(\pi k(x, y)) = 0$ for all $k(x, y)$. Therefore, if there is residual scaling after recovery of the $\tilde{k}(x, y)$ term, a scaling factor can be applied to correct the $\tilde{k}(x, y)$ term and retrieve the expected $k(x, y)$

value by minimizing the number of pixels whose $\sin(\pi k(x, y))$ values are non-zero. The global scaling factor, $s_{i,min}$, can be determined by finding the local minimum of the following function:

$$\sum_{x=1}^N \sum_{y=1}^M |\sin(\pi s_i \tilde{k}(x, y))|; \quad i \in \mathbb{R} \tag{19}$$

$$k(x, y) = s_{i,min} \cdot \tilde{k}(x, y) \tag{20}$$

Note the periodic nature of the $\sin(\pi s k(x, y))$, wherein scaling $s \approx 2$ yields $\approx \sin(\pi s k(x, y))$, therefore the local scaling minima should be found in the region $s_i \approx [-2, 2]$. Most scaling values, in both the simulated and experimental data, were found to be within $s_{i,min} = [0.8, 1.3]$.

Thus, the final unwrapped phase can be obtained via the following:

$$\phi(x, y) = \phi_w + s_{i,min} \cdot \frac{1}{2\pi} \nabla_{\perp}^{-2} [Im\{\frac{1}{P(x, y)} \nabla_{\perp}^2 P(x, y)\} - \nabla_{\perp}^2 \phi_w(x, y)] \tag{21}$$

making use of [Eqs. (1), (6), (16–18), and (20)].

2.5. Phase unwrapping image processing

2.5.1. Noise reduction

Images and signals are often corrupted by noise due to faulty image acquisition techniques/sensors. Although noisy phaseograms can be handled by the spectral differentiation techniques proposed above, the degree of noise often becomes a limiting factor in calculating the Laplacian of a wrapped phase, especially when the wrapped phase contains clusters of impulse noise near or at the $[-\pi, \pi]$ boundary. Widely available filters such as the 2-D Wiener filter provide a satisfactory approach for the removal of noise from images; [16] however, such filters modify pixel values whether or not they are corrupt. For most cases improvement on the local phase gradient can be observed, hence the wide implementation of the 2D Wiener filter in phase unwrapping algorithms, however when clusters of impulses like noise are present in the wrapped phaseogram the filters can ultimately distort or blur surrounding features, leading to unreliable wrapped phases and phase gradients. To circumvent this problem, it is important to first remove corrupted impulse like pixels and subsequently apply a global filter. The SD-ROM filter provides such capabilities [17].

2.5.2. SD-ROM Filter

Briefly, the SD-ROM filter is an efficient non-linear filter that suppresses impulse corrupted images whilst preserving uncorrupted pixels. Consider $x(n)$, the luminescence value of a noisy image $[0, 255]$; where (n) refers to the pixel location (x, y) in the wrapped noisy phaseogram. Within $x(n)$, is a 3×3 window of pixels. The pixels surrounding $x(n)$ are denoted as $w(n)$, where $w(n)$ consists of pixels $w_i(n) = [w_1(n), w_2(n), w_3(n), \dots, w_8(n)]$; where $w_i(n), i = [1, 8], i \in \mathbb{Z}$, corresponds to a left to right, top to bottom mapping of the pixels surrounding $x(n)$, see Fig. 2. The pixels found in $w(n)$ can be rank-ordered $r_i(n) = [r_1(n), r_2(n), r_3(n), \dots, r_8(n)]$; in ascending order such that $r_1(n) \leq r_2(n) \leq r_3(n) \leq \dots \leq r_8(n)$. Thereafter, the rank-ordered mean of the window $w(n)$ can be found, where $m(n) = \frac{r_4(n)+r_5(n)}{2}$. Finally, the rank-ordered differences within the window $w(n)$ can be found; where, $d(n) = [d_1(n), d_2(n), d_3(n), d_4(n)]$. The values pertaining to $d_j(n), j = [1, 4], j \in \mathbb{Z}$; are designated with the following rule:

$$d_j(n) = \begin{cases} r_j(n) - x(n) & \text{for } x(n) \leq m(n) \\ x(n) - r_{9-j}(n) & \text{for } x(n) > m(n) \end{cases}$$

The SD-ROM filter detects impulse noise using four thresholding values, namely $T_j, j = [1, 4], j \in \mathbb{Z}$. If $d_j(n) > T_j$ then the pixel $x(n)$ is considered corrupt. Note that $T_1 < T_2 < T_3 < T_4$. The

algorithm performs well for values of $T_1 = 8, T_2 = 20, T_3 = 40, T_4 = 50$, however, the thresholding values can be modified on a case by case basis for a-priori knowledge of the impulse magnitude, suggested thresholds limits are as follows: $T_1 \leq 15, 15 \leq T_2 \leq 25, 30 \leq T_3 \leq 50, 40 \leq T_4 \leq 60$. If the pixel is corrupt the pixel value $x(n)$ is changed to $y(n) = m(n)$. This algorithm can also be implemented recursively, where the window $w(n)$ slides through the images (left to right, top to bottom) replacing corrupt $x(n)$ for $y(n)$. In the recursive use of the SD-ROM filter, previously modified pixels $y(n)$ contribute to the rank, mean, and rank ordered difference values see Fig. 2. In this article, we implemented the non-recursive SD-ROM filter.

Note that if the wrapped phaseogram contains non-impulse noise or the impulse noise has been removed, the 2-D Wiener filter can be utilized to smooth the phase gradients present in the signal. It is recommended to first use the SD-ROM filter to remove impulse clusters and then use a 2D Wiener filter to reduce non-impulse like noise in the wrapped phaseogram, see [Eqs. (22–23)].

$$\phi_w^y(x, y) = SDROM(\phi_w(x, y)) \tag{22}$$

$$\phi_w^{NF}(x, y) = 2D\text{-Wiener}(\phi_w^y(x, y)); \text{ Filter Size } [11, 11] \tag{23}$$

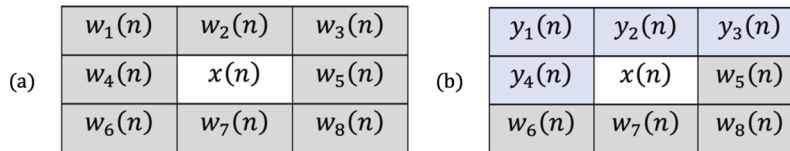


Fig. 2. (a) A schematic of the non-recursive SD-ROM analysis window $w(n)$. (b) A schematic of the recursive SD-ROM sliding analysis window $w(n)$.

2.5.3. Quality map

Global phase unwrapping algorithms, specifically those using spectral methods, are robust enough to deal with phaseograms containing a significant amount of noise. However, quality maps are often implemented to remove unwanted pixels from highly distorted wrapped phaseogram in an effort to improve the accuracy and efficiency of phase unwrapping algorithms. Furthermore, quality maps can significantly improve phase unwrapping when the noise itself has been wrapped [18,19]. The measure of pixel "quality" $R(x, y)$ is determined by the variance of the pixel's phase derivative.

$$R(x, y) = \frac{\sqrt{\sum_{i=x-n/2}^{x+n/2} \sum_{j=y-m/2}^{y+m/2} (\Delta_{i,j}^x - \Delta_{i,j}^{-x})^2}}{n \times m} + \frac{\sqrt{\sum_{i=x-n/2}^{x+n/2} \sum_{j=y-m/2}^{y+m/2} (\Delta_{i,j}^y - \Delta_{i,j}^{-y})^2}}{n \times m} \tag{24}$$

Where, n and m correspond to the window size with center pixels (x, y) ; the terms $\Delta_{i,j}^x$ and $\Delta_{i,j}^y$ are the partial derivatives of the phase. The terms $\Delta_{i,j}^{-x}$ and $\Delta_{i,j}^{-y}$ are the averages of the partial derivatives in the $n \times m$ window. One can retrieve a quality based phase binary mask by applying a moderate threshold, where pixels with quality $> T$ are kept.

$$Q(x, y) = \begin{cases} 0 & \text{for } R(x, y) < T \\ 1 & \text{for } R(x, y) \geq T \end{cases} \tag{25}$$

Here T , is to be considered a threshold value between $R(x, y) = 0$ and $\max(R(x, y))$. It is noted that for this study the thresholding value of 0.9 was used. Implementation of the quality based

phase mask is shown below:

$$\phi_w^Q(x, y) = \phi_w(x, y) \cdot Q(x, y) \quad (26)$$

Note, it is suggested that the PDV filtered wrapped phaseogram $Q(x, y) \cdot P(x, y)$ be used before the SD-ROM and 2-D Wiener filter if a high degree of noise is present or if a wrapped signal is present within a wrapped noisy background, see Fig. 8(a).

2.5.4. Fuzzy Logic Edge detection

For special cases where the fringes of a wrapped phaseogram are highly corrupted and noise processing cannot produce reliable fringes, unwrapping the phaseogram and obtaining a reliable $k(x, y)$ becomes difficult. However, upon unwrapping, a reliable boundary between regions with differing $k(x, y)$ is possible. This condition makes it possible to reconstruct the boundary of the noisy wrapped phaseogram by estimating the location of the corrupted $[-\pi, \pi]$ boundaries. Here it is suggested that the $2\pi\Delta_x k(x, y)$ and $2\pi\Delta_y k(x, y)$ at the boundaries between successive $k(x, y)$ should differ greatly from the gradients within a single $k(x, y)$ region; where regions inside the $k(x, y) \approx 0$ and gradients at successive $k(x, y)$ boundaries should ideally be 2π (or at minimum >0 when $\tilde{k}(x, y) \neq k(x, y)$).

Using this condition, and taking advantage of MATLAB's Fuzzy Logic App, one can estimate the location of the wrapped phase $[-\pi, \pi]$ discontinuities (FLED). The following parameters were utilized to estimate the wrapped phase discontinuity locations. For pixels where $2\pi\Delta_x k(x, y)$ or $2\pi\Delta_y k(x, y)$ are zero, the Fuzzy Logic Routine associates a value of 1 with the pixel, indicative of non-edge-like surfaces. For pixels where $2\pi\Delta_x k(x, y)$ or $2\pi\Delta_y k(x, y)$ are \gg zero, the Fuzzy Logic Routine associates a value of 0, indicative of edge-like-surfaces. For pixels who's $2\pi\Delta_x k(x, y)$ or $2\pi\Delta_y k(x, y)$ are not sufficiently close to either 2π or 0, the Fuzzy Logic routine then determines the probability of a pixel lying with an edge by utilizing a weighted value function. Parameters S_x and S_y are used to determine the number of standard deviations away from $2\pi\Delta_x k(x, y)$ or $2\pi\Delta_y k(x, y)$ that should be considered non-edge-like boundaries, triangle functions then determine the weighted values of pixels that lie within S_x and S_y and their likelihood of convergence towards an edge-like boundary. The parameters utilized in this article are shown in Fig. 3.

Once the weighted value function $I(x, y)$ is recovered, a binary mask can be created in accordance with the fuzzy logic parameters; where:

$$B(x, y) = \begin{cases} 0 & \text{for } I(x, y) < 0.5 \\ 1 & \text{for } I(x, y) \geq 0.5 \end{cases} \quad (27)$$

Pixel value reassignment along the mask can also be computed; where,

$$F(x, y) = \begin{cases} -\pi & \text{for } I(x, y) \geq 0.5; \quad \phi_w(x, y) < 0 \\ 0 & \text{for } I(x, y) < 0.5 \\ \pi & \text{for } I(x, y) \geq 0.5; \quad \phi_w(x, y) > 0 \end{cases} \quad (28)$$

Upon finding the binary mask and the boundary condition values, the wrapped phase noise-free wrapped phase is updated:

$$\phi_w'(x, y) = \phi_w^{NF}(x, y) \cdot B(x, y) + F(x, y) \quad (29)$$

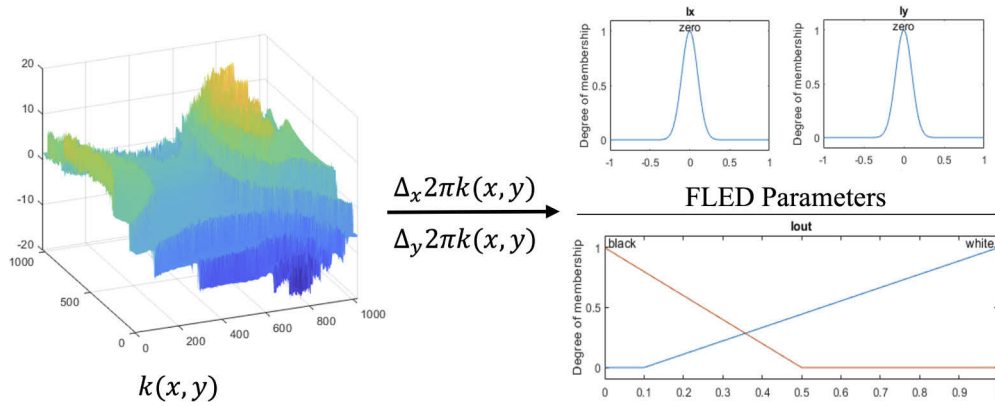


Fig. 3. Generalized Fuzzy Logic Scheme: $k(x, y)$ serves as the input for the Fuzzy Logic Edge detector. $2\pi\Delta_x k(x, y)$ and $2\pi\Delta_y k(x, y)$ are the relevant values associated with $k(x, y)$ that inform the classification of edge-like pixels and non-edge-like pixels; they are subsequently classified and weighted to produce $I(x, y)$.

Algorithm I: Noise-Free Wrapped Phase

Laplacian	$\mathfrak{K}(x, y) = \frac{1}{2\pi} \nabla_{\perp}^2 \left[\text{Im} \left\{ \left(\frac{1}{P(x, y)} \nabla_{\perp}^2 P(x, y) \right) - \nabla_{\perp}^2 \phi_w(x, y) \right\} \right]$	(18)
Scaling $\mathfrak{K}(x, y)$	$k(x, y) = s_{i, \min} * \mathfrak{K}(x, y)$	(19-20)
Unwrapped Phase	$\phi(x, y) = \phi_w(x, y) + s * \frac{1}{2\pi} \nabla_{\perp}^2 \left[\text{Im} \left\{ \left(\frac{1}{P(x, y)} \nabla_{\perp}^2 P(x, y) \right) - \nabla_{\perp}^2 \phi_w(x, y) \right\} \right]$	(21)

Algorithm II: Noisy Wrapped Phase

Denoising	$\phi_w^Y(x, y) = SD - ROM(\phi_w(x, y));$ $\phi_w^{WF}(x, y) = 2D \text{ Wiener}(\phi_w^Y(x, y)); \text{ Filter Size } [11, 11]$	(22-23)
Laplacian	$\mathfrak{K}(x, y) = \frac{1}{2\pi} \nabla_{\perp}^2 \left[\text{Im} \left\{ \left(\frac{1}{P(x, y)} \nabla_{\perp}^2 P(x, y) \right) - \nabla_{\perp}^2 \phi_w^{WF}(x, y) \right\} \right];$ $P(x, y) \propto \phi_w^{WF}(x, y)$	(18)
Scaling $\mathfrak{K}(x, y)$	$k(x, y) = s_{i, \min} * \mathfrak{K}(x, y)$	(19-20)
FLED	$\phi_w'(x, y) = B(x, y) \phi_w^{NF}(x, y) + F(x, y)$	(27-29)
Updated Laplacian	$\mathfrak{K}(x, y) = \frac{1}{2\pi} \nabla_{\perp}^2 \left[\text{Im} \left\{ \left(\frac{1}{P(x, y)} \nabla_{\perp}^2 P(x, y) \right) - \nabla_{\perp}^2 \phi_w'(x, y) \right\} \right];$ $P(x, y) \propto \phi_w'(x, y)$	(18)
Scaling $\mathfrak{K}(x, y)$	$k(x, y) = s_{i, \min} * \mathfrak{K}(x, y)$	(19-20)
Unwrapped Phase	$\phi(x, y) = \phi_w'(x, y) + s * \frac{1}{2\pi} \nabla_{\perp}^2 \left[\text{Im} \left\{ \left(\frac{1}{P(x, y)} \nabla_{\perp}^2 P(x, y) \right) - \nabla_{\perp}^2 \phi_w'(x, y) \right\} \right]$	(21)

Algorithm III: High Noise or Background Wrapped Phase

Quality Map	$\phi_w^Q(x, y) = \phi_w(x, y) Q(x, y)$	(24-26)
Denoising	$\phi_w^Y(x, y) = SD - ROM(\phi_w^Q(x, y));$ $\phi_w^{WF}(x, y) = 2D \text{ Wiener}(\phi_w^Y(x, y)); \text{ Filter Size } [11, 11]$	(22-24)
Laplacian	$\mathfrak{K}(x, y) = \frac{1}{2\pi} \nabla_{\perp}^2 \left[\text{Im} \left\{ \left(\frac{1}{P(x, y)} \nabla_{\perp}^2 P(x, y) \right) - \nabla_{\perp}^2 \phi_w^{WF}(x, y) \right\} \right];$ $P(x, y) \propto \phi_w^{WF}(x, y)$	(18)
Scaling $\mathfrak{K}(x, y)$	$k(x, y) = s_{i, \min} * \mathfrak{K}(x, y)$	(19-20)
FLED	$\phi_w'(x, y) = B(x, y) \phi_w^{NF}(x, y) + F(x, y)$	(27-29)
Updated Laplacian	$\mathfrak{K}(x, y) = \frac{1}{2\pi} \nabla_{\perp}^2 \left[\text{Im} \left\{ \left(\frac{1}{P(x, y)} \nabla_{\perp}^2 P(x, y) \right) - \nabla_{\perp}^2 \phi_w'(x, y) \right\} \right];$ $P(x, y) \propto \phi_w'(x, y)$	(18)
Scaling $\mathfrak{K}(x, y)$	$k(x, y) = s_{i, \min} * \mathfrak{K}(x, y)$	(19-20)
Unwrapped Phase	$\phi(x, y) = \phi_w'(x, y) + s * \frac{1}{2\pi} \nabla_{\perp}^2 \left[\text{Im} \left\{ \left(\frac{1}{P(x, y)} \nabla_{\perp}^2 P(x, y) \right) - \nabla_{\perp}^2 \phi_w'(x, y) \right\} \right]$	(21)

2.6. Performance assessment

The performance of the phase unwrapping algorithms was carried out using the the standard deviation of the resolved phase error, given by

$$\sigma_{\epsilon} = \sqrt{\mathbf{E}[\epsilon^2] - \mathbf{E}[\epsilon]^2}$$

where $\epsilon = \phi_i - \phi_r$ are the phase difference between the simulated ideal phase ϕ_i and the recovered phase map ϕ_r [16]. We also evaluated the execution time of the FFT, [12] DCT, [14] and proposed scaled and FLED FFT based Laplacian phase unwrapping algorithms on a MacBook Pro running Catalina 10.15.7 (2.2 GHz Quad-Core Intel i7 processor, 16 GB RAM) and MATLAB R2019b.

3. Results and discussion

3.1. Noise removal

To assess the degree of noise that was successfully removed via the SD-ROM and with 2D Wiener treatments, the gradient maps of the wrapped phaseogram were inspected. As proposed in the Methods section of the paper, the gradients of the wrapped phaseogram are expected to be zero, or near zero within the $[-\pi, \pi]$ boundaries. Presented in Fig. 4 are the wrapped phaseogram, the noise reduction treatment, and the gradients at regions featuring both high degrees of noise and fringe density.

Fig. 4 presents the results associated with the noise-reducing treatments, the input wrapped phaseogram contained $\sigma = 0.5222$ standard deviations of added speckle noise. The local phase gradients $\Delta\phi_w(x, y)$ in Fig. 3(a) present a high degree of discontinuity within the $(-\pi, \pi)$ reaching values that are similar to those expected near or at the wrapping boundary, this feature makes it particularly difficult to unwrap the phaseogram using the spectral differentiation method proposed by [Eqs. (11) and (13)], see Fig. 5 for results. The local phase gradients $\Delta\phi_w(x, y)$ of the wrapped phaseogram upon 2D Wiener filtration Fig. 3(b) generated feature smooth, near-zero, phase gradients within the $(-\pi, \pi)$ region and present the expected gradient values at or near the $[-\pi, \pi]$ boundary. However, when compared to the local phase gradients of the SD-ROM filtered phaseogram Fig. 3(c) the 2D-Wiener filter appears to miss impulse like noise near or at the $[-\pi, \pi]$ boundaries. This phenomenon can be detrimental to the relative phase unwrapping process, as corrupted wrapped boundaries can lead to unreliable $k(x, y)$ values necessary for phase unwrapping. It is noted that the SD-ROM filter alone fails to remove non-impulse like corrupted data, where the local phase gradients within the $(-\pi, \pi)$ region displays some degree of discontinuous behavior. The proposed SD-ROM + 2D-Wiener filtering mechanism in Fig. 3(d) presents a superior local phase gradient $\Delta\phi_w(x, y)$ mapping, where both impulse and non-impulse like noise has been removed from both the inner $(-\pi, \pi)$ and $[-\pi, \pi]$ boundaries. It is noted that similar results were obtained from a variety of wrapped noisy phaseogram, see Fig. 7. This particular wrapped, noisy phaseogram Fig. 4(a) was used to accentuate noise removal from regions containing high noise and wrapped fringe densities.

3.2. Phase unwrapping

Being that the presented phase unwrapping algorithms are in part largely based on work described by Schofield et al., 2003, and subsequently Shi et al., 2010, we have decided to compare the spectral differentiation technique established via [Eq. (21)] to their phase unwrapping techniques. Note, Schofield et al., 2003, first presented the Laplacian-spectral differentiation-based solution for wrapped signals. Shi et al., 2010, built on Sheffield's proposed solution by demonstrating that the solution was valid not only for the FFT but also for the discrete cosine transform (DCT). However, in both works, the authors suggested rounding the $k(x, y)$ value(s) to the nearest integer value to obtain the final unwrapped phaseogram. The discretization of $k(x, y)$ is correct, however,

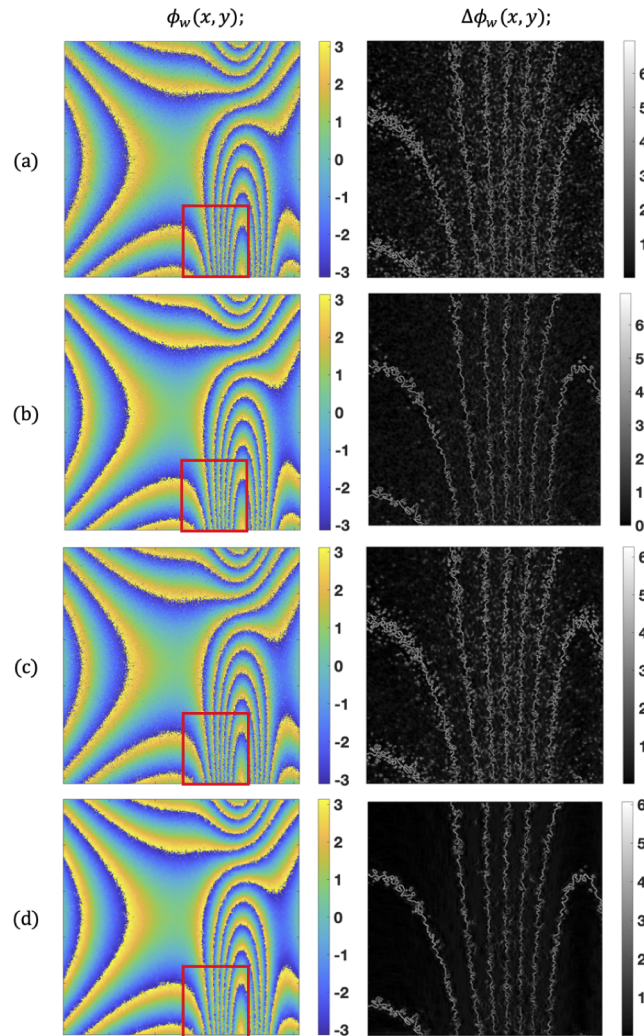


Fig. 4. Data presented herein is from a simulated noisy phaseogram containing $\sigma = 0.5222$ standard deviations of added speckle noise. The first column labeled $\phi_w(x, y)$ contains a four 1024×1024 images of wrapped phaseograms with (a) no treatment, (b) 2D Wiener filtering using an $[11, 11]$ window size, (c) SD-ROM filter, and (d) the SD-ROM, 2D Wiener filtering combination. The red square highlights the regions of interest presented in column two. The second column labeled $\Delta\phi_w(x, y)$ contains 300×324 images of the gradients of the wrapped phaseograms with images (a-d) containing the same treatment as their column one counterparts.

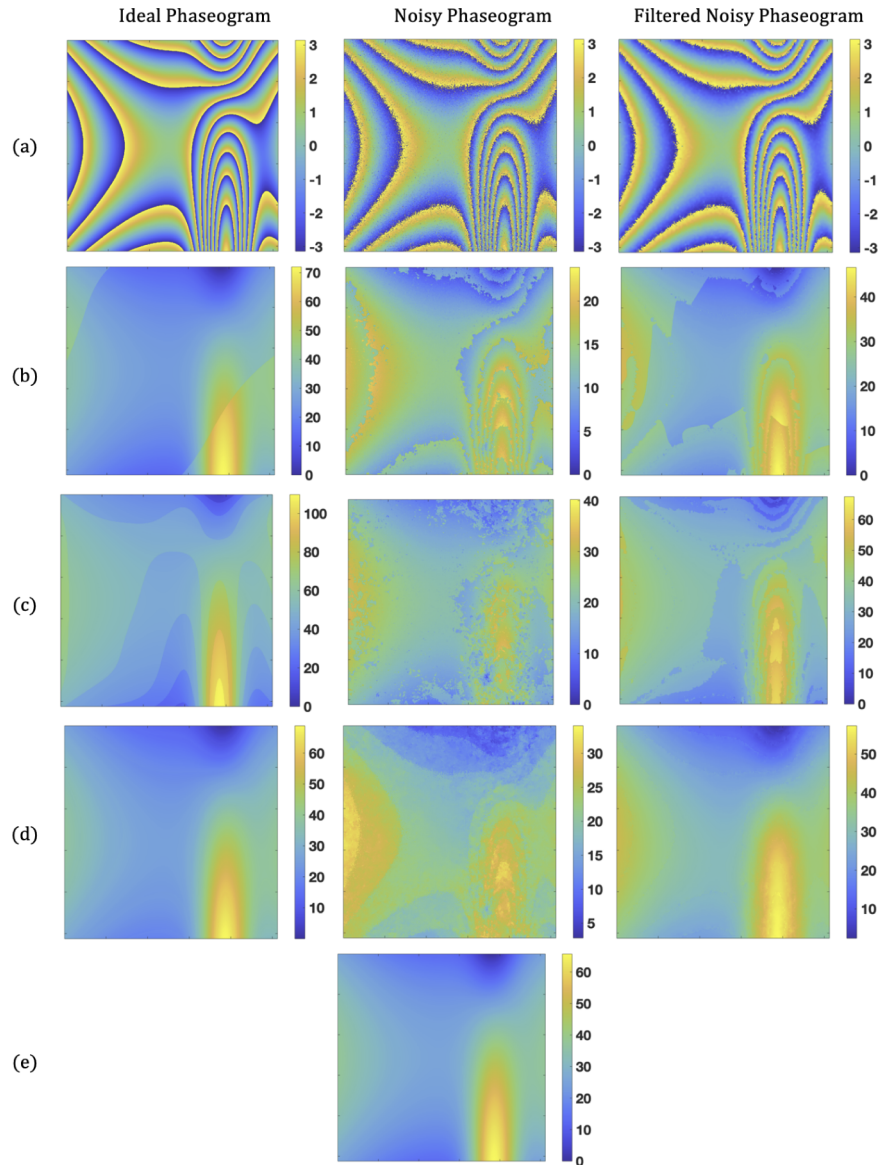


Fig. 5. Data presented herein is from a simulated noisy phaseogram containing a $\sigma = 0.5222$ standard deviations of added speckle noise. The first column labeled Ideal Phaseogram contains a four 1024×1024 images of wrapped noise-free phaseogram with (a) the wrapped phaseogram, (b) the unwrapped solution via the Schofield FFT method, (c) the unwrapped solution via the Shi DCT method, (d) the proposed $k(x, y)$ scaling method, and (e) the ideal unwrapped phase. The second column contains a noisy wrapped phaseogram with $\sigma = 0.5222$ simulated noise, the noise is not removed from the wrapped phaseogram and is processed with the phase unwrapping treatments mentioned in column one (b-d). Similarly, the third column contains a noisy wrapped phase that has been processed using the SD-ROM and 2D Wiener filtering combination; the filtered wrapped phaseogram is then processed with the phase unwrapping treatments mentioned in columns one and two (b-d).

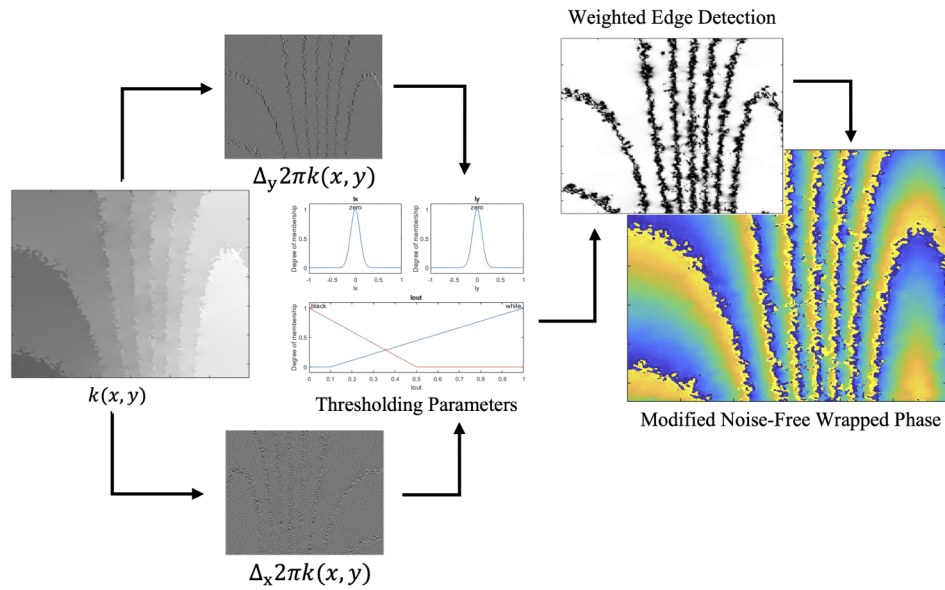


Fig. 6. The schematic demonstrates the data utilized and created by the Fuzzy Logic Edge Detection routine. The relevant input data comes from taking the local x - and y - directional gradients of the scaled $s \times 2\pi\tilde{k}(x, y)$ recovered phase wrapping intervals. The Fuzzy Logic Edge Detection routine weighs whether or not to designate a pixel as an edge based on the input thresholding parameters. Thereafter a Weighted-Gradient Based output matrix locates regions with high likelihoods of forming $[-\pi, \pi]$ conditions.

because of the inherent scaling effects incurred by the spectral differentiation technique, $k(x, y)$ cannot simply be rounded to the nearest integer as the image will incur unwanted rounding error in the unwrapped phase signal when $k(x, y)$ approaches $n \pm 0.5; n \in \mathbb{Z}$. Apart from the several modifications mentioned for pre-processing and post-processing wrapped/unwrapped phaseogram the major difference between Schofield and Shi's phase retrieval technique and the one proposed in this article is the $k(x, y)$ estimation technique, where we propose the use of $\sin(\pi\tilde{k}(x, y))$ presented in [Eq. (22)]. This proposal is based on the assumption that the wrapped signal varies from the unwrapped phase signal by $2\pi k(x, y)$, where $k(x, y)$ is an integer value, the function $\sin(\pi\tilde{k}(x, y))$ will be zero for all $k(x, y)$ given that the unwrapped $\tilde{k}(x, y)$ values are equal to the expected $k(x, y)$. However, in the case, that scaling is synthetically introduced into the wrapped phaseogram, $\sin(\pi\tilde{k}(x, y))$ will be expected to be non-zero. [Eqs. (22–23)] propose finding the scaling value by minimizing the global number of non-zero pixels in the $\sin(\pi\tilde{k}(x, y))$ identity. Presented in Fig. 5 are the unwrapping results via Shi, Schofield, and the proposed scaling method, again described by [Eqs. (22–23)].

Three treatments were used to compare the performance of each phase unwrapping algorithm and their proposed $k(x, y)$ estimation techniques, these included evaluating each phase unwrapping algorithm for ideal wrapped phase, noisy wrapped phase, and noise filtered wrapped phaseogram. The noisy wrapped phaseogram used for comparison contained $\sigma = 0.5222$ standard deviations of added speckle noise and had areas of relatively high fringe densities. It is noted that the scaling $k(x, y)$ phase unwrapping technique presented via Algorithm I: Noise-Free Phase Unwrapping and Algorithm II: Noisy Phase Unwrapping, does not make use of the proposed Fuzzy Logic Edge Detection technique described by [Eqs. (27–29)] as the purpose is to solely compare the phase unwrapping performance.

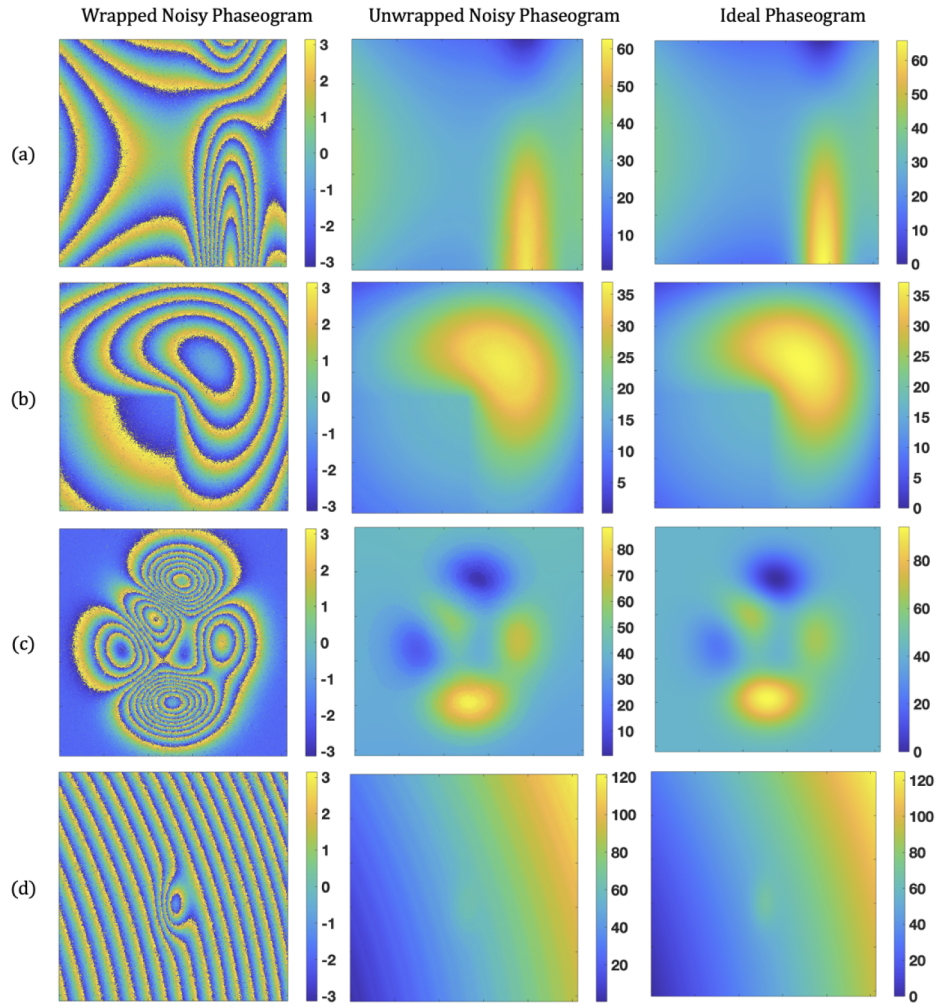


Fig. 7. Data presented herein is from a simulated noisy phaseogram containing varying degrees of noise, denoted by σ_n . The first column labeled Wrapped Noisy Phaseogram contains four 1024×1024 images of wrapped noisy phaseograms. The second column labeled Unwrapped Noisy Phaseogram contains four 1024×1024 images of the unwrapped noisy phaseograms using Algorithm II, where the FLED routine was implemented. The third column labeled Ideal Phaseogram contains four 1024×1024 images of the ideal unwrapped phaseograms.

Notice that both the Schofield and Shi algorithms struggle to unwrap the ideal wrapped interferogram, with notable jumps and dips in phase, coming from the rounded $k(x, y)$ values, whilst the proposed algorithm does not. The noisy and noise filtered wrapped phaseogram are challenging to unwrap because of the discontinuous local phase gradients $\Delta\phi_w(x, y)$ as well as the corrupt wrap boundaries, however, all three algorithms perform better with the SD-ROM, 2D Wiener filter; notably the proposed scaling $k(x, y)$ qualitatively outperforms the Schofield and Shi algorithms in all three tests. See Table 1 for an assessment of execution time and standard deviation of the resolved phase error.

Table 1. Ideal and filtered noisy phase unwrapping execution time and error.

Approach	Ideal		Filtered Noisy	
	t	σ_ϵ	t	σ_ϵ
FFT [12]	0.796 s	2.4190	5.492 s	4.2658
DCT [14]	0.533 s	6.5691	5.647 s	4.7497
Scaled FFT	1.968 s	1.8457	6.835 s	4.1338

3.3. Fuzzy Logic Edge detection

As mentioned in the Methods section of the paper, the local phase gradients $2\pi\Delta_x k(x, y)$ and $2\pi\Delta_y k(x, y)$ in between and at $[-\pi, \pi]$ boundaries are important to successfully unwrapping of wrapped phaseogram using the Laplacian spectral differentiation approach described via [Eq. (21)]. Notably, the Laplacian spectral differentiation technique relies on the assumption that the wrapped and unwrapped local phase gradients are equal within the $(-\pi, \pi)$ and that the $[-\pi, \pi]$ boundaries occur where the signal has been wrapped with the subsequent region differs by a factor of $\pm 2\pi k(x, y)$; however, due to noise corruption and increased fringe density, such boundaries can be difficult to process. Thus, the Fuzzy Logic Edge Detection (FLED) routine was added to Algorithm II and III, in an effort to both amplify the existing $[-\pi, \pi]$ pixel regions as well as approximate the location of corrupted $[-\pi, \pi]$ boundaries, via a local gradient thresholding technique.

Described via [Eqs. (27–29)] and Fig. 3, the FLED routine uses the local gradient values of the recovered-scaled $k(x, y)$ matrix. Note that the $k(x, y)$ matrix has unique properties stemming from the relationship between $\phi(x, y)$ and $\phi_w(x, y)$, namely that $k(x, y) \in \mathbb{Z}$; where the local phase gradients within $(-\pi, \pi)$ are zero, for the exception of the boundaries where wrapping occurs, here the local phase gradients should be equal to 2π . Using these parameters, and the recovered, scaled $s \times 2\pi k(x, y)$ parameter can allow us to estimate boundary conditions based on local phase gradients that exceed 0.

For this article seven thresholding values were chosen to detect edges in a variety of noisy wrapped phaseogram; these values remained unchanged. For regions where $2\pi\Delta_x k(x, y)$ or $2\pi\Delta_y k(x, y)$ are zero, the Fuzzy Logic Routine associates a value of 1, high likelihood of non-edge-like boundary. Measuring the likelihood of edge-like boundaries is associated with a value of zero. Zero membership was determined by specifying the values $S_x = 0.1$ and $S_y = 0.1$ around 0, these values specify the standard deviation for the zero membership function for the gradient inputs $2\pi\Delta_x k(x, y)$ and $2\pi\Delta_y k(x, y)$; where if the input gradient matrices are within 0.1 standard deviations of 0, they are weighted closer to 1 than to 0. The exact weighting schematic can be seen in Fig. 6. It is noted that the Fuzzy Logic Edge Detection thresholding values remained fixed for the entire study and the data presented in Fig. 7 and Fig. 8.

Using the Fuzzy Logic Edge Detection routine established via MATLAB's Fuzzy Logic application, various noisy phaseograms with varying degrees of fringe densities and noise, σ_n , were unwrapped using Algorithm II.

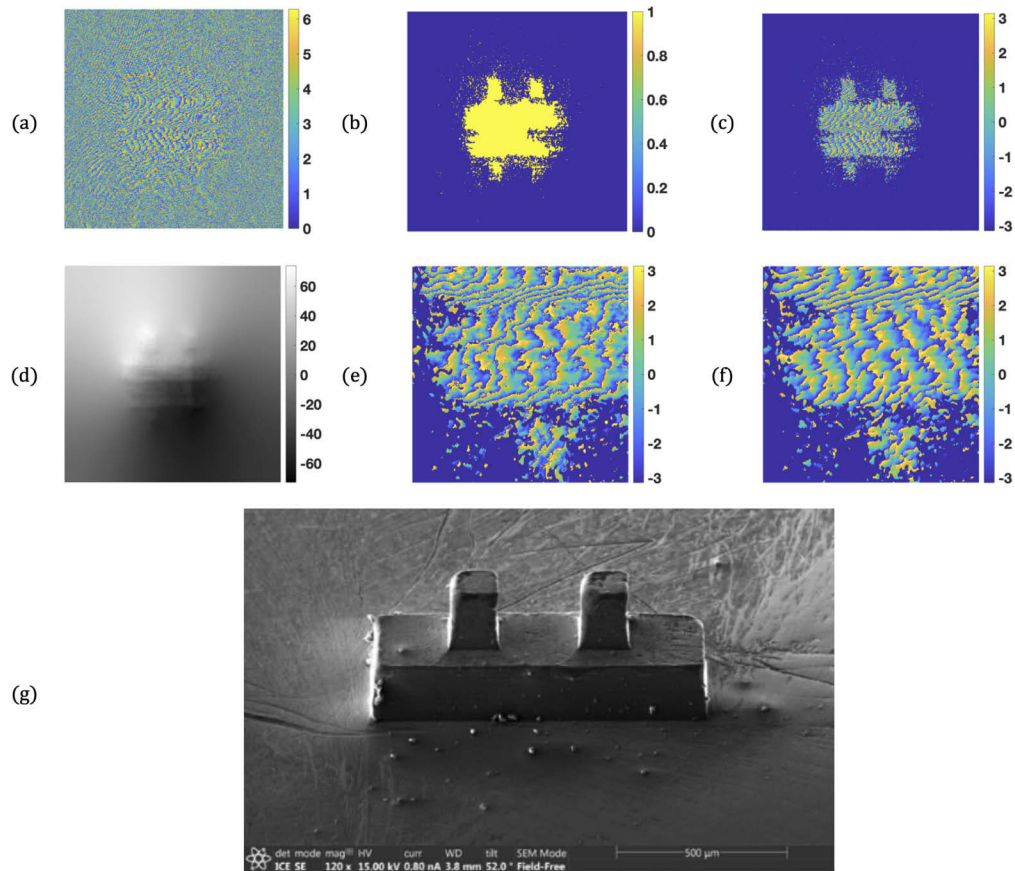


Fig. 8. Data presented herein were obtained experimentally in a recent RF emulator study conducted in our lab. Briefly, phase information was retrieved from optical signals scattered by the object of interest, using a Twyman-Green interferometer configuration. The retrieved wrapped phaseogram (a) contains a very high degree of noise. Here PDV filtering (b), along with the SD-ROM, 2D Wiener filtering (c), was implemented to generate a noise filtered phaseogram wrapped phaseogram. The recovered unwrapped phaseogram (d) was re-wrapped (e) and compared to the original denoised wrapped phaseogram (f). (g) Contains an SEM of the experimental model utilized.

The use of the modified spectral differentiation technique developed in [Eq. (21)], in combination with SD-ROM filtering [Eqs. (22–23)] and the FLED [Eqs. (27–29)] routine displayed reliable phase retrieval across a variety phaseogram with varying noise profiles and fringe densities. However, it is noted that there some of the retrieved phaseograms displayed minor scaling issues, particularly Figs. 7(a) and 7(c). As observed, the scaling effects occurred in regions with very high noise and fringe density. Modifications to the FLED parameters are highly encouraged in such situations; where variable FLED parameters should be applied on a case by case basis if wrapped phaseogram present very high noise and or fringe densities. For a comparison of the of execution time and standard deviation of the resolved phase error between the FFT, DCT, and the proposed FLED phase unwrapping algorithms see Table 2. All three algorithms utilized the SD-ROM and 2D Wiener filter as a denoising mechanism.

Note that the execution time of the FLED FFT technique is substantially greater than the FFT and DCT and phase unwrapping algorithms. The increase in execution time comes solely from the implementation of the FLED algorithm; where, as observed in Table 1, the execution time of the proposed scaled Laplacian technique is comparable to those presented by the FFT and DCT techniques. Alterations to MATLAB's Fuzzy Logic Application could drastically decrease run time.

Table 2. Noisy phase unwrapping execution time and error.

Phaseogram	σ_n	FFT		DCT		FLED FFT	
		t	σ_ϵ	t	σ_ϵ	t	σ_ϵ
(a)	0.5222	4.759 s	4.269	4.485 s	4.750	41.191 s	2.555
(b)	0.4933	4.758 s	1.535	4.476 s	2.476	39.550 s	0.696
(c)	0.6202	4.792 s	7.132	4.520 s	6.252	41.000 s	3.287
(d)	0.6510	5.123 s	5.398	4.496 s	15.594	43.365 s	1.311

Algorithm III was designed for unwrapped phaseograms containing very high levels of noise coming from the imaging source, experimental setup, or background noise phase wrapping as demonstrated by Fig. 8(a) [6]. Algorithm III was tested with data acquired from an RF emulator study [6]. As seen in Fig. 8 the acquired wrapped data had a very high degree of noise. The unwrapped result was re-wrapped using the wrapping operator discussed, see Eq. (1), and thereafter compared to the original denoised wrapped phaseogram. The results concur with the original data, where fringe density and direction are maintained. Furthermore, the re-wrapped results demonstrate greater fringe continuity when compared to the original wrapped data. As demonstrated, Algorithm III handles the phase unwrapping, denoising, and edge detection routine for the very high noise wrapped phaseogram shown in Fig. 8. It is noted, as before, that the thresholding parameters established for the Fuzzy Logic Edge Detection routine remained unchanged. Although the following is speculative and outside of the scope of the study, the FLED routine may be able to provide "better" phase unwrapping results on a case by case basis with the modification of the thresholding parameters.

4. Conclusion

Phase unwrapping in the presence of noise and unreliable fringes can pose a difficult problem for global phase unwrapping algorithms; however, in utilizing a directed approach that aims to tackle both problems simultaneously, phase unwrapping such signals can be reliably accomplished. In this article we have presented a three-pronged approach to successfully unwrap 2D signals; namely, i) SD-ROM based denoising, ii) reliable $k(x, y)$ estimation, iii) FLED based fringe quality improvement. The proposed algorithm has relatively low computation and implementation complexity and has many variable parameters in both the denoising and FLED routines, making it

well suited for a variety of complex phase unwrapping cases. Further work with complex wrapped phaseograms is suggested to test the working limits of the proposed algorithm. Furthermore, although not in the scope of the present article, the following additions are suggested for improvement on the current algorithm: (i) pixel by pixel $k(x, y)$ scaling, see [Eqs. (19–20)], in combination with a merit function to optimize phase retrieval and (ii) implementation the robust FLED technique not only for edge detection but also for gradient quality detection and estimation at the $[-\pi, \pi]$ boundaries, as well as optimization of MATLAB's Fuzzy Logic App to decrease overall execution time.

Funding. Office of Naval Research (N00014-14-1-0505).

Acknowledgments. The authors would like to thank Dr. Pascal Picart, Professor at LeMans Université for allowing us access to his simulated wrapped phaseograms database. The authors also acknowledge the support of the Office of Naval Research under grant #N00014-14-1-0505.

Disclosures. The authors declare no conflicts of interest.

References

1. W. Li, A. V. Avram, B. Wu, X. Xiao, and C. Liu, "Integrated laplacian-based phase unwrapping and background phase removal for quantitative susceptibility mapping," *NMR Biomed.* **27**(2), 219–227 (2014).
2. D. Cohoe, I. Hanczarek, J. K. Wallace, and J. Nadeau, "Multiwavelength digital holographic imaging and phase unwrapping of protozoa using custom fiji plug-ins," *Front. Phys.* **7**, 94 (2019).
3. M. D. Pritt, "Phase unwrapping by means of multigrid techniques for interferometric sar," *IEEE Trans. Geosci. Remote. Sens.* **34**(3), 728–738 (1996).
4. G. S. Spagnolo, G. Guattari, C. Sapia, D. Ambrosini, D. Paoletti, and G. Accardo, "Contouring of artwork surface by fringe projection and fft analysis," *Opt. Lasers Eng.* **33**(2), 141–156 (2000).
5. M. Padilla, M. Servin, and G. Garnica, "Fourier analysis of rgb fringe-projection profilometry and robust phase-demodulation methods against crosstalk distortion," *Opt. Express* **24**(14), 15417–15428 (2016).
6. R. S. Ketchum, P. E. Alcaraz, and P.-A. Blanche, "A 300 thz tabletop radar range with phase retrieval capabilities," *Nature - Scientific Reports*, submitted for publication.
7. R. Kulkarni and P. Rastogi, "Simultaneous unwrapping and low pass filtering of continuous phase maps based on autoregressive phase model and wrapped kalman filtering," *Opt. Lasers Eng.* **124**, 105826 (2020).
8. J. Pineda, J. Bacca, J. Meza, L. A. Romero, H. Arguello, and A. G. Marrugo, "Spud: simultaneous phase unwrapping and denoising algorithm for phase imaging," *Appl. Opt.* **59**(13), D81–D88 (2020).
9. J. Tan, Y. Bai, B. Dong, and Z. He, "Phase noise reduction in wavelength scanning interferometry using a phase synthesis approach," *Opt. Commun.* **475**, 126295 (2020).
10. X. Wang, S. Fang, X. Zhu, K. Kou, Y. Liu, and M. Jiao, "Phase unwrapping based on adaptive image in-painting of fringe patterns in measuring gear tooth flanks by laser interferometry," *Opt. Express* **28**(12), 17881–17897 (2020).
11. K. Yan, Y. Yu, T. Sun, A. Asundi, and Q. Kema, "Wrapped phase denoising using convolutional neural networks," *Opt. Lasers Eng.* **128**, 105999 (2020).
12. M. A. Schofield and Y. Zhu, "Fast phase unwrapping algorithm for interferometric applications," *Opt. Lett.* **28**(14), 1194–1196 (2003).
13. V. Volkov, Y. Zhu, and M. De Graef, "A new symmetrized solution for phase retrieval using the transport of intensity equation," *Micron* **33**(5), 411–416 (2002).
14. W. Shi, Y. Zhu, and Y. Yao, "Discussion about the dct/fft phase-unwrapping algorithm for interferometric applications," *Optik* **121**(16), 1443–1449 (2010).
15. E. Barnhill, P. Kennedy, C. L. Johnson, M. Mada, and N. Roberts, "Real-time 4d phase unwrapping applied to magnetic resonance elastography," *Magn. Reson. Med.* **73**(6), 2321–2331 (2015).
16. S. Montessoro and P. Picart, "Quantitative appraisal for noise reduction in digital holographic phase imaging," *Opt. Express* **24**(13), 14322–14343 (2016).
17. E. Abreu and S. K. Mitra, "A signal-dependent rank ordered mean (sd-rom) filter—a new approach for removal of impulses from highly corrupted images," in *1995 International Conference on Acoustics, Speech, and Signal Processing*, vol. 4 (1995), pp. 2371–2374 vol.4.
18. D. C. Ghiglia and M. D. Pritt, *Two-dimensional Phase Unwrapping : Theory, Algorithms, and Software* (Wiley, New York, 1998).
19. Y. Lu, X. Wang, and X. Zhang, "Weighted least-squares phase unwrapping algorithm based on derivative variance correlation map," *Optik* **118**(2), 62–66 (2007).