RESEARCH ARTICLE

Tile-Based Two-Dimensional Phase Unwrapping for Digital Holography Using a Modular Framework

Georgios C. Antonopoulos[1]*, **Benjamin Steltner**[1], **Alexander Heisterkamp**[2], **Tammo Ripken**[1], **Heiko Meyer**[1]

1 Biomedical Optics Department, Laser Zentrum Hannover e.V., Hanover, Germany, **2** Institute of Quantum Optics, Gottfried Wilhelm Leibniz Universität Hannover, Hanover, Germany

* g.antonopoulos@lzh.de

Citation: Antonopoulos GC, Steltner B, Heisterkamp A, Ripken T, Meyer H (2015) Tile-Based Two-Dimensional Phase Unwrapping for Digital Holography Using a Modular Framework. PLoS ONE 10(11): e0143186. doi:10.1371/journal.pone.0143186

Editor: Heye Zhang, Shenzhen institutes of advanced technology, CHINA

Received: August 23, 2015

Accepted: November 1, 2015

Published: November 24, 2015

Data Availability Statement: All relevant data are within the paper and its Supporting Information files. The source code to the software can be found under https://github.com/gc-ant/digiholo2D.

Funding: The authors have no support or funding to report.

Competing Interests: The authors have declared that no competing interests exist.

Abstract

A variety of physical and biomedical imaging techniques, such as digital holography, interferometric synthetic aperture radar (InSAR), or magnetic resonance imaging (MRI) enable measurement of the phase of a physical quantity additionally to its amplitude. However, the phase can commonly only be measured modulo 2π, as a so called wrapped phase map. Phase unwrapping is the process of obtaining the underlying physical phase map from the wrapped phase. Tile-based phase unwrapping algorithms operate by first tessellating the phase map, then unwrapping individual tiles, and finally merging them to a continuous phase map. They can be implemented computationally efficiently and are robust to noise. However, they are prone to failure in the presence of phase residues or erroneous unwraps of single tiles. We tried to overcome these shortcomings by creating novel tile unwrapping and merging algorithms as well as creating a framework that allows to combine them in modular fashion. To increase the robustness of the tile unwrapping step, we implemented a model-based algorithm that makes efficient use of linear algebra to unwrap individual tiles. Furthermore, we adapted an established pixel-based unwrapping algorithm to create a quality guided tile merger. These original algorithms as well as previously existing ones were implemented in a modular phase unwrapping C++ framework. By examining different combinations of unwrapping and merging algorithms we compared our method to existing approaches. We could show that the appropriate choice of unwrapping and merging algorithms can significantly improve the unwrapped result in the presence of phase residues and noise. Beyond that, our modular framework allows for efficient design and test of new tile-based phase unwrapping algorithms. The software developed in this study is freely available.

Introduction

The ability to measure the phase of a complex physical signal is a key advantage of imaging methods such as digital holography [1], interferometric synthetic aperture radar (InSAR) [2], or magnetic resonance imaging [3]. However, due to the 2π periodicity of a phase map, the resulting measurement typically yields the true phase modulo 2π. Such a phase map is called *wrapped* and will show distinctive jumps (wraps) of magnitude 2π. From a mathematical point of view this is a consequence of the inverse tangent operation that is typically involved in retrieving the phase of a complex signal. Phase unwrapping is the process of removing the 2π ambiguity from the wrapped phase map. This process is trivial for the one dimensional case, but it becomes extremely challenging in two dimensions, even more so when measurement errors deteriorate the quality of the wrapped phase map [4]. Phase unwrapping has been studied for decades, yet due to a growing number of applications, it remains an active area of research until today [5–9].

Phase unwrapping algorithms can be classified in several ways. We follow Harráez *et al.* by dividing phase unwrapping algorithms into three major categories: Global algorithms, path following algorithms, and region-based algorithms [10]. Global algorithms typically work by minimizing a cost function that takes all pixels of the wrapped phase map into account [11]. Solutions are obtained based on transform methods [12–14], Bayesian estimation [7, 15, 16], or network optimization [4, 17, 18]. Path following algorithms unwrap the phase map by detecting 2π phase jumps between neighboring pixels along a path. These algorithms operate by using simple linear paths [6], by using sophisticated branch-cut algorithms [19–22], or by choosing an unwrapping path based on a quality criterion [10, 23]. Region-based algorithms work by dividing the phase map into subregions. Phase unwrapping is first performed on subregions and unwrapped regions are grown or merged gradually [24–26].

Tile-based algorithms are a special case of the region-based algorithms [25, 27–30]. The phase map is tessellated into rectangular subregions called tiles that are processed in two steps: First, tiles are unwrapped individually and second, tiles are merged to a continuous phase map. This approach has some very appealing properties, most notably that the tile unwrapping step can be implemented computationally efficiently by parallelization. However tile-based algorithms are prone to error propagation in case of failed unwraps of single tiles as well as phase residues. In this paper we developed a modular C++11 framework for tile-based phase unwrapping. By casting the tile-based approach into a modular framework we were able to implement different tile unwrapping and merging algorithms. Next to established algorithms we also implemented original versions of tile unwrapping and merging strategies. We then compared various combinations of unwrapping and merging algorithms to existing approaches, both tile-based and pixel-based. We could show that our algorithms can improve the result of phase unwrapping in the presence of noise and residues. Our modular software framework enabled us to easily combine different tile unwrapping and merging algorithms for the quantification of the resulting phase unwraps. It was designed as a tool for developing and testing tile-based phase unwrapping algorithms, thus ease of use and readability are emphasized over execution speed. While algorithms were implemented efficiently, the modularity of the framework does introduce an overhead that results in a speed penalty. The software developed for this study is fully documented and freely available as open source software. The source code, a precompiled binary and an ImageJ Plugin are accessible using the GitHub repository https://github.com/gc-ant/digiholo2D.

Mathematical formalism

Here we will give a brief introduction to the notation and the mathematical formalism used throughout this paper, for detailed overviews see [11, 31]. The two-dimensional unwrapped phase distribution, which is a priori unknown, will be designated with $\phi_u(x, y)$. A wrap operator $\mathcal{W}$ is introduced to obtain the corresponding wrapped phase map $\phi_w(x, y)$:

$$\phi_w(x,y) := \mathcal{W}\phi_u(x,y) := \phi_u(x,y) - 2\pi n(x,y), \text{ s.t. } \phi_w(x,y) \in [-\pi,\pi), n(x,y) \in \mathbb{Z}. \quad (1)$$

The wrap operator adds a signed integer multiples of 2π to the pixels of the unwrapped phase map so that each pixel gets shifted to the interval $[-\pi, \pi)$. Phase unwrapping is the problem of obtaining this integer jump map from the wrapped phase with the aim of obtaining the unwrapped (physical) phase map. The true physical phase map need not be a continuous function but it can be piecewise continuous. In this paper we assume that the phase jumps at locations of discontinuities are of magnitude less than π. This assumption holds for smoothly varying objects such as cells or appropriate technical surfaces when imaged using digital holography [32–37]. Furthermore we require that discrete sampling to a grid (x_i, y_j) is performed such that the *Itoh condition* is satisfied [6]:

$$-\pi < \phi_u(x_i, y_j) - \phi_u(x_m, y_n) < \pi \quad \forall \text{ next neighbors } (x_i, y_j), (x_m, y_n). \tag{2}$$

Under the assumptions above, the solution to the phase unwrapping problem is unique [31].

An important artifact of quantitative phase imaging are so called phase residues, which can be understood as follows: A true phase map ϕ_u of a complex physical quantity is a scalar function. Therefore the integral of the gradient $\vec{\nabla}\phi_u$ along any closed loop must evaluate to zero and the integration is path independent. This implies that for any closed loop along a corresponding wrapped phase map $\phi_w = \mathcal{W}\phi_u$ the number of positive 2π jumps between adjacent pixels will equal the number of negative jumps. This is true, if the non-wrapped phase map is either continuous or its discontinuities are of magnitude less than 2π and it is sampled according to the Itoh condition [38]. Real wrapped phase maps, however, will often violate this condition at sites termed *residues*[19]. For further illustration of residues refer to S2 Fig. The problem of unwrapping a wrapped phase distribution with residues does not have a unique solution.

Tile-based phase unwrapping

Let the two-dimensional wrapped phase map $\phi_w(x, y)$ be given on a discrete rectangular pixel raster of width N_X and height N_Y, so that x_i, $i = 1, \ldots, N_X$ and y_j, $j = 1, \ldots, N_Y$. The basic idea of tile-based phase unwrapping is to tessellate the wrapped phase map into $N_{\tau W} \times N_{\tau H}$ rectangular tiles $\tau_{w,h}$ with indices $w = 1, \ldots, N_{\tau W}$ and $h = 1, \ldots, N_{\tau H}$ [25, 27–30]. After that, phase unwrapping is performed in two consecutive steps:

1. Tile unwrapping: Each tile $\tau_{w,h}$ is unwrapped individually. This step can, in principle, be performed for all tiles simultaneously.
2. Tile merging: In this step, tiles are merged to a continuous surface using a *merging algorithm* by adding integer multiples of 2π to entire tiles.

This process is schematically shown in Fig 1. The total number of tiles for a given tessellation is given as $N_\tau = N_{\tau W} \times N_{\tau H}$.

Model-based phase unwrapping

Model-based phase unwrapping is a type of region based unwrapping, which uses model functions to achieve phase unwrapping on subregions [39–41]. These subregions are merged to a continuous phase map in a second step, which makes it straightforward to adapt it to a tile-based approach. The foundation of model-based phase unwrapping is given by the following fact: Given an arbitrary function $f(x, y)$ it is true that

$$|\phi_u(x, y) - f(x, y)| < \pi \Rightarrow \phi_u(x, y) = f(x, y) + \mathcal{W}(\phi_w(x, y) - f(x, y)). \tag{3}$$

This statement follows from Eq 1, because for any numbers $r \in [-\pi, \pi)$ and $n \in \mathbb{Z}$ it is true that

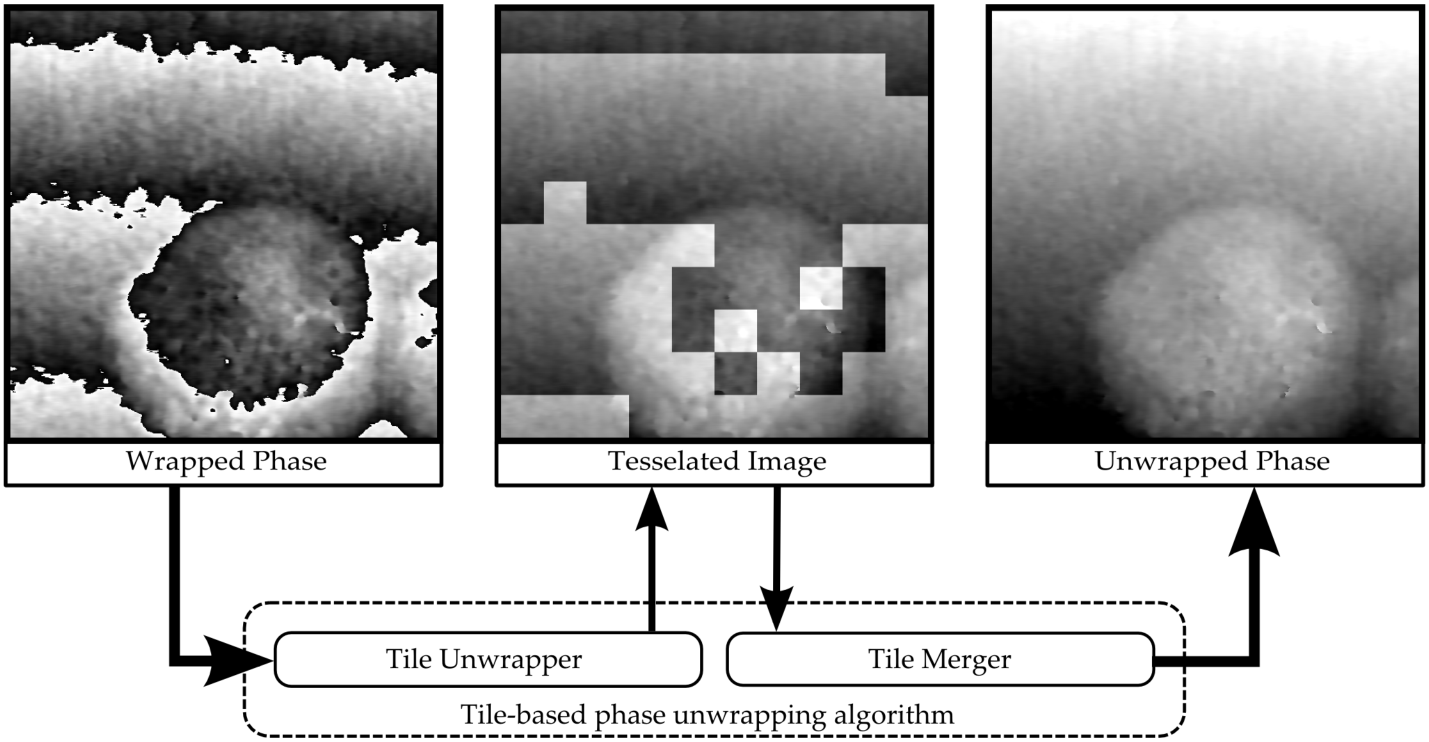

Fig 1. Tile-based phase unwrapping schematic. A wrapped phase image is tessellated into rectangular subregions called tiles. First these tiles are unwrapped individually and second they are merged to a continuous surface.

doi:10.1371/journal.pone.0143186.g001

$\mathcal{W}(r + 2\pi n) = \mathcal{W}(r) = r$. A way of intuitively understanding Eq 3 is to interpret f as a reasonably good guess for the unknown unwrapped phase ϕ_u. If such a guess can be obtained from the wrapped phase, then it is possible to calculate the unwrapped phase by adding a correction term $\mathcal{W}(\phi_w - f)$ to this guess function f. In model-based phase unwrapping a model for the guess function is assumed, e.g. polynomial functions [40, 41].

The derivative of the unwrapped phase can easily be obtained from the corresponding wrapped phase map [6]: For a discrete sampling under the Itoh condition it is true that

$$\delta_{x,y}\phi_u(x, y) = \mathcal{W}(\delta_{x,y}\phi_w(x, y)) \tag{4}$$

where δ_x and δ_y are the finite difference operators in x and y direction respectively. This information can be used to fit a model function to the wrapped phase data [41]. We propose to solve the least squares fitting problem by casting it into a linear system that is solved with singular value decomposition (SVD). This is detailed in the next section.

Algorithms

Model-based least squares unwrapping of single tiles (MLSQU)

Unwrapping individual tiles is the first step of the two-step algorithm as described above. Here $\phi_w(x, y)$ is the discretized wrapped phase distribution given on a tile $\tau_{w,\,h}$ and $\phi_u(x, y)$ be the corresponding unwrapped phase. The coordinate system (x, y) belongs to the tile $\tau_{w,\,h}$. Our model-based unwrapping approach is based on Eqs 3 and 4. We make an ansatz for the guess function as a linear combination of N_B base functions v_l

$$f(x,y,\vec{c},\rho) = \rho + \sum_{l=1}^{N_B} c_l \cdot v_l(x,y) \tag{5}$$

where $v_l(x, y)$ are analytically given functions, c_l are the expansion coefficients and ρ is a constant offset. We assume, without loss of generality, that the constant offset of f is solely given by ρ. The expansion coefficients are written as a vector $\vec{c} = (c_l)_{l=1,\ldots,N_B}$.

Calculating the expansion coefficients. To facilitate the transition to matrix notation we replace the position double index (x, y) with a single index k that enumerates the pixels in a row-major order. Thus we have $k = 1, \ldots, N^{\tau}_{px}$, where N^{τ}_{px} is the number of pixels in a tile. We now write any function $F(x, y)$ as $F(k)$. A solution for the function expansion coefficients is obtained by least squares fitting the gradient of f to the gradient of the unwrapped phase distribution obtained by Eq 4:

$$\chi^2(\vec{c}) = \sum_k \parallel \vec{\delta} f(k,\vec{c}) - \vec{\delta}\phi_u(k) \parallel_2^2 \ \rightarrow min.$$

This minimization is cast into a linear algebra problem by introducing a system matrix $\underline{A}$ and a vector $\vec{g}$ as follows:

$$\begin{aligned} \chi^2(\vec{c}) &= \parallel \underline{A} \cdot \vec{c} - \vec{g} \parallel_2^2 \rightarrow min. \\ \underline{A} &= \begin{pmatrix} \underline{A_x} \\ \\ \underline{A_y} \end{pmatrix} \in \Re^{2N^{\tau}_{px} \times N_B}, \ \text{with} \ (\underline{A_{x,y}})_{kl} = \delta_{x,y} v_l(k) \in \Re^{N^{\tau}_{px} \times N_B} \\ \vec{g} &= \begin{pmatrix} \vec{g}_x \\ \\ \vec{g}_y \end{pmatrix} \in \Re^{2N^{\tau}_{px}}, \ \text{with} \ (\vec{g}_{x,y})_k = \mathcal{W}\delta_{x,y}\phi_w(k) \in \Re^{N^{\tau}_{px}} \end{aligned} \tag{6}$$

The gradient vector $\vec{g}$ contains the derivatives of the phase distribution within a tile. The system matrix $\underline{A}$ is comprised of the values of the derivatives of the base functions on a tile. Minimization of expression 6 is performed by using the singular value decomposition (SVD) of the system matrix [42]. For a fixed set of base functions, the system matrix only depends on the geometry of a tile. This means that the SVD of the system matrix needs to be calculated only once for any set of tiles with the same width and height. Exploiting this fact is crucial in making this approach computationally feasible. For a tessellation into tiles with identical width and height, unwrapping will be performed with maximum efficiency. Numerical linear algebra was implemented using the Eigen C++ library [43].

Calculating the constant offset. Considering Eq 5 suggests determining ρ by subtracting the value of the linear combination of base functions with known expansion coefficients from the value of the tile at an arbitrary point (x_0, y_0). This approach is feasible in low noise settings and for applications where additional computational overhead is not desired. For maximum robustness against noise, we decided to incorporate Strand's tile unwrapping algorithm into our procedure [29]. Accordingly, we define a penalty function $\Pi(\tau)$ for a tile τ:

$$\Pi(\tau) = \langle \parallel \delta^1_x \phi_w(x,y) \parallel_2 \rangle + \langle \parallel \delta^1_y \phi_w(x,y) \parallel_2 \rangle \tag{7}$$

where $\delta^1_{x,y}$ are the first order forward difference operators and $\langle . \rangle$ signifies the average over all positions in the tile τ. Note that the application of the finite difference operator to the wrapped phase is not followed by a wrap operation, so that phase jumps will contribute with 2π to the penalty function. Thus this function is a measure for the number of phase jumps within a tile.

First we calculate

$$\phi_{rem}(x, y) := \mathcal{W}(\phi_w(x, y) - f(x, y, \vec{c}, \rho = 0)).$$

This is the remainder of the phase after the removal of the surface given by the model function f with $\rho = 0$. Next the constant ρ is calculated as

$$\rho = \underset{\rho' \in [0, 2\pi)}{\arg\min}\ \Pi(\mathcal{W}(\phi_d(x, y) - \rho') + \rho'). \tag{8}$$

This is the application of Strand's unwrapping approach to the remainder of the phase. A detailed description and an intuitive interpretation of Eqs 7 and 8 is found in [29]. Minimization of Eq 8 is performed by evaluating the penalty function at N_ρ equidistantly spaced values for $\rho \in [0, 2\pi)$. Finally, using Eq 3, the model-based tile unwrap is performed using the constant ρ as calculated above. Our tile unwrapping approach thus contains Strand's tile unwrapping method as a special case for a model function $f(x, y) = const. = \rho$.

Merging Algorithms

After the unwrap of all individual tiles is completed, neighboring tiles can show phase jumps of integer multiples of $\pm 2\pi$ or 0 with respect to each other. A merging algorithm, or *merger*, joins the tiles of a tessellated image to a continuous phase surface by adding integer multiples of $\pm 2\pi$ to entire tiles. Adding a value to a tile means adding this value to every pixel within the tile.

We term the set of neighboring pixels for two tiles τ_A, τ_B the *junction* $\mathcal{J}_{AB}$. The set of differences between neighboring pixels for a junction is introduced as

$$\mathcal{D}(\mathcal{J}_{AB}) := \{(P_B - P_A) | P_A \in \tau_A, P_B \in \tau_B \text{ are neighbor pixels}\}.$$

Since the phase distribution is tessellated into a rectangular grid, any tile at coordinates $A = (w_A, h_A)$ can have up to four neighbors, corresponding to the directions up $(w_A, h_A - 1)$, down $(w_A, h_A + 1)$, left $(w_A - 1, h_A)$ and right $(w_A + 1, h_A)$. To quantify the phase jumps between neighboring tiles, we introduce the mean difference $\Delta\mathcal{J}_{AB}$ and the variance $Var\mathcal{J}_{AB}$ of a junction as

$$\begin{aligned} \Delta\mathcal{J}_{AB} &:= Mean(\mathcal{D}(\mathcal{J}_{AB})) \\ Var\mathcal{J}_{AB} &:= Variance(\mathcal{D}(\mathcal{J}_{AB})) \end{aligned}$$

Algorithm 1 unwraps two neighboring tiles with respect to each other by using the mean difference of the corresponding junction. Algorithms in this paper are given in pseudocode notation.

```
Algorithm 1 Merge tile $\tau_A$ to tile $\tau_B$
Procedure MERGETILES ($\tau_A$, $\tau_B$)
    ▷ Find the mean difference of neighboring pixels:
    delta ← $\Delta\mathcal{J}_{AB}$
    ▷ Round difference to nearest integer multiple of $\pm 2\pi$:
    jumpval ← $2\pi\cdot$ round (delta/$2\pi$)
    Add jumpval to all pixels of $\tau_A$
```

Unidirectional tile merger. In [29], Strand *et al.* suggest a simple and fast merging algorithm that requires only one pass over each tile, see Algorithm 2. It is a modification of Itoh's classic one dimensional unwrapping approach [6].

```
Algorithm 2 Unidirectional tile merger (column-wise)
▷ Go through tiles column-wise left to right:
```

```
for $iw \leftarrow 1, \ldots, N_{\tau W}$ do
  ▷ Go through each column of tiles from top to bottom:
  for $j \leftarrow 1, \ldots, N_{\tau H}$ do
    ▷ Find top and left neighbors of current tile $\tau_{iw, ih}$, if they exist:
    neighbors $\leftarrow \{\tau_{iw-1, ih}, \tau_{iw, ih-1}\} \cap \{\tau_{w, h} | w = 1, \ldots, N_{\tau W}, h = 1, \ldots, N_{\tau H}\}$
    ▷ Find mean value of junction differences to neighbours:
    delta ← mean value of junction differences of tile $\tau_{iw, ih}$ to neighbors
    ▷Round difference to nearest integer multiple of $\pm 2\pi$:
   jumpval $\leftarrow 2\pi \cdot$ round(delta/$2\pi$)
   Add delta to all pixels of $\tau_{iw, ih}$
```

Reliability guided SRNCP merger (τSRNCP). We developed a reliability guided tile merger based on the SRNCP (*"sorting by reliability following a non-continuous path"*) algorithm by Herráez *et al.* [10]. The fundamental principle of the pixel-based SRNCP algorithm is to analyze the junctions of neighboring pixels and unwrap them sequentially according to a reliability value. Junctions with the highest reliability are processed first and unwrapped pixels are grouped to identify sets of pixels with no remaining phase jumps between each other.

The reliability of the pixel-based SRNCP algorithm makes use of the second derivative of the phase map. This quantity cannot simply be generalized to tiles. To adapt this algorithm to a tile-based approach, we introduce a new reliability measure for junctions and tiles, respectively. The reliability of a tile is calculated as follows: First, calculate the variance $Var\mathcal{J}_{AB}$ of all junctions of this tile and assign its average value to V. The reliability of the tile is defined as $R = V^{-1}$. This means that a tile has high reliability, when it segues into all of its neighbors continuously. This is also true if the tiles differ by multiples of 2π along the junction, since the variance is invariant under a constant offset. Noise and phase residues, however, will decrease the reliability. The reliability of a junction is defined as the product of the reliabilities of the two tiles belonging to it. The process of grouping and unwrapping tiles with respect to each other is analogous to the pixel-based SRNCP and detailed in Algorithm 3.

It is by the definition of the reliability that our tile-based SRNCP merger substantially differs from the pixel-based SRNCP algorithm. Our definition of reliability is meaningful only for a tile-based approach and does not converge to the pixel-based method in the limit of single pixel tiles. Whenever we speak of the pixel-based SRNCP algorithm in this paper we refer to the original implementation of Herráez *et al.*, while the tile-based τSRNCP merger refers to our adaption of a merging algorithm.

```
Algorithm 3 Tile-based $\tau$SRNCP merger
    CALCULATION OF JUNCTION RELIABILITIES
$\mathcal{L} \leftarrow$ List of all junctions in the tessellated image.
calculate reliability for each junction in $\mathcal{L}$.
  GROUPING AND MERGING TILES
set each tile as belonging to no group.
sort $\mathcal{L}$ in descending order of junction reliability.
for all junctions $\mathcal{J}_{AB} \in \mathcal{L}$ do
  if neither $\tau_A$ nor $\tau_B$ belong to a group then
    MERGETILES ($\tau_A$, $\tau_B$) and generate a new group containing both tiles.
  else if only $\tau_B$ has a group then
    MERGETILES ($\tau_A$, $\tau_B$) and add $\tau_A$ to group of $\tau_B$.
  else if only $\tau_A$ has a group then
    MERGETILES ($\tau_B$, $\tau_A$) and add $\tau_B$ to group $G_A$.
  else          ▷ Both tiles belong to group
    Merge all tiles of smaller group to the tile $\tau_{A, B}$ with the larger group.
    Assimilate the smaller group in the larger group.
```

Methods

To assess the quality of the proposed method for synthetic and measured datasets, we have evaluated different combinations of tile unwrappers and mergers. We have further analyzed the performance of the pixel-based SRNCP algorithm. For model-based tile unwrapping we chose monomial base functions, so the resulting model function *f* on every tile can be written as

$$f(x,y) = \rho + \sum_{i=0,\ldots,P_x} \sum_{j=0,\ldots,P_y} c_{ij} \cdot x^i y^j$$

where P_x and P_y are the maximum degrees of powers in *x* and *y* respectively. We set $P_x = P_y = P$, resulting in a polynomial of degree $2P$ and in a number of $(P + 1)^2 - 1$ coefficients. The term $x^0\, y^0$ is absorbed into the constant offset ρ.

Application to synthetic phase maps with noise

A ground-truth synthetic phase map ϕ_u was generated on a quadratic raster with $N_X = N_Y = 600$ pixels. The top left pixel of this raster is designated as the origin (0, 0). The synthetic phase consists of two overlapping Gaussian profiles and is given analytically by

$$\phi_u(x,y) = 4\pi \cdot \exp\left(- \frac{\left(x - \frac{N}{3}\right)^2 + \left(y - \frac{N}{3}\right)^2}{2 \cdot (6N)^2} \right) + 8\pi \cdot \exp\left(- \frac{\left(x - \frac{2N}{3}\right)^2 + \left(y - \frac{2N}{3}\right)^2}{2 \cdot (8N)^2} \right).$$

While the particular shape of the profile was chosen arbitrarily, Gaussian phase profiles are often used as synthetic phase distributions [16, 44, 45]. Furthermore the Gaussian phase profile cannot simply be fitted by a set of monomial base functions. We obtained a noisy phase distribution $\phi_u^\sigma(x,y) = \phi_u(x,y) + \mathcal{N}_\sigma(x,y)$ by adding a zero mean white Gaussian noise distribution $\mathcal{N}_\sigma(x,y)$ with standard deviation σ. Noisy wrapped phase distributions were generated using $\phi_w^\sigma = \mathcal{W}\phi_u^\sigma$ for $\sigma \in [0, 1]$ in steps of 0.1. Wrapped phase maps were processed with different combinations of tile-based unwrapping and merging algorithms as well as the pixel-based SRNCP algorithm.

To quantify the fidelity of the result obtained by an algorithm ϕ_u^{alg} to the ground truth phase ϕ_u^σ, we calculated the *deviation D* of the ground truth and unwrapped phase maps

$$D = \frac{1}{N_X N_Y} \sqrt{\sum_{x,y} \left| \phi_u^{alg}(x,y) - \phi_u^\sigma(x,y) - \mu \right|^2}$$

with $\mu = \langle \phi_u^{alg} - \phi_u^\sigma \rangle$ to compensate for a possible constant offset between these phase maps. The mean deviation of 25 distributions for each Gaussian noise level σ was calculated and plotted as a function of the noise level σ.

Adding zero mean white Gaussian noise to phase images is a common practice to quantify the performance of phase unwrapping algorithms [7, 16, 29, 46]. It must be noted, however, that zero mean white Gaussian noise is not a valid model for many noise sources in real world applications [29, 47, 48]. Yet, the results give an accurate indication of the robustness of the algorithm against violations of the Itoh sampling condition due e.g. to low signal to noise ratio of measured datasets [49].

Application to a synthetic phase map with noise and residues

The performance of an unwrapping algorithm in the presence of residues is an important issue in many applications [21, 22, 50]. To qualitatively gauge the performance of the tile-based approach presented here in the presence of phase residues, we generated a synthetic phase map as follows: First we created a noiseless ground truth distribution as detailed in the previous section. Next, we placed a rectangle over a phase wrap line and applied a Gaussian blur filter with a width of 5 pixels. This creates a smooth border and removes the phase jump within this area. Finally, we added a zero mean Gaussian noise distribution with $\sigma = 0.3$ to the entire phase map and wrapped the resulting image to obtain a noisy wrapped phase with residues. Since there is no unique solution to unwrapping any wrapped phase distribution with residues, we compared the performance of several tile-based algorithms and the pixel-based SRNCP unwrapper by qualitatively assessing the influence of the phase residue on the result.

Application to measured digital holographic datasets. The proposed algorithm was also applied to real world wrapped phase distributions that were obtained with a digital holography setup as reported by Kalies and Antonopoulos *et al.* [51]. Trypsinized ZMTH3 canine adenoma cells, kindly provided by Murua Escobar *et al.*, served as a sample [52]. The wrapped phase map has a dimension of 1400 × 1400 pixels.

Results and Discussion

Synthetic phase with noise

We analyzed the deviation of the unwrapped phase from the noisy ground truth as a function of different noise levels σ for various algorithms (see Fig 2 for the data and Fig 3 for exemplary images). Deviations were calculated by averaging over 25 phase maps for each noise level. We compared different combinations of tile-based unwrapping and merging algorithms with the pixel-based SRNCP unwrapper. We also reproduced Strand's original unwrapping algorithm by combining Strand's tile unwrapper and the unidirectional merger. For Strand's algorithm we used a tessellation into $N_\tau = 40 \times 40$ tiles. Our proposed algorithm was tested by tessellating the phase into $N_\tau = 20 \times 20$ tiles and applying our model-based tile unwrapper with polynomial order $P = 2$. For merging we used our τSRNCP merger. Up to a noise level of $\sigma = 0.3$, these three phase unwrapping algorithms yield very good results with average deviations of the order of 10^{-5}. The mean deviation rises to the order of 10^{-2} up to a noise level of $\sigma = 0.6$. For higher noise levels the deviations rise to 0.4 and 0.49 at $\sigma = 1.0$ for Strand's algorithm and the proposed algorithm respectively. In contrast, a major rise of the deviation (7.5 at $\sigma = 1.0$) is observable in the case of the pixel-based SRNCP merger. This is caused by propagation of unwrapping failures of single pixels to large areas. For the tile-based approaches, erroneous unwraps remain within tiles and are much less likely to be propagated to further tiles in the merging step. This is the reason for the robustness of the tile-based approach to high frequency noise. Note that a tile-based τSRNCP merger also inherits this robustness while still operating on the reliability guided merging principle of the original algorithm.

We explored the influence of different merging algorithms on the result by testing it in a case where the unwrap of several tiles fails. For this, tiles were unwrapped with Strand's algorithm using a tessellation into $N_\tau = 20 \times 20$ tiles. Some tiles contained more than one phase wrap, so that they could not be correctly unwrapped even in the noise-free case. For noise levels below $\sigma = 0.8$, the τSRNCP produced lower deviation from the ground truth than the unidirectional merger. This is due to a tendency of the unidirectional merger to propagate unwrap failures to subsequent tiles, see S1 Fig. The τSRNCP merger is much less likely to propagate errors to large areas since it processes reliable junctions first. This demonstrates, that employing a

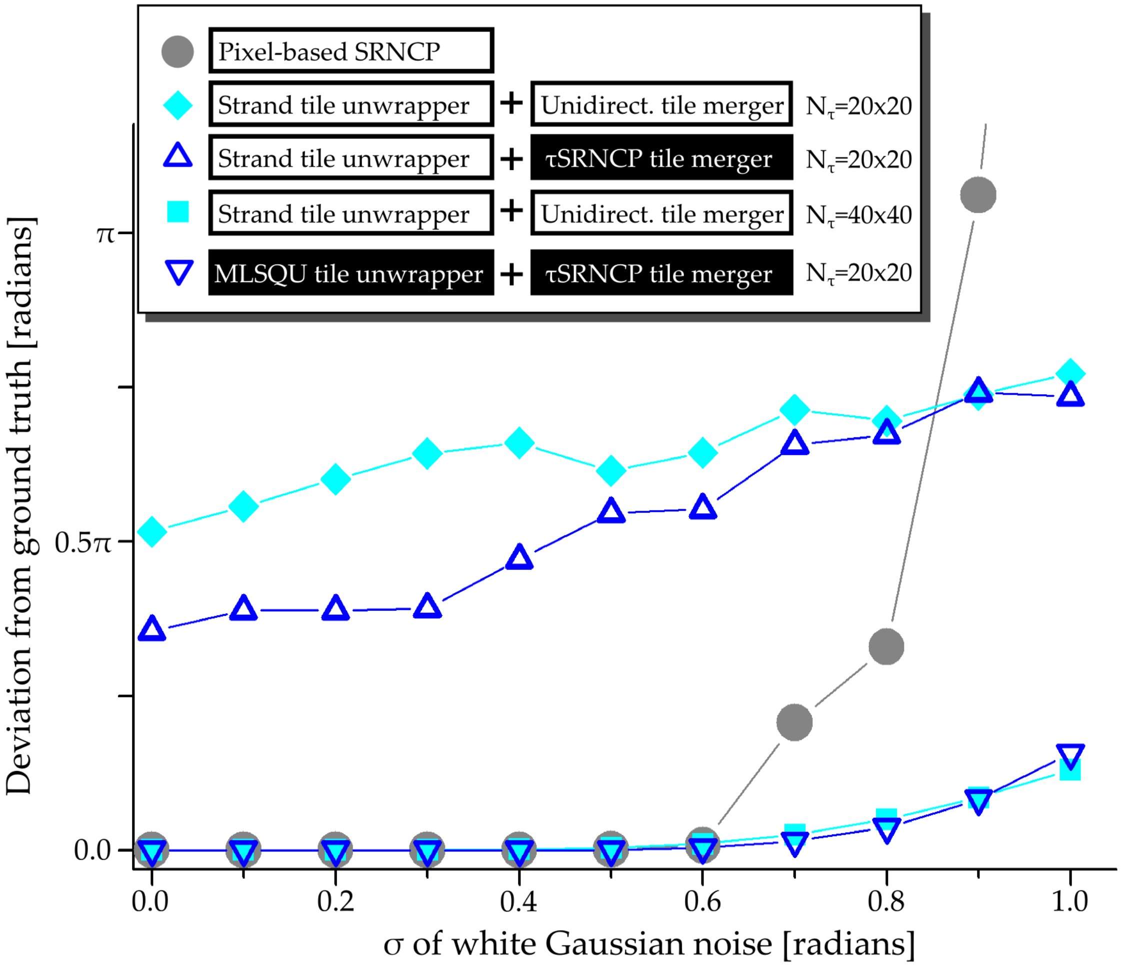

Fig 2. Deviation of the unwrapped phase from a noisy ground truth for different algorithms. The mean deviation of 25 phase images to the ground truth with a white Gaussian noise of specified σ is shown. (▽) The proposed algorithm using a model-based tile unwrapper with polynomial base $P = 2$ and $N_\tau = 20 \times 20$ tiles and $N_\rho = 40$ steps for minimization. Merging was performed using our τSRNCP merger. (•) The pixel-based SRNCP algorithm. At $\sigma = 1.0$ the deviation is outside the range of the plot with a value of 7.7 radians. (■) Strand's original algorithm using a tessellation into $N_\tau = 40 \times 40$ tiles and $N_\rho = 20$. The plots for (✦) and (△) show the results of using Strand's unwrapper with a tile size containing more than one phase wrap using $N_\rho = 40$. In this case, the unwrapping of those tile fails and the deviation is analyzed for two different merging algorithms: (✦) the unidirectional merger and (△) the τSRNCP merger. Algorithms proposed in this paper are in black boxes.

doi:10.1371/journal.pone.0143186.g002

quality guided merger can reduce the influence of failed tile unwrapping on the result compared to a linear approach. For reference, runtimes of the algorithms used in this section are given in Table 1. Due to the computational overhead introduced by the framework and the complexity of linear algebra operations, algorithms involving the MLSQU tile unwrapper have longer execution times compared to simpler algorithms. It has to be noted, however, all unwrapping processes were executed sequentially on a single core, since no parallelization has been implemented, yet.

Synthetic phase with noise and residues

Results of unwrapping a synthetic phase map with noise and residues are shown in Fig 4. We unwrapped the phase map with the pixel-based SRNCP algorithm (Fig 4B) and Strand's original unwrapping and merging algorithm (Fig 4C). For Strand's algorithm we used a tile count of $N_\tau = 40 \times 40$ tiles in x- and y-directions. Both algorithms propagate unwrapping errors over

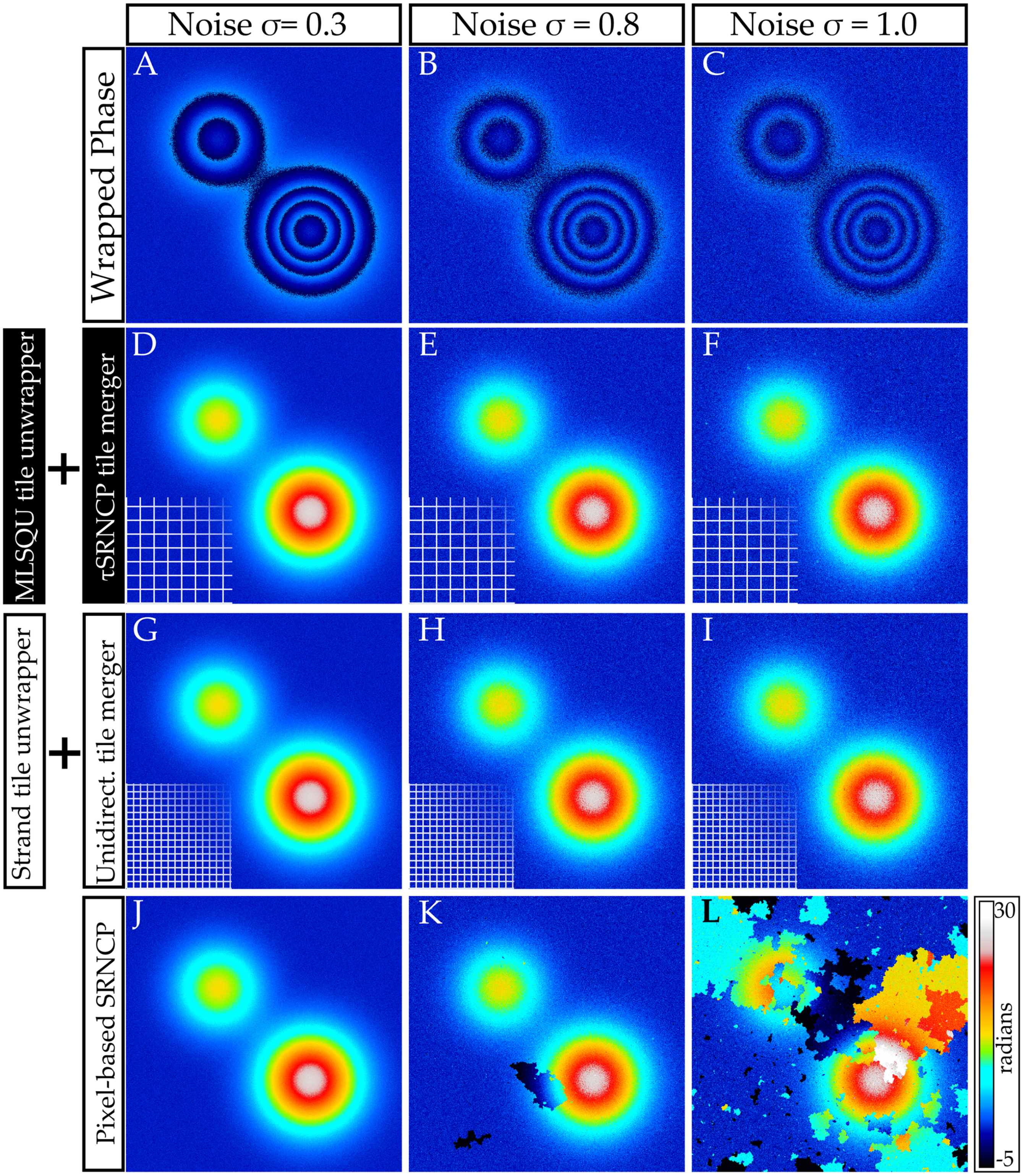

Fig 3. Comparison of phase unwrap with different algorithms for various noise levels. A,B,C: Wrapped phase distributions for different noise levels. **D, E,F:** Tile-based unwrap with a tessellation into 20 × 20 tiles. Tile unwrapping was performed with the model-based least squares unwrapper (MLSQU) using $P = 2$, $N_\rho = 40$. Merging was performed with the tile-based τSRNCP merger. **G,H,I:** Tile-based unwrap with a tessellation into 40 × 40 tiles. Tile unwrapping was done with Strand's unwrapper ($N_r ho = 20$) and a unidirectional merger. This corresponds to Strand's original algorithm. **J,K,L:** Unwrapped phase map using the pixel-based SRNCP algorithm. For a full resolution graphic see S1 Fig. Algorithms proposed in this paper are in black boxes.

doi:10.1371/journal.pone.0143186.g003

Table 1. Runtimes of phase unwrapping algorithms. For tile-based algorithms the total runtime is the sum of the runtimes of tile unwrapper and tile merger. For the model-based tile unwrapper (MSLQU) the precalculation of the system matrix has to be performed once and is given in braces. A synthetic noisy distribution with $\sigma = 0.6$ and dimensions of 600×600 pixels was unwrapped in all cases. Runtimes were measured on a high-end desktop PC (Intel® i5-4570 3.2GHz CPU, 32 GB RAM) with sequential execution on a single core. Runtimes are given as the mean of five independent measurements.

Tesselation N_τ	Algorithm	Runtime
40 × 40	Strand tile unwrapper ($N_\rho = 40$)	453.0 ms
	Unidirectional tile Merger	3.0 ms
20 × 20	Strand tile unwrapper ($N_\rho = 20$)	236.4 ms
	Unidirectional tile merger	1.0 ms
20 × 20	Strand tile unwrapper ($N_\rho = 20$)	235.0 ms
	τSRNCP tile merger	3.4 ms
20 × 20	MLSQU tile unwrapper ($N_\rho = 20$, $P = 2$)	578.8 ms (+1.6 ms)
	τSRNCP tile merger	4.0 ms
2 × 2	MLSQU tile unwrapper ($N_\rho = 60$, $P = 2$)	1342 ms (+1533 ms)
	τSRNCP tile merger	< 1 ms
none	Pixel-based SRNCP unwrapper	155.4 ms

doi:10.1371/journal.pone.0143186.t001

approximately a third and a quarter of the image, respectively. In the synthetic noisy case without residues, both algorithms were shown to yield very good results for up to double the noise level. Next, we compared the results of combining the proposed model-based tile unwrapper with Strand's unidirectional merger as well as our proposed τSRNCP merger. A polynomial model with $P = 2$ was used and the tile count set to $N_\tau = 20 \times 20$, thereby doubling the area of each tile compared to Fig 4C. Merging the tiles with the unidirectional merger propagates unwrapping errors due to the phase residue to a number of neighboring tiles, see Fig 4D. When the tiles are merged with the τSRNCP merger, the unwrapped phase is affected only on the four tiles containing the residue. Since all four tiles are affected by the phase residue at one or more borders, their reliability is decreased compared to tiles with no residues. This ensures that they will be processed last in the merging process. Thus, unwrapping errors are not propagated to more tiles. The affected area corresponds to 0.25% of the total area of the phase map. This shows, that combining model-based tile unwrapping with a reliability guided merging process can prevent the propagation of unwrapping errors in an unwrapped phase map in the case where residues are present.

Finally, we tessellated the image into merely 2 × 2 tiles and unwrapped the tiles with the MLSQU tile unwrapper using a polynomial with $P = 6$, see Fig 4F. The residue is located within the top left tile and after unwrapping the residue has minimal impact on the tile. Merging is performed using the τSRNCP merger, but using the unidirectional merger will produce the same result since the residue does not affect the border of the tile. Thus, by choosing the size of the tiles large enough that residues are located within the tiles, the final unwrapped phase will only be affected on pixels close to the residue. This demonstrates the value of our model-based tile unwrapper compared to Strand's method: the proposed model-based tile unwrapping algorithm is capable of processing tiles with more than just one phase wrap inside.

In summary, we have shown three key points on synthetic datasets: First, a tile-based approach increases the robustness of the reliability guided SRNCP unwrapping algorithm against high frequency noise. Second, the τSRNCP merging algorithm can significantly increase the quality of an unwrapped phase map when tile unwrapping fails due to noise or residues. Finally, the model-based tile unwrapping algorithm proposed in here allows for increased tile size compared to Strand's approach. The unwrapper performs well in the presence of noise and

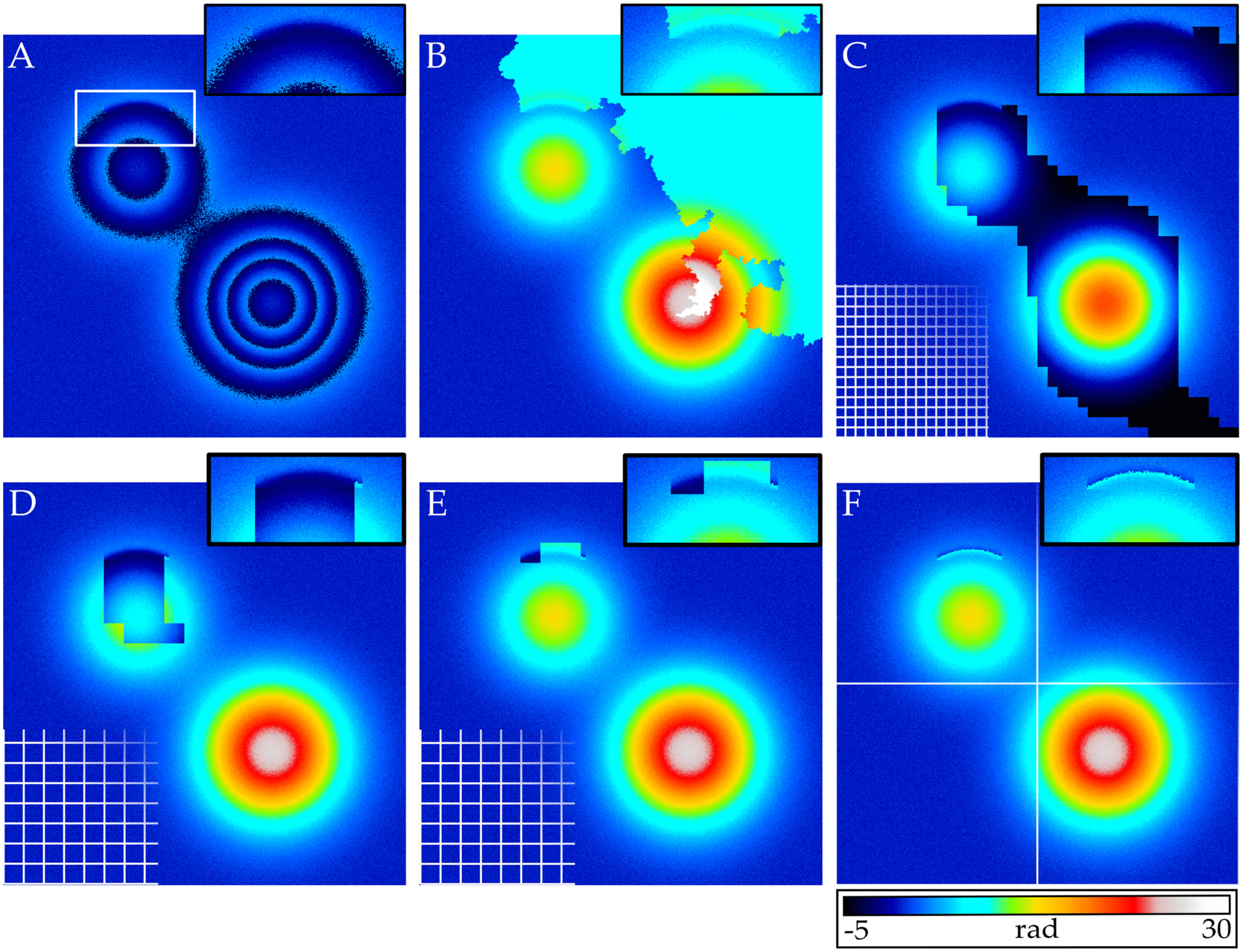

Fig 4. Unwrap of a synthetic phase map with noise and residues. A: Wrapped phase distribution with a noise level of $\sigma = 0.3$. The residue is indicated by an arrow and the a close-up of the surrounding area is shown in an inset. **B:** Pixel-based SRNCP unwrap. **C:** Tilebased unwrap with $N_\tau = 40 \times 40$ tiles using Strand's tile unwrapper ($N_\rho = 20$) and Strand's unidirectional merger. **D:** Unwrapped phase using the proposed MLSQU tile unwrapper with $P = 2$, $N_\rho = 40$ and the unidirectional tile merger. Tile count is $N_\tau = 20 \times 20$, resulting in double the tile size compared to C. **E:** Same as D, but using the proposed tile-based τSRNCP algorithm for merging. **F:** Unwrapped phase using MLSQU tile unwrapper with $P = 6$, $N_\rho = 60$ and the tile-based τSRNCP merger. The image was tessellated into a 2 × 2 grid. All images have dimensions of 600 × 600 pixels and the same color scale as Fig 2.

doi:10.1371/journal.pone.0143186.g004

can be beneficial in the presence of phase residues. On a more general level, we believe that this validates our approach of creating a modular tile-based phase unwrapping concept that works with different combinations of unwrapping and merging algorithms.

Digital holographic phase measurements

We applied our phase unwrapping algorithm to a wrapped phase image of ZMTH3 canine adenoma cells obtained with digital holography, see Fig 5. The wrapped phase map contains multiple residues but it is free of high frequency noise. We compared the phase unwrap of the pixel-based SRNCP algorithm with our tile-based approach using a model-based unwrapper ($P = 2$,

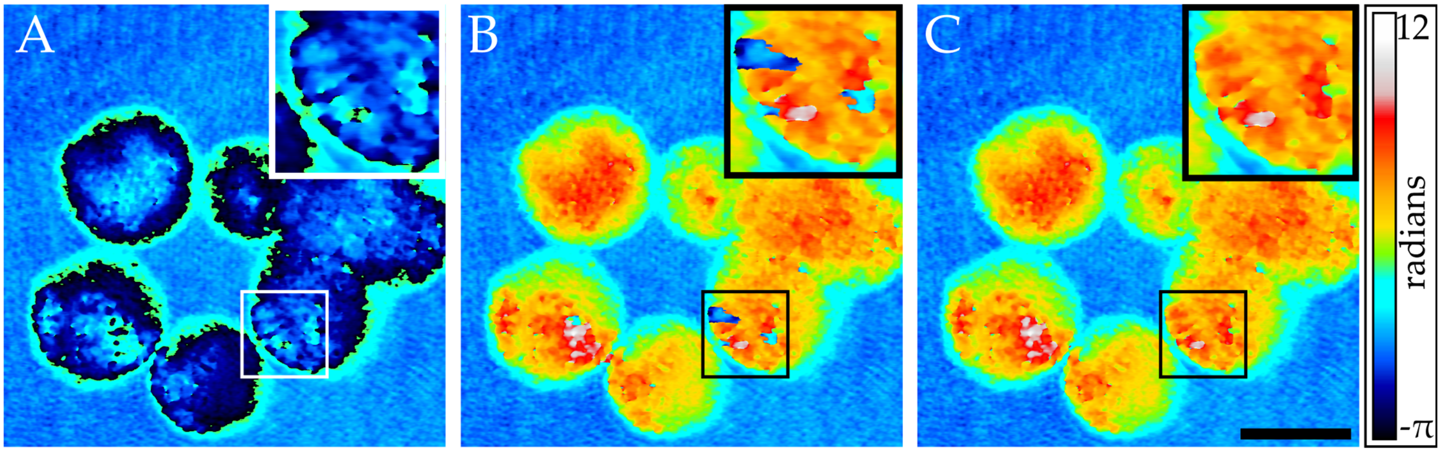

Fig 5. Phase unwrap of ZMTH3 canine adenoma cells captured with digital holography. A: Wrapped phase map. **B:** Pixel-based unwrap with the SRNCP algorithm. **C:** Unwrap using the proposed approach by tessellating the image into 8 × 8 tiles and unwrapping individual tiles with the model-based approach using polynomial order $P = 4$. Tiles were merged with the unidirectional merger. A magnification of a region with phase residues is shown. The phase maps consist of 1400 × 1400 pixels. Scale bar is 15 *μm*.

doi:10.1371/journal.pone.0143186.g005

$N_p = 40$) and a unidirectional merger. Both algorithms appear to be successful in unwrapping the given phase distribution and do not produce obvious inaccuracies. Upon closer inspection, however, the pixel-based unwrap shows several areas in which unwrapping errors due to phase residues were propagated to larger regions. Our tile-based approach performs more robustly in the presence of residues in accordance with the results from the previous section.

Conclusion

We have presented a modular tile-based phase unwrapping approach that was cast into a C++11 program with a modular framework. In a two step process, individual tiles are unwrapped first and then merged to an unwrapped phase map using a merging algorithm. A set of tile unwrapping and merging algorithms was implemented as suggested by Strand *et al.* We developed a model-based unwrapper that contains Strand's unwrapper as a special case. Both algorithms performed well in synthetic high noise scenarios. We could further show that our model-based approach is superior to Strand's original unwrapping algorithm in scenarios with phase residues due to its ability to process larger tiles. This was demonstrated for a synthetic wrapped phase map as well as a real world phase map obtained from digital holography of canine adenoma cells. Model-based tile unwrapping was formulated as a least squares minimization using linear algebra. As a next step this could allow use of parallelization to speed up the unwrapping process significantly, since linear algebra lends itself nicely to parallel computing. In this study we did not implement parallelization and can therefore not take full advantage of this approach, yet.

We could further create a reliability guided tile merger from the pixel-based SRNCP algorithm. This so called τSRNCP merger performed superior compared to a linear merger in scenarios with erroneous unwraps as well as phase residues. It also performed superior to the pixel-based SRNCP algorithm in the presence of noise and residues. We believe that this validates the idea of the modular tile-based phase unwrapping concept behind our software framework. It also indicates that adapting well-established pixel-based algorithms to a tile-based approach is possible and worth-while. The computational load on these merging algorithms is significantly decreased to their pixel-based counterparts, because they operate on the tiles of the tessellated image instead of single pixels. This fact could make the implementation of more complex algorithms, such as optimization based on simulated annealing, computationally feasible for large phase maps.

Typically, the choice of phase unwrapping algorithm is—to an extent—dependent on the properties of wrapped datasets themselves, e.g. the level of noise. Furthermore there might be user requirements that favor a fast algorithm over one with high robustness against noise. It is obviously false, that a slower, more complex algorithm will generally result in better unwrapped phase maps. However, there is a trade-off between speed and robustness. We feel that the proposed framework offers an advantage that might make it interesting for some applications: The modular framework makes it possible to tailor unwrapping algorithms to specific needs. It serves as an easily extensible toolbox that can be used to test and create new tile-based phase unwrapping algorithms before committing to an individual algorithm. It might further be possible to perform the tailoring process automatically. For that, a sufficiently large set of unwrapping and merging algorithms is needed. Then it is conceivable that the particular combination of unwrapper and merger can be constructed automatically using machine learning with appropriate training sets. One example of a specially tailored algorithm is a combination of the Strand tile unwrapper and the unidirectional merger using a parallel implementation. This could be fast enough for live imaging, if a reduced image quality is acceptable. Another example stems from the properties of the MLSQU tile unwrapping algorithm: Here the precalculation of the system matrix makes it possible to apply high order polynomial fits to large tiles. As a consequence, high numbers of images can be processed efficiently. While the computational load is likely still rather high compared to other algorithms, a possible gain in image quality could make up for this. By making our software publicly available we hope that other researchers may find it useful to use and improve upon it.

Supporting Information

S1 Fig. Full resolution version of Fig 3. Additional rows show results for tessellation of the original image into 20 × 20 and using Strand's unwrapper ($N_\rho = 40$) for individual tiles. In this case, the unwrap of all tiles that contain more than one phase wraps fails. Merging is performed using two different algorithms: **M,N,O:** Unidirectional Merger and **P,Q,R:** tile-based τSRNCP merger. Algorithms proposed in this paper are in black boxes.
(TIF)

S2 Fig. Schematic illustration of phase residues. A Wrapped phase image of a silica microsphere in water. At the location of the arrows a so-called fringe washout is visible. This smoothes the phase, such that in a wrapped phase no phase jumps will be detectable. **B** Schematic of the isolines of 2π phase jumps between adjacent pixels. At the position of the fringe washout the isolines are discontinued. These open isolines are errors of the measurement and do not represent physical reality. For an error-free wrapped phase map of a physical phase, all isolines inside the area of measurement must be closed. The tips of open isolines are so called residues. A closed loop (C_1) around such a residue will count a different number of positive and negative phase jumps. In contrast, any closed loop (C_2) around a closed isoline will count the same number of positive and negative phase jumps. This illustration given here is rather intuitive and it suffices for the purposes of this study. For a more technical definition refer to [19].
(TIF)

Acknowledgments

We would like to thank Raoul Lorbeer, Marko Heidrich, Stefan Kalies, and Margaux Schmeltz for vivid discussions on the topic of phase unwrapping as well as helpful comments on the manuscript.

Author Contributions

Conceived and designed the experiments: GCA BS. Performed the experiments: GCA BS. Analyzed the data: GCA BS. Contributed reagents/materials/analysis tools: AH TR HM. Wrote the paper: GCA BS AH TR HM. Conceived and designed the framework and original algorithms: GCA. Wrote the C++ implementation: BS GCA. Wrote additional ImageJ Plugins: GCA.

References

1. Schnars U, Jüptner W. Direct recording of holograms by a CCD target and numerical reconstruction. Applied optics. 1994; 33(2):179–181. doi: 10.1364/AO.33.000179 PMID: 20862006
2. Graham LC. Synthetic interferometer radar for topographic mapping. Proceedings of the IEEE. 1974; 62(6):763–768. doi: 10.1109/PROC.1974.9516
3. Chavez S, Xiang QS, An L. Understanding phase maps in MRI: a new cutline phase unwrapping method. Medical Imaging, IEEE Transactions on. 2002; 21(8):966–977. doi: 10.1109/TMI.2002.803106
4. Chen CW, Zebker HA. Network approaches to two-dimensional phase unwrapping: intractability and two new algorithms. J Opt Soc Am A. 2000 Mar; 17(3):401–414. Available from: http://josaa.osa.org/abstract.cfm?URI=josaa-17-3-401. doi: 10.1364/JOSAA.17.000401
5. Tribolet J. A new phase unwrapping algorithm. IEEE Transactions on Acoustics Speech and Signal Processing. 1977; 25(2):170–177. doi: 10.1109/TASSP.1977.1162923
6. Itoh K. Analysis of the phase unwrapping algorithm. Appl Opt. 1982 Jul; 21(14):2470. doi: 10.1364/AO.21.002470 PMID: 20396058
7. Saika Y, Uezu T. Bayesian Inference for Phase Unwrapping using Multiple Interferograms Based on Statistical Mechanics of Three-state Ising Model. Recent Advances in Electrical and Computer Engineering Bayesian. 2014;p. 23–28.
8. Epple FM, Ehn S, Thibault P, Koehler T, Potdevin G, Herzen J, et al. Phase unwrapping in spectral x-ray differential phase-contrast imaging with an energy-resolving photon-counting pixel detector. 2014;.
9. Robinson S, Schödl H, Trattnig S. A method for unwrapping highly wrapped multi-echo phase images at very high field: UMPIRE. Magn Reson Med. 2014 Jul; 72(1):80–92. doi: 10.1002/mrm.24897 PMID: 23901001
10. Herráez MA, Burton DR, Lalor MJ, Gdeisat MA. Fast two-dimensional phase-unwrapping algorithm based on sorting by reliability following a noncontinuous path. Applied Optics. 2002; 41(35):7437–7444. doi: 10.1364/AO.41.007437 PMID: 12502301
11. Ghiglia DC, Pritt MD. Two-dimensional phase unwrapping: theory, algorithms, and software. Wiley New York:; 1998.
12. Ghiglia DC, Romero LA. Robust two-dimensional weighted and unweighted phase unwrapping that uses fast transforms and iterative methods. JOSA A. 1994; 11(1):107–117. doi: 10.1364/JOSAA.11.000107
13. Pritt MD, Shipman JS. Least-squares two-dimensional phase unwrapping using FFT's. Geoscience and Remote Sensing, IEEE Transactions on. 1994; 32(3):706–708. doi: 10.1109/36.297989
14. Schofield MA, Zhu Y. Fast phase unwrapping algorithm for interferometric applications. Optics letters. 2003; 28(14):1194–1196. doi: 10.1364/OL.28.001194 PMID: 12885018
15. Marroquin JL, Tapia M, Rodriguez-Vera R, Servin M. Parallel algorithms for phase unwrapping based on Markov random field models. JOSA A. 1995; 12(12):2578–2585. doi: 10.1364/JOSAA.12.002578
16. Dias JM, Leitão JM. The Z πM algorithm: a method for interferometric image reconstruction in SAR/SAS. Image Processing, IEEE Transactions on. 2002; 11(4):408–422. doi: 10.1109/TIP.2002.999675
17. Flynn TJ. Two-dimensional phase unwrapping with minimum weighted discontinuity. JOSA A. 1997; 14(10):2692–2701. doi: 10.1364/JOSAA.14.002692
18. Costantini M. A novel phase unwrapping method based on network programming. Geoscience and Remote Sensing, IEEE Transactions on. 1998; 36(3):813–821. doi: 10.1109/36.673674
19. Goldstein RM, Zebker HA, Werner CL. Satellite radar interferometry: Two-dimensional phase unwrapping. Radio science. 1988; 23(4):713–720. doi: 10.1029/RS023i004p00713
20. Cusack R, Huntley J, Goldrein H. Improved noise-immune phase-unwrapping algorithm. Applied Optics. 1995; 34(5):781–789. doi: 10.1364/AO.34.000781 PMID: 21037595
21. Gutmann B, Weber H. Phase unwrapping with the branch-cut method: role of phase-field direction. Applied optics. 2000; 39(26):4802–4816. doi: 10.1364/AO.39.004802 PMID: 18350072

22. Ambale B, Denney TS, Gupta H, Lloyd S, Dell'Italia L. 3D left ventricular strain by phase unwrapping: a simulated annealing based branch-cut placement method. In: Biomedical Imaging: From Nano to Macro, 2009. ISBI'09. IEEE International Symposium on. IEEE; 2009. p. 466–469.

23. Quiroga J, Bernabeu E. Phase-unwrapping algorithm for noisy phase-map processing. Applied optics. 1994; 33(29):6725–6731. doi: 10.1364/AO.33.006725 PMID: 20941216

24. Fornaro G, Sansosti E. A two-dimensional region growing least squares phase unwrapping algorithm for interferometric SAR processing. Geoscience and Remote Sensing, IEEE Transactions on. 1999; 37 (5):2215–2226. doi: 10.1109/36.789618

25. Baldi A. Two-dimensional phase unwrapping by quad-tree decomposition. Appl Opt. 2001 Mar; 40 (8):1187–1194. Available from: http://ao.osa.org/abstract.cfm?URI=ao-40-8-1187. doi: 10.1364/AO.40.001187 PMID: 18357104

26. Herráez MA, Gdeisat MA, Burton DR, Lalor MJ. Robust, fast, and effective two-dimensional automatic phase unwrapping algorithm based on image decomposition. Applied Optics. 2002; 41(35):7445–7455. doi: 10.1364/AO.41.007445 PMID: 12502302

27. Towers D, Judge T, Bryanston-Cross P. A quasi heterodyne holographic technique and automatic algorithms for phase unwrapping. In: 33rd Annual Techincal Symposium. International Society for Optics and Photonics; 1989. p. 95–119.

28. Stephenson P, Burton DR, Lalor MJ. Data validation techniques in a tiled phase unwrapping algorithm. Optical Engineering. 1994; 33(11):3703–3708. doi: 10.1117/12.181183

29. Strand J, Taxt T, Jain AK. Two-dimensional phase unwrapping using a block least-squares method. Image Processing, IEEE Transactions on. 1999; 8(3):375–386. doi: 10.1109/83.748892

30. Zhong H, Tang J, Liu D. A fast phase unwrapping algorithm based on minimum discontinuity by blocking. In: Future Computer and Communication (ICFCC), 2010 2nd International Conference on. vol. 1. IEEE; 2010. p. V1–717.

31. Egidi N, Maponi P. A comparative study of two fast phase unwrapping algorithms. Applied mathematics and computation. 2004; 148(3):599–629. doi: 10.1016/S0096-3003(02)00920-7

32. Kemper B, Vollmer A, Rommel CE, Schnekenburger J, von Bally G. Simplified approach for quantitative digital holographic phase contrast imaging of living cells. Journal of biomedical optics. 2011; 16 (2):026014–026014. doi: 10.1117/1.3540674 PMID: 21361698

33. Pham H, Ding H, Sobh N, Do M, Patel S, Popescu G. Off-axis quantitative phase imaging processing using CUDA: toward real-time applications. Biomedical optics express. 2011; 2(7):1781–1793. doi: 10.1364/BOE.2.001781 PMID: 21750757

34. Girshovitz P, Shaked NT. Compact and portable low-coherence interferometer with off-axis geometry for quantitative phase microscopy and nanoscopy. Optics express. 2013; 21(5):5701–5714. doi: 10.1364/OE.21.005701 PMID: 23482143

35. Coppola G, Di Caprio G, Gioffré M, Puglisi R, Balduzzi D, Galli A, et al. Digital self-referencing quantitative phase microscopy by wavefront folding in holographic image reconstruction. Optics letters. 2010; 35(20):3390–3392. doi: 10.1364/OL.35.003390 PMID: 20967076

36. Memmolo P, Iannone M, Ventre M, Netti PA, Finizio A, Paturzo M, et al. On the holographic 3D tracking of in vitro cells characterized by a highly-morphological change. Optics express. 2012; 20(27):28485–28493. doi: 10.1364/OE.20.028485 PMID: 23263084

37. Ferraro P, De Nicola S, Finizio A, Pierattini G, Coppola G. Recovering image resolution in reconstructing digital off-axis holograms by Fresnel-transform method. Applied physics letters. 2004; 85 (14):2709–2711. doi: 10.1063/1.1791735

38. Ghiglia DC, Mastin GA, Romero LA. Cellular-automata method for phase unwrapping. JOSA A. 1987; 4 (1):267–280. doi: 10.1364/JOSAA.4.000267

39. Liang ZP. A model-based method for phase unwrapping. IEEE transactions on medical imaging. 1995; 15(6):893–897. doi: 10.1109/42.544507

40. Friedlander B, Francos JM. Model based phase unwrapping of 2-D signals. Signal Processing, IEEE Transactions on. 1996; 44(12):2999–3007. doi: 10.1109/78.553474

41. Langley J, Zhao Q. A model-based 3D phase unwrapping algorithm using Gegenbauer polynomials. Physics in medicine and biology. 2009; 54(17):5237. doi: 10.1088/0031-9155/54/17/011 PMID: 19671967

42. Flannery BP, Press WH, Teukolsky SA, Vetterling W. Numerical recipes in C. Press Syndicate of the University of Cambridge, New York. 1992;24.

43. Guennebaud G, Jacob B, et al. Eigen v3; 2010. http://eigen.tuxfamily.org.

44. Stramaglia S, Nico G, Pasquariello G, Guerriero L. Phase-unwrapping method based on stochastic relaxation. Proc SPIE. 1997; 3217:4–12. Available from: http://dx.doi.org/10.1117/12.295591. doi: 10.1117/12.295591

45. Adi K, Suksmono AB, Mengko TLR, Gunawan H. Phase unwrapping by Markov chain Monte Carlo energy minimization. Geoscience and Remote Sensing Letters, IEEE. 2010; 7(4):704–707. doi: 10.1109/LGRS.2010.2046393

46. Memmolo P, Esnaola I, Finizio A, Paturzo M, Ferraro P, Tulino AM. SPADEDH: a sparsity-based denoising method of digital holograms without knowing the noise statistics. Opt Express. 2012 Jul; 20 (15):17250–17257. Available from: http://www.opticsexpress.org/abstract.cfm?URI=oe-20-15-17250. doi: 10.1364/OE.20.017250

47. Zebker HA, Villasenor J. Decorrelation in interferometric radar echoes. Geoscience and Remote Sensing, IEEE Transactions on. 1992 Sep; 30(5):950–959. doi: 10.1109/36.175330

48. Gudbjartsson H, Patz S. The Rician distribution of noisy MRI data. Magnetic resonance in medicine. 1995; 34(6):910–914. doi: 10.1002/mrm.1910340618 PMID: 8598820

49. Goldstein GL. Smart Temporal Phase Unwrapping for Biological Objects. The University of Arizona; 2013. Available from: http://hdl.handle.net/10150/311573.

50. Li C, Zhu DY. A residue-pairing algorithm for insar phase unwrapping. Progress In Electromagnetics Research. 2009; 95:341–354. doi: 10.2528/PIER09070706

51. Kalies S, Antonopoulos GC, Rakoski MS, Heinemann D, Schomaker M, Ripken T, et al. Investigation of Biophysical Mechanisms in Gold Nanoparticle Mediated Laser Manipulation of Cells Using a Multimodal Holographic and Fluorescence Imaging Setup. PLOS ONE. 2015 Apr; 10(4):e0124052. Available from: http://dx.doi.org/10.1371/journal.pone.0124052. doi: 10.1371/journal.pone.0124052 PMID: 25909631

52. Escobar HM, Meyer B, Richter A, Becker K, Flohr A, Bullerdiek J, et al. Molecular characterization of the canine HMGB1. Cytogenetic and genome research. 2003; 101(1):33–38. doi: 10.1159/000073415