ABSTRACT

**QUANTITATIVE DYNAMIC CELLULAR IMAGING BASED ON 3D
UNWRAPPED OPTICAL COHERENCE PHASE MICROSCOPY**

by
ArunKumar Gunasekar

Phase wrapping artifacts are frequently encountered in phase resolved imaging and sensing techniques based on interferometry. When the value of the actual phase (ϕ) ranges beyond $(-\pi, \pi]$, the extracted phase value (wrapped phase ψ) is artificially increased or decreased by a multiple of 2π.

In this study, we develop a 3D phase unwrapping method that exploits the correlation of 3D phase data ($\phi(x,y,t)$) over the other dimension (t). We validate our 3D unwrapping method using simulated data, as well as experimental data obtained by optically computed phase microscopy (OCPM) recently developed in our lab. Our results show that the 3D phase unwrapping method achieves accurate and robust reconstruction of phase $\phi(x,y,t)$. The application of the 3D phase unwrapping method is not limited to OCPM. Instead, it is a generic approach that can be used for other phase imaging modalities.

QUANTITATIVE DYNAMIC CELLULAR IMAGING BASED ON 3D
UNWRAPPED OPTICAL COHERENCE PHASE MICROSCOPY

by
ArunKumar Gunasekar

A Thesis
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Electrical Engineering

Helen and John C. Hartmann Department of
Electrical and Computer Engineering

May 2022

**QUANTITATIVE DYNAMIC CELLULAR IMAGING BASED ON 3D
UNWRAPPED OPTICAL COHERENCE PHASE MICROSCOPY**

ArunKumar Gunasekar

Dr. Xuan Liu, Thesis Advisor   Date
Associate Professor of Electrical and Computer Engineering, NJIT

Dr. Dong-Kyun Ko, Committee Member   Date
Associate Professor of Electrical and Computer Engineering, NJIT

Dr. Yuanwei Zhang, Committee Member   Date
Assistant Professor of Chemistry and Environmental Science, NJIT

BIOGRAPHICAL SKETCH

Author: ArunKumar Gunasekar

Degree: Master of Science

Date: May 2022

Date of Birth:

Place of Birth:

Undergraduate and Graduate Education:

- Master of Science in Electrical Engineering,
  New Jersey Institute of Technology, Newark, NJ, 2022

- Bachelor of Science in Electronics and Communication Engineering,
  St. Joseph's Institute of Technology, Chennai, TN, India, 2020

Major: Electrical Engineering

DEDICATION

I am dedicating this thesis to four beloved people who have meant and continue to mean so much to me.

First and foremost, to my Parents. Both my father (Gunasekar Purushothaman) and mother (Selvi Gunasekar) are working hard to pay for my tuition. We as a family already have a larger dept to pay, which stopped me for thinking of taking Master of Science. But they encouraged me to take it, baring all the pain they have to overcome.

I would also like to dedicate this to my younger brother (Arjun Gunasekar). As I'm in United States, he is now taking care of all my responsibilities that I have in India.

Last but not least, to my fiancée (LAKSHMI SAHAJA) who motivates me to reach my goals, helps me to overcome struggles and spends time helping me in my studies even though she has her own work life that she balances with personal life productively. She is one of the most important person responsible for completing this thesis.

My heartful thanks to all who's believing in me, helping me and motivating me to successfully live my dreams and goals.

ACKNOWLEDGMENT

I would like to acknowledge and give my warmest thanks to my thesis advisor (Dr. Xuan Liu) who made this work possible. Her guidance and advice carried me through all the stages of writing my thesis. I would also like to thank my committee members, for your brilliant comments and suggestions. Thanks to NJIT.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF FIGURES
(Continued)

Figure   **Page**

CHAPTER 1

INTRODUCTION

1.1 Phase Wrapping and Unwrapping

Phase wrapping is one of the steps in phase extraction methods. When all the phase points are constrained to the range -π ≤ Phase Offset < π, it is called a wrapped phase. The phase value is increased or decreased by a multiple of 2π to put the phase value within +/- π of the Phase Offset value if the actual phase is outside this range. Various factors can cause a wrapped phase, one among them is natural waves.

The phase wrapping artifacts are frequently encountered in phase resolved imaging and sensing techniques based on interferometry. When the value of the actual phase (ϕ) ranges beyond (-π, π], the extracted phase value (wrapped phase ψ) is artificially increased or decreased by a multiple of 2π. To quantify the phase accurately, researchers developed various phase unwrapping algorithms.

The final step in phase extraction methods is phase unwrapping, where the correct phase is retrieved from the wrapped phase by eliminating 2π discontinuities. The path of phase unwrapping is determined according to the parameter map. This means that the pixel which has a higher parameter value in the parameter map will be phase unwrapped earlier. The goal of the phase unwrapping methods is to add a correct multiple of 2π to the wrapped phase such that a continuous unwrapped phase map is reconstructed. Phase unwrapping is needed to make the distribution of phase continuous. Unwrapping requires knowledge of the fringe sign and the phase of the starting point.

Currently, most phase unwrapping algorithms are applied to 2D phase defined in (x, y). With the capability of phase measurement extends to a third dimension (z in space coordinate or time t), there is a need for a 3D phase unwrapping algorithm.

1.2 Background Information

Optical coherence tomography (OCT) is a non-contact imaging technique that produces higher resolution cross-sectional images from within optical scattering media (e.g., biological tissue). A principle called low-coherence interferometry is used, where the original light beam is divided into two, one of which is made to hit on the targeted tissue and the scattered back-reflected light is merged with a second beam (reference beam). This technique is valuable in areas where traditional microscopic diagnosis by means of biopsy is dangerous or not available.

It is a pioneering technology that has taken steps from the laboratory to the clinic and back again. The use of OCT technology was initially constrained to retinal imaging and later pursued a unique path in clinical practice. It has been medically illustrated in a varied set of medical and surgical specialties. The rise in OCT investigations has taken place in parallel with the development of next-generation technologies that have opened new boundaries of OCT application. An OCT instrument comprises a Michelson-type interferometer with a focused sample arm beam and a lateral-scanning mechanism in basic form. OCT instrumentation has undergone significant improvement in the past years, elated by a series of technological advancements, and, as a result, can be adopted in various applications.

Phase imaging is an imaging technique that has a wide range of applications. It exploits differences in the refractive index of different materials to differentiate between structures under analysis. This has uses in biological, medical and geological science.

Phase imaging has emerged as a valuable method for investigating cells and tissues. This technique operates on unlabeled specimens and is complementary to established fluorescence microscopy. Also, this methodology demonstrates lower phototoxicity and no photobleaching. Phase imaging provides an equitable measure of morphology and dynamics, with zero variability due to contrast agents. Considering the stupendous advancement witnessed especially in the past 10–15 years, several technologies have become sufficiently reliable and translated to biomedical laboratories. The Phase Imaging field is now transitioning from a technology-development-driven to an application-focused field due to commercialization efforts.

1.3 Objective

In this study, we develop a 3D phase unwrapping method that exploits the correlation of 3D phase data ($\phi\,(x, y, t)$) over the other dimension (t). We validate our 3D unwrapping method using simulated data, as well as experimental data obtained by optically computed phase microscopy (OCPM) recently developed in our lab. Our results show that the 3D phase unwrapping method achieves accurate and robust reconstruction of phase $\phi\,(x, y, t)$. The application of the 3D phase unwrapping method is not limited to OCPM. Instead, it is a generic approach that can be used for other phase imaging modalities.

The PSNR (Peak Signal to Noise Ratio) is low compared to other methods, which in turn increases the accuracy of the proposed method. The results have an additional

dimension, time which is not found in conventional methods. With this, the outputs can be viewed in video format for better understanding of the samples, and this is more of a generalized approach. All of these contribute to the reconstruction of phase with robustness, opening a wide range of research opportunities.

CHAPTER 2

THEORY

In phase resolved measurements, the phase is obtained by taking the argument of a complex

number: $\psi\,(x,\,y,\,t) = \arg\,(S\,(x,\,y,\,t)) = atan\,[\mathrm{Imag}\,(S)\,/\,\mathrm{Real}\,(S)]$ and $\psi\,(x,\,y,\,t) \in (-\pi,\,\pi]$.

The wrapped phase ($\psi\,(x,\,y,\,t)$) and actual phase ($\phi\,(x,\,y,\,t)$) has the relationship in Eq (1)

where n is an integer and W indicates the wrapping operation. For sampled data, Eq (1) is

discretized as Eq (2) ($x = x_0 + i\delta x$, $y = y_0 + j\delta y$, $t = t_0 + k\delta t$ where i, j and k are integers). $\phi_{i,}$

$_{j,\,k}$ is the element of 3D data $\boldsymbol{\Phi}$ with indices i, j and k.

$$\Psi\,(x,\,y,\,t) = W\,[\phi\,(x,\,y,\,t)] = \phi\,(x,\,y,\,t) + 2\pi n \quad (1)$$

$$\psi_{i,\,j,\,k} = W\,(\phi_{i,\,j,\,k}) = \phi_{i,\,j,\,k} + 2\pi n \quad (2)$$

Given sufficiently dense spatial and temporal sampling, the phase differences in i,

j and k for ϕ ($\phi_{i,\,j,\,k}$ - $\phi_{i-1,\,j,\,k}$, $\phi_{i,\,j,\,k}$ - $\phi_{i,\,j-1,\,k}$, $\phi_{i,\,j,\,k}$ - $\phi_{i,\,j,\,k-1}$) are unlikely to take values beyond

$(-\pi,\,\pi]$. If the phase differences for ψ ($\psi_{i,\,j,\,k}$ - $\psi_{i-1,\,j,\,k}$, $\psi_{i,\,j,\,k}$ - $\psi_{i,\,j-1,\,k}$, $\psi_{i,\,j,\,k}$ - $\psi_{i,\,j,\,k-1}$) are smaller

than $-\pi$ or larger than π, it is likely due to phase wrapping and the actual phase difference

is estimated through the wrapping operation ($W\,(\psi_{i,\,j,\,k}$ - $\psi_{i-1,\,j,\,k})$, $W\,(\psi_{i,\,j,\,k}$ - $\psi_{i,\,j-1,\,k})$, $W\,(\psi_{i,}$

$_{j,\,k}$ - $\psi_{i,\,j,\,k-1}$)). The goal of 3D phase unwrapping is to determine $\phi_{i,\,j,\,k}$ from the measurement

of $\psi_{i,\,j,\,k}$, such that the change of ϕ is consistent with the wrapped change of ψ in least square

sense. In other words, the goal of 3D unwrapping is to obtain $\phi_{i,\,j,\,k}$ that minimizes f defined

in Eq (3) where β is a weight for the third dimension.

$$f = \Sigma_i \Sigma_j \Sigma_k [\phi_{i,j,k} - \phi_{i-1,j,k} - W\,(\psi_{i,j,k} - \psi_{i-1,j,k})]^2 + \Sigma_i \Sigma_j \Sigma_k [\phi_{i,j,k} - \phi_{i,j-1,k} -$$

$$W\,(\psi_{i,j,k} - \psi_{i,j-1,k})]^2 + \beta\,\Sigma_i \Sigma_j \Sigma_k [\phi_{i,j,k} - \phi_{i,j,k-1} - W\,(\psi_{i,j,k} - \psi_{i,j,k-1})]^2 \quad (3)$$

To solve the least square problem described above, we denote $\boldsymbol{\phi}_{jk} = [\phi_{1,j,k}, \phi_{2,j,k}, \phi_{3,j,k}, ...]'$; $\boldsymbol{\phi}_{ik} = [\phi_{i,1,k}, \phi_{i,2,k}, \phi_{i,3,k}, ...]'$; $\boldsymbol{\phi}_{ij} = [\phi_{i,j,1}, \phi_{i,j,2}, \phi_{i,j,3}, ...]'$; $\boldsymbol{v}_{jk} = \boldsymbol{W}(\psi_{i,j,k} - \psi_{i-1,j,k})$, $\boldsymbol{v}_{ik} = \boldsymbol{W}(\psi_{i,j,k} - \psi_{i,j-1,k})$, $\boldsymbol{v}_{ij} = \boldsymbol{W}(\psi_{i,j,k} - \psi_{i,j,k-1})$, and re-write Eq (3) as Eq (4) where A = [1 - 1 0 0 1 - 1 ... ...] ($A_{i,j} = 1$ for $I = j$; $A_{i,j} = -1$ for $j = i + 1$; otherwise, $A_{i,j} = 0$).

$$f = jk\|A_{jk} - v_{jk}\|2 + ik\|A_{ik} - v_{ik}\|2 + \beta ij\|A_{ij} - v_{ij}\|2 \quad (4)$$

f is minimized when its gradient regarding $\phi_{i,j,k}$ vanishes to 0, leading to Eq (5) and then Eq (6) where $\rho_{i,j,k}$ is defined as Eq (7).

$$2\phi_{i,j,k} - \phi_{i-1,j,k} - \phi_{i+1,j,k} - [\boldsymbol{W}(\psi_{i,j,k} - \psi_{i-1,j,k}) - \boldsymbol{W}(\psi_{i-1,j,k} - \psi_{i-2,j,k})] + 2\phi_{i,j,k} - \phi_{i,j-1,k} - \phi_{i,j+1,k} -$$

$$[\boldsymbol{W}(\psi_{i,j,k} - \psi_{i,j-1,k}) - \boldsymbol{W}(\psi_{i,j-1,k} - \psi_{i,j-2,k})] + 2\beta\phi_{i,j,k} - \beta\phi_{i,j,k-1} - \beta\phi_{i,j,k+1} - \beta[\boldsymbol{W}(\psi_{i,j,k} - \psi_{i,j,k-1}) - \boldsymbol{W}(\psi_{i,j-1,k} - \psi_{i,j,k-2})] = 0 \quad (5)$$

$$(4 + 2\beta)\phi_{i,j,k} - \phi_{i-1,j,k} - \phi_{i+1,j,k} - \phi_{i,j-1,k} - \phi_{i,j+1,k} - \beta\phi_{i,j,k-1} - \beta\phi_{i,j,k+1} = \rho_{i,j,k} \quad (6)$$

$$\rho_{i,j,k} = \boldsymbol{W}(\psi_{i,j,k} - \psi_{i-1,j,k}) - \boldsymbol{W}(\psi_{i-1,j,k} - \psi_{i-2,j,k}) + \boldsymbol{W}(\psi_{i,j,k} - \psi_{i,j-1,k}) - \boldsymbol{W}(\psi_{i,j-1,k} - \psi_{i,j-2,k})$$

$$+ \beta\boldsymbol{W}(\psi_{i,j,k} - \psi_{i,j,k-1}) - \beta\boldsymbol{W}(\psi_{i,j-1,k} - \psi_{i,j,k-2}) \quad (7)$$

Applying 3D Discrete cosine transform (DCT) on Eq (6), we obtain Eq (8) where Φ and ρ are the DCT of $\boldsymbol{\Phi}$ and $\boldsymbol{\rho}$ ($\Phi = \boldsymbol{D}\boldsymbol{\Phi}$ and $\rho = \boldsymbol{D}\boldsymbol{\rho}$), respectively. Φ can be obtained from ρ through simple algebraic operation, as shown in Eq (9). By computing $\boldsymbol{\rho}$ and its DCT directly from the wrapped phase, we obtain Φ and the unwrapped phase $\boldsymbol{\Phi}$ in spatial-temporal domain through inverse 3D DCT ($\boldsymbol{D}^{-1}$) as shown in Eq (10).

$$(4 + 2\beta)\hat{\phi}_{i',j',k'} - 2\cos\left(\frac{\pi i'}{2N_x}\right)\hat{\phi}_{i',j',k'} - 2\cos\left(\frac{\pi j'}{2N_y}\right)\hat{\phi}_{i',j',k'} -$$

$$2\beta\cos\left(\frac{\pi k'}{2N_t}\right)\hat{\phi}_{i',j',k'} = \hat{\rho}_{i',j',k'} \quad (8)$$

$$\widehat{\phi}_{i',j',k'} = \frac{\widehat{\rho}_{i',j',k'}}{4+2\beta-2cos\left(\frac{\pi i'}{2N_x}\right) - 2cos\left(\frac{\pi j'}{2N_y}\right) - 2\beta cos\left(\frac{\pi k'}{2N_t}\right)} \quad (9)$$

$$\boldsymbol{\Phi = D^{-1} \left(\widehat{\phi}\right)} \quad (10)$$

CHAPTER 3

IMPLEMENTATION

We first validated 3D phase unwrapping using simulated data: $\phi_{i, j, k} = Ae^{-\frac{(i-i_0)^2 + (j-j_0)^2}{2c^2}}$

where i, j and k are indices for x coordinate, y coordinate and time, respectively ($i = 0, 1,$

$2, \ldots N_x$-1; $j = 0, 1, 2, \ldots N_y$-1; $k = 0, 1, 2, \ldots N_t$-1). Figure 3.1 (a) shows the ground truth

2D phase simulated with $A = 10\pi$, $c = 20$, and $k = 0$. The wrapped phase is shown in Figure

3.1 (b) $\psi_{i, j, k} = W (\phi_{i, j, k} + n_{i, j, k})$ with a noise standard deviation of 0.5 ($\sigma_n = 0.5$). We

implemented 2D phase unwrapping described in [REF] and obtained the result in Figure

3.1 (c). We also utilized 3D phase unwrapping to obtain the result in Figure 3.1 (d).

Although the results 2D and 3D unwrapping both follow the profile of the ground truth,

the 2D unwrapped phase is noisier compared to the 3D unwrapped phase. This can be seen

more clearly in Figure 3.1 (e) where we plot the value along the central row of the 2D phase
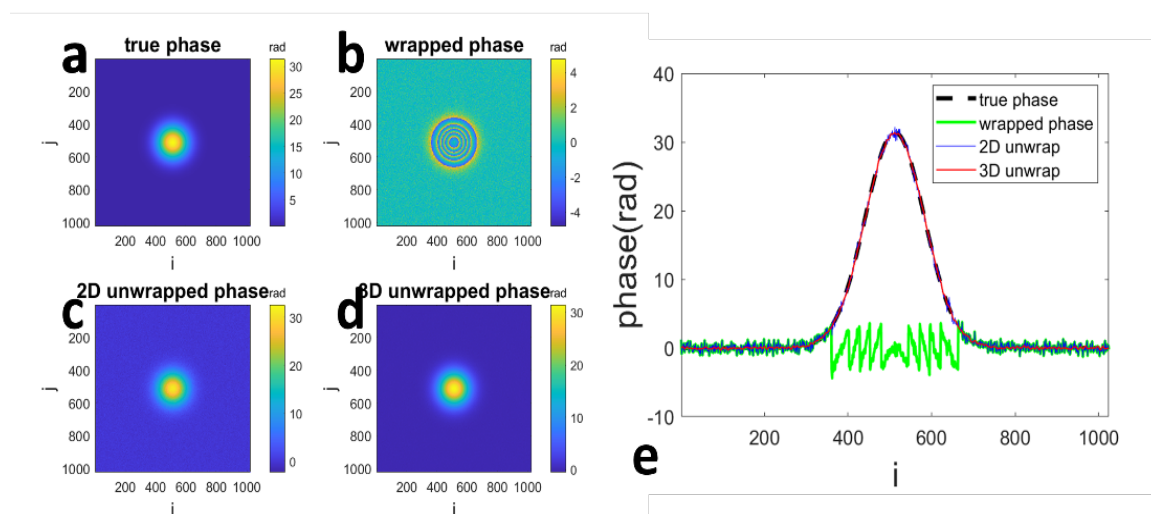
image ($\psi_{i, j, k}$ for $k = 0$).

Figure 3.1 (a) A 2D frame of unwrapped phase; (b) wrapped phase with noise; (c) 2D unwrapped phase; (d) 3D unwrapped phase; (e) phase along the central row.

To quantitatively compare the quality of 3D and 2D phase unwrapping, we simulated wrapped phase at different noise levels (σ_n = 0.1, 0.2, 0.3, …). We calculated the Peak Signal to Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM) for 2D and 3D unwrapped phase images. The results obtained are shown in Figure 3.2 (a) and (b). 3D phase unwrapping outperforms 2D phase unwrapping in both metrics, suggesting a better recovery of ground truth phase image.
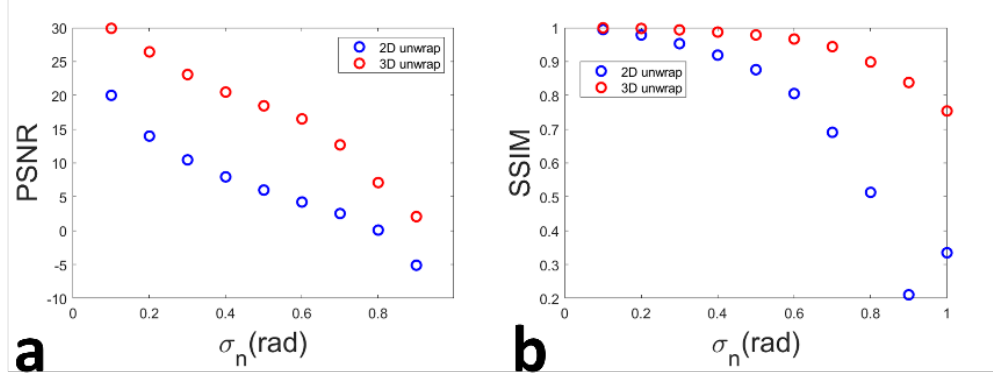
Figure 3.2 (a) PSNR of 2D and 3D unwrapped phase at different noise levels; (b) SSIM of 2D and 3D unwrapped phase at different noise levels.

For a sequence of phase images acquired over time, 3D unwrapping considers the phase change between frames (the first and second terms in Eq (2) and (3)), even when the phase shift is imposed globally to all the pixels. On the other hand, 2D phase unwrapping tries to match the phase change along x and y dimension. Hence a solution of 2D unwrapping problem (Φ_{2D}) remains valid, when the solution is offset by an arbitrary constant ($\Phi_{2D}+\Phi_0$). This will lead to incorrect reconstruction of global phase shift between frames. To demonstrate the advantage of 3D unwrapping in preserving the inter-frame phase shift, we simulated 3D phase data with a constant phase shift between adjacent frames: $\phi_{i,j,k} = Ae^{-\frac{(i-i_0)^2+(j-j_0)^2}{2c^2}} + vk$ where $A = 10\pi$, $c = 20$, and $v = 0.2\pi$. Assuming the

phase represents the 3D topology of a surface, we show the ground truth, wrapped phase

$(\psi_{i,j,k} = W(\phi_{i,j,k} + n_{i,j,k}))$ with a noise standard deviation of 0.5), 2D unwrapped phase and

3D unwrapped phase as 3D rendered surface profile in Figure 3.3 (a) – (d) ($k = 0$). As the

index k increase, $\phi_{i,j,k}$ experiences a global shift, as demonstrated in the video showing the

rendered phase at different time. The 3D unwrapped phase preserves the relative phase

variation within each frame and follows the global phase shift between different frames.

However, the 2D unwrapped phase does not follow the global phase shift due to an arbitrary

offset added to the reconstruction. To further demonstrate the difference, we evaluate the

phase values at $i = i_0$ and $j = j_0$. Figure 3.3 (e) shows the ground truth phase $\phi_{i0,j0,k}$ (black

dashed line), wrapped phase $\psi_{i0,j0,k}$ (green,), 2D unwrapped phase (blue), and 3D

unwrapped phase (red). The result of 3D unwrapping accurately follows the ground truth,

while the result of 2D unwrapping at the point remains the same, because of the global
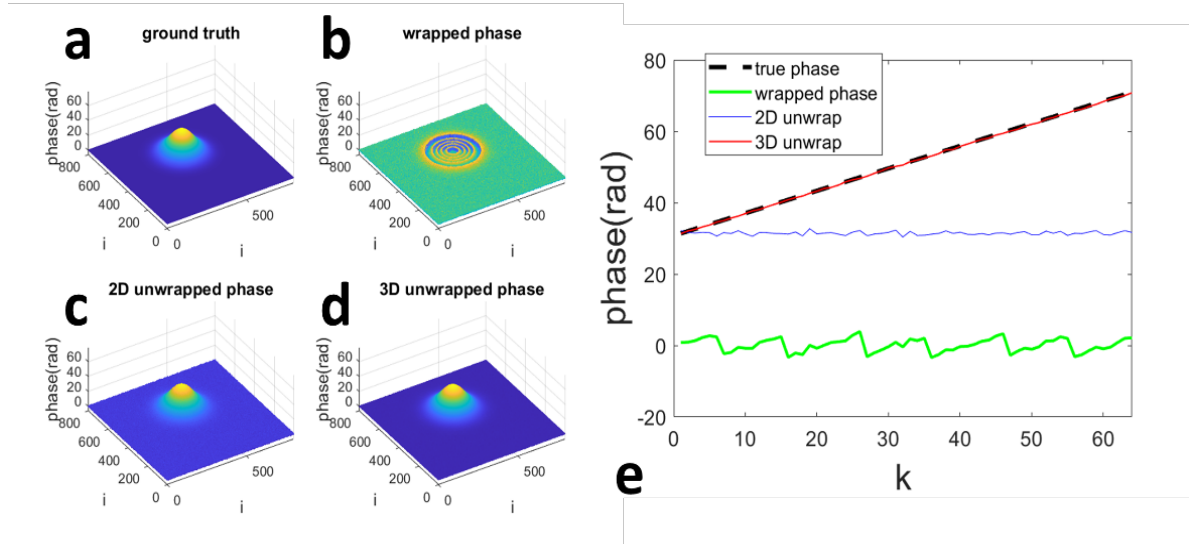
offset at each frame.

Figure 3.3 Rendered 3D surface topology based on ground truth phase (a), wrapped phase (b), (c) 2D unwrapped phase, and (d) 3D unwrapped phase; (e) the values at the center of the 2D image of the ground truth, wrapped phase, 2D unwrapped phase and 3D unwrapped phase.

We further applied 3D unwrapping to experimental data. We performed OCPM imaging on cultured cells with motion excited by magnetic nanoparticles. Details about the imaging platform and sample have been described in our previous publication. Figure 3.4 shows the first frame of the image sequence. Figure 3.4 (a) and (d) show the wrapped phase and 3D rendered surface profile according to the wrapped phase. Similarly, Figure 3.4 (b) and (d) correspond to the 2D unwrapped phase. Similarly, Figure 3.4 (c) and (e) correspond to the 3D unwrapped phase. The red arrows indicate the cells, and the black arrows indicate a cluster of magnetic nanoparticles. The contrast of phase imaging originates from the cells and the nanoparticles that altered the optical path length. Figure 3.4 (a) is apparently affected by phase wrapping artifacts. The region corresponding to the cells show alternation between yellow and blue, suggesting phase jump from π to $-\pi$. The phase wrapping artifact is quite prominent, because the cells took a spherical shape, after we processed the petri dish with Trypsin and detached the cells from the bottom of the petri dish. After unwrapping, Figure 3.5 (b), (c), (e) and (f) do not show abrupt change in phase values. Notably, the image obtained from 3D unwrapping has better contrast compared to the image obtained from 2D unwrapping. For example, the magnetic nanoparticle cluster indicated by the black arrow can be visualized more clearly in Figure 3.4 (c) and (e), compared to in Figure 3.4 (b) and (d). This is consistent with results in Figure 3.2 that suggests 3D unwrapping achieves better reconstruction quality. The advantage of 3D unwrapping compared to 2D unwrapping can be seen more clearly in Figure 3.5 that compares frames of 2D unwrapped and 3D unwrapped images obtained at different time ($t = 0$, $t = 0.12$s, $t = 0.24$s, and $t = 0.36$s). The arrows in Figure 3.5 (e) – (h) point to a cluster of magnetic nanoparticles that moved under external magnetic field and pushed the

detached cell to move. Due to a suboptimal contrast, the cluster of magnetic nanoparticles is not visible in images obtained from 2D unwrapping. The video in compares the 3D surface profiles rendered based on wrapped phase, 2D and 3D unwrapped phase.
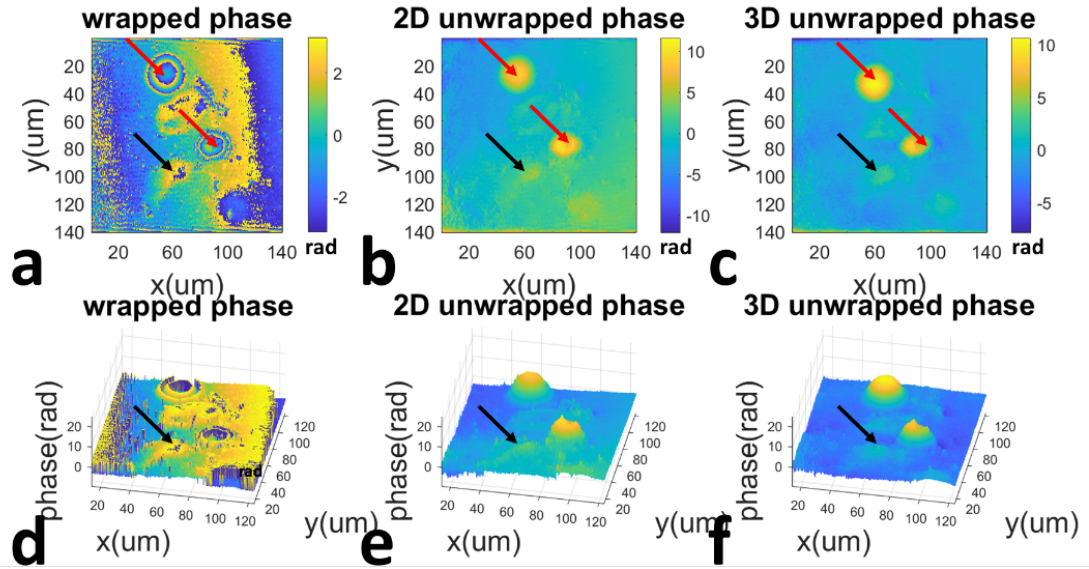
Figure 3.4 OCPM image obtained from a sample with detached HeLa cells and magnetic nanoparticles (a) wrapped phase; (d) 3D surface topology rendered using wrapped phase; (b) 2D unwrapped phase; (e) 3D surface topology rendered using 2D unwrapped phase; (c) 3D unwrapped phase; (f) 3D surface topology rendered using 3D unwrapped phase.
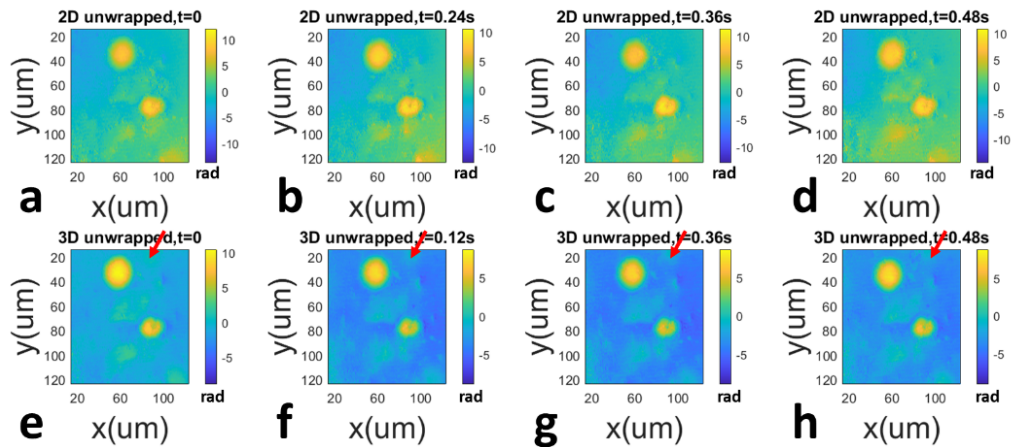
Figure 3.5 2D unwrapped images of detached HeLa cells and magnetic nanoparticles obtained at $t = 0$ (a), $t = 0.12$s (b), $t = 0.24$s (c), and $t = 0.36$s (d); 3D unwrapped images of detached HeLa cells and magnetic nanoparticles obtained at $t = 0$ (e), $t = 0.12$s (f), $t = 0.24$s (g), and $t = 0.36$s (h).

We utilized OCPM to image the process of cell detachment. We introduced Trypsin to petri dish with HeLa cells to initiate the cell detachment process. A sequence of phase images was acquired to study the detachment process. Figure 3.6 (a) and (b) show the wrapped phase image and 3D unwrapped phase image at t = 14min. The abrupt change from yellow to blue or blue to yellow in Figure 3.6 (a) is the phase wrapping artifact. Figure 3.6 (b) is free of such artifact. Figure 3.6 (c) and (d) show the 3D surface profile rendered with the wrapped and 3D unwrapped phase image. The video shows the change of surface profile over time using 3D unwrapped phase images.
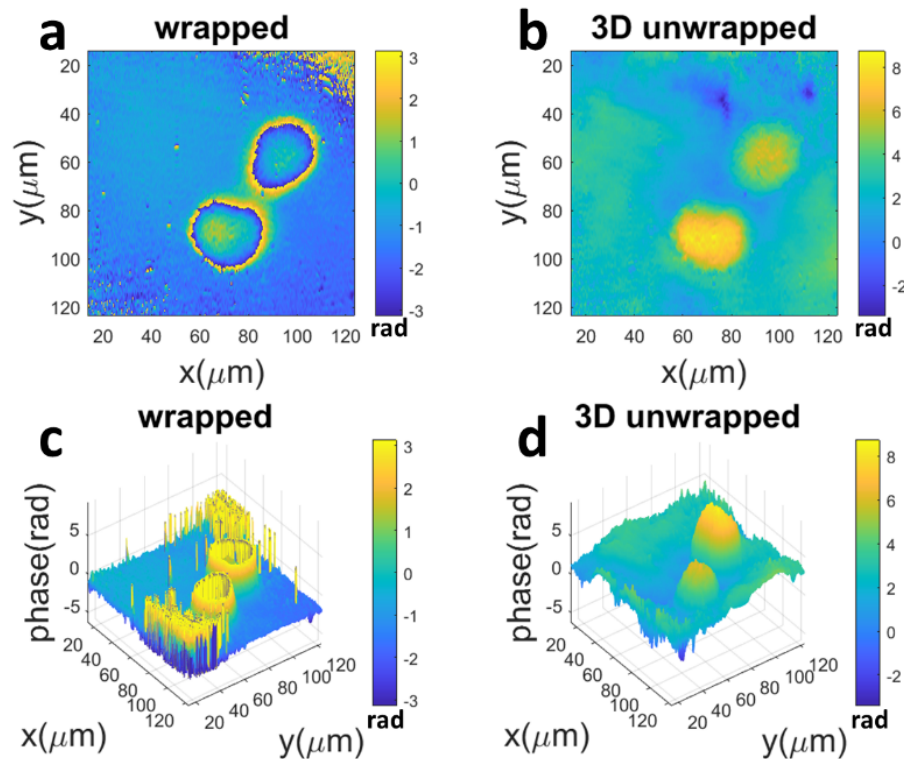

Figure 3.6 (a) Wrapped phase images of HeLa cells when the detachment started; (b) 3D unwrapped phase images of HeLa cells when the detachment started; (c) 3D surface topology rendered using wrapped phase; (d) 3D surface topology rendered using 3D unwrapped phase.

To illustrate how cell morphology changes during detachment, we show the phase image (3D unwrapped) of one cell (bottom cell in Figure 3.6 (a) and (b)) obtained at different time in Figure 3.7. As the cell detached from the surface of the petri dish, the lateral extension of the cell became smaller and the phase retardation due to change in optical path length increases.
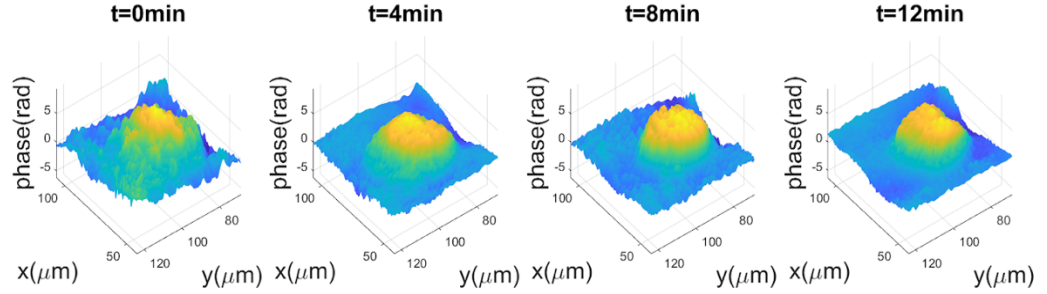
Figure 3.7 3D unwrapped phase of a cell detaching from the surface of the petri dish at different time: (a) t = 0min; (b) t = 4min; (c) t = 8min; (d) t = 12min.

Results in Figures 3.6 and 3.7 imply that accurate phase unwrapping is critical for quantitative study of cellular morphology change during detachment. To further validate this, we calculated the optical path lengths at the center of both cells in Figure 3.6 using wrapped (ψ) and 3D unwrapped ($\hat{\phi}$) phases: $d_{\text{wrapped}} (x_0, y_0, t) = \lambda_0 \psi (x_0, y_0, t) / (4\pi)$ and $d_{\text{unwrapped}} (x_0, y_0, t) = \lambda_0 (x_0, y_0, t) / (4\pi)$. Figure 3.8 (a) and (b) display the change of optical path length (d_{wrapped} and $d_{\text{unwrapped}}$) for the center of the top cells. Figure 3.8 (c) and (d) correspond to results obtained from the bottom cells. Figure 3.8 (a) and (c) show large variation of optical path length due to phase wrapping artifact. Figure 3.8 (b) and (d) show the optical path length at the center of the cell increases over time during detachment. This is consistent with the fact that the mass of the cell becomes more concentrated following detachment.
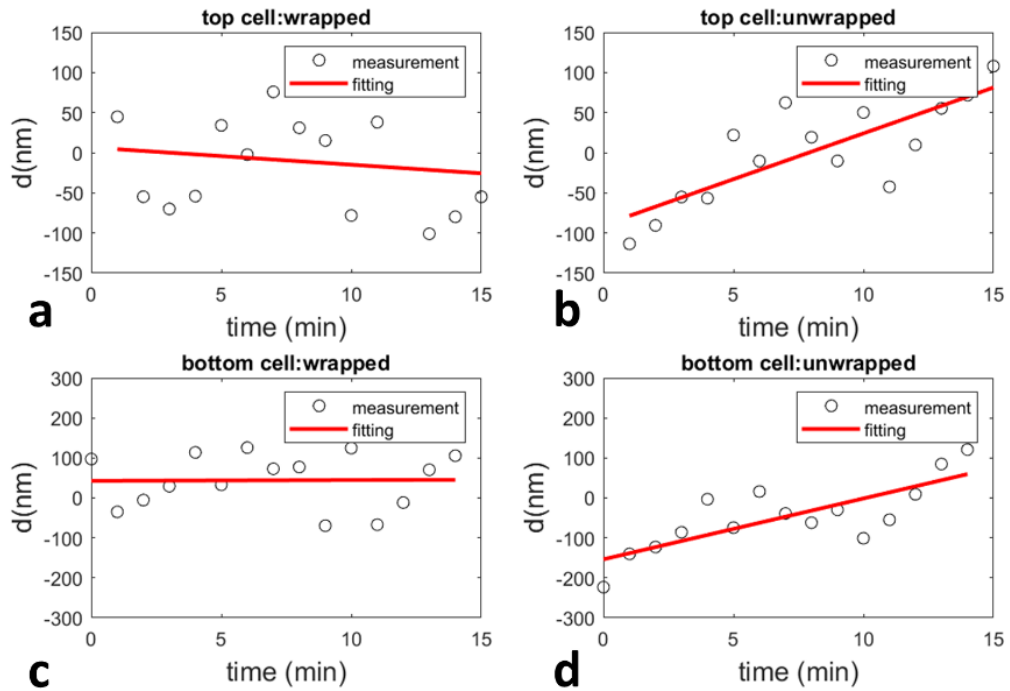
Figure 3.8 Optical path length at the center of the cell (a) from the top cell calculated using wrapped phase; (b) from the top cell calculated using 3D unwrapped phase; (c) from the bottom cell calculated using wrapped phase; (d) from the bottom cell calculated using 3D unwrapped phase.

Code 1

```
clc, clear, close all
load('matlab_move1_3d.mat');
load('phase_2d_unwrap.mat');

phi_unwrap_3d=phase_unwrap_3d(phase_3d(:,:,33:64));

x=1:1:1024;
y=1:1:1024;
x=x*5.5/40;
y=y*5.5/40;
phase_wrap_disp=-phase_3d(:,:,4);
phase_3d_disp=-phi_unwrap_3d(:,:,4);
phase_2d_disp=-phi_unwrap_2d(:,:,4);

figure('Position', [100 100 1200 350]);
subplot(2,3,4)
mesh(x(101:900),(y(101:900)),flipud(phase_wrap_disp(101:900,101:900)));
zlim([-3*pi 8*pi])
xlim([x(101) x(900)]),ylim([y(101) y(900)])
view([11 53])
xlabel('x(um)','FontSize',22),ylabel('y(um)','FontSize',22),
zlabel('phase(rad)','FontSize',22),
title('wrapped phase','FontSize',22)

subplot(2,3,5)
mesh(x(101:900),(y(101:900)),flipud(phase_2d_disp(101:900,101:900)));
zlim([-3*pi 8*pi])
xlim([x(101) x(900)]),ylim([y(101) y(900)])
view([11 53])
xlabel('x(um)','FontSize',22),ylabel('y(um)','FontSize',22),
zlabel('phase(rad)','FontSize',22),
title('2D unwrapped phase','FontSize',22)

subplot(2,3,6)
mesh(x(101:900),(y(101:900)),flipud(phase_3d_disp(101:900,101:900)));
zlim([-3*pi 8*pi])
xlim([x(101) x(900)]),ylim([y(101) y(900)])
view([11 53])
```

```matlab
xlabel('x(um)','FontSize',22),ylabel('y(um)','FontSize',22),
zlabel('phase(rad)','FontSize',22),
title('3D unwrapped phase','FontSize',22)

subplot(2,3,1)
imagesc(x,y,phase_wrap_disp),colorbar
set(gca,'FontSize',14),xlabel('x(um)','FontSize',22), ylabel('y(um)','FontSize',22),truesize,
title('wrapped phase','FontSize',22)

subplot(2,3,2)
imagesc(x,y,phase_2d_disp),colorbar,
set(gca,'FontSize',14),xlabel('x(um)','FontSize',22), ylabel('y(um)','FontSize',22),truesize,
title('2D unwrapped phase','FontSize',22)

subplot(2,3,3)
imagesc(x,y,phase_3d_disp),colorbar,
set(gca,'FontSize',14),xlabel('x(um)','FontSize',22), ylabel('y(um)','FontSize',22),truesize,
title('3D unwrapped phase','FontSize',22)


x=1:1:1024;
y=1:1:1024;
x=x*5.5/40;
y=y*5.5/40;
phase_wrap_disp=-phase_3d(:,:,4);
phase_3d_disp=-phi_unwrap_3d(:,:,4);
phase_2d_disp=-phi_unwrap_2d(:,:,4);
i0=25;a0=2;
figure('Position', [100 100 1200 450]);
subplot(2,4,1)
imagesc(x,y,-phi_unwrap_2d(:,:,i0)),colorbar
xlim([x(101) x(900)]),ylim([y(101) y(900)])
xlabel('x(um)','FontSize',22),ylabel('y(um)','FontSize',22),
zlabel('phase(rad)','FontSize',22),
title('2D unwrapped,t=0','FontSize',12)

subplot(2,4,2)
imagesc(x,y,-phi_unwrap_2d(:,:,i0+a0)),colorbar
xlim([x(101) x(900)]),ylim([y(101) y(900)])
xlabel('x(um)','FontSize',22),ylabel('y(um)','FontSize',22),
zlabel('phase(rad)','FontSize',22),
title('2D unwrapped,t=0.24s','FontSize',12)

subplot(2,4,3)
imagesc(x,y,-phi_unwrap_2d(:,:,i0+a0*2)),colorbar
xlim([x(101) x(900)]),ylim([y(101) y(900)])
```

```matlab
xlabel('x(um)','FontSize',22),ylabel('y(um)','FontSize',22),
zlabel('phase(rad)','FontSize',22),
title('2D unwrapped,t=0.36s','FontSize',12)

subplot(2,4,4)
imagesc(x,y,-phi_unwrap_2d(:,:,i0+a0*3)),colorbar
xlim([x(101) x(900)]),ylim([y(101) y(900)])
xlabel('x(um)','FontSize',22),ylabel('y(um)','FontSize',22),
zlabel('phase(rad)','FontSize',22),
title('2D unwrapped,t=0.48s','FontSize',12)


subplot(2,4,5)
imagesc(x,y,-phi_unwrap_3d(:,:,i0)),colorbar
xlim([x(101) x(900)]),ylim([y(101) y(900)])
xlabel('x(um)','FontSize',22),ylabel('y(um)','FontSize',22),
zlabel('phase(rad)','FontSize',22),
title('3D unwrapped,t=0','FontSize',12)

subplot(2,4,6)
imagesc(x,y,-phi_unwrap_3d(:,:,i0+a0)),colorbar
xlim([x(101) x(900)]),ylim([y(101) y(900)])
xlabel('x(um)','FontSize',22),ylabel('y(um)','FontSize',22),
zlabel('phase(rad)','FontSize',22),
title('3D unwrapped,t=0.12s','FontSize',12)

subplot(2,4,7)
imagesc(x,y,-phi_unwrap_3d(:,:,i0+a0*2)),colorbar
xlim([x(101) x(900)]),ylim([y(101) y(900)])
xlabel('x(um)','FontSize',22),ylabel('y(um)','FontSize',22),
zlabel('phase(rad)','FontSize',22),
title('3D unwrapped,t=0.36s','FontSize',12)

subplot(2,4,8)
imagesc(x,y,-phi_unwrap_3d(:,:,i0+a0*3)),colorbar
xlim([x(101) x(900)]),ylim([y(101) y(900)])
xlabel('x(um)','FontSize',22),ylabel('y(um)','FontSize',22),
zlabel('phase(rad)','FontSize',22),
title('3D unwrapped,t=0.48s','FontSize',12)

figure('Position', [100 100 1200 350]);

for frame_index=1:1:32

    frame_index
```

```matlab
   subplot(1,3,1)
   mesh((x(101:900)),y(1:800),flipud(-phase_3d(1:800,101:900,frame_index)));
   zlim([-3*pi 8*pi])
   xlim([x(101) x(900)]),ylim([y(1) y(800)])
   xlabel('x(um)','FontSize',22),ylabel('y(um)','FontSize',22),
zlabel('phase(rad)','FontSize',22),
   title(['wrapped phase t=' num2str(frame_index*0.12,'%0.2f') 's'],'FontSize',16),
   view([140 52])

   subplot(1,3,2)
   mesh((x(101:900)),y(1:800),flipud(-phi_unwrap_2d(1:800,101:900,frame_index)));
   zlim([-3*pi 8*pi])
   xlim([x(101) x(900)]),ylim([y(1) y(800)])
   xlabel('x(um)','FontSize',22),ylabel('y(um)','FontSize',22),
zlabel('phase(rad)','FontSize',22),
   title(['2D unwrapped phase t=' num2str(frame_index*0.12,'%0.2f') 's'],'FontSize',16),
   view([140 52])

   subplot(1,3,3)
   mesh((x(101:900)),y(1:800),flipud(-(phi_unwrap_3d(1:800,101:900,frame_index))));
   zlim([-3*pi 8*pi])
   xlim([x(101) x(900)]),ylim([y(1) y(800)])
   xlabel('x(um)','FontSize',22),ylabel('y(um)','FontSize',22),
zlabel('phase(rad)','FontSize',22),
   title(['3D unwrapped phase t=' num2str(frame_index*0.12,'%0.2f') 's'],'FontSize',16),
   view([140 52])
   F(frame_index) = getframe(gcf) ;

   drawnow
end
writerObj = VideoWriter('3d_2d_dis0118');
writerObj.FrameRate = 9;
% set the seconds per image
% open the video writer
open(writerObj);
for i=1:32
   % convert the image to a frame
   frame = F(i) ;
   writeVideo(writerObj, frame);
   return
end

close(writerObj);
```

REFERENCE CODE 2

Code 2

```
clc, clear, close all

load('phase_3d_detach_0116.mat');
phase_3d=phi;
phi_unwrap_3d = phase_unwrap_3d(phase_3d);

x=1:1:1024;
y=1:1:1024;
x=x*5.5/40;
y=y*5.5/40;
phase_wrap_disp=phase_3d(:,:,15);
phase_3d_disp=phi_unwrap_3d(:,:,15);

figure('Position', [0 0 800 600]);
subplot(2,2,1)
imagesc(x(101:900),(y(101:900)),(phase_wrap_disp(101:900,101:900)));
colorbar
xlim([x(101) x(900)]),ylim([y(101) y(900)])
xlabel('x(\mum)','FontSize',18),ylabel('y(\mum)','FontSize',18),
zlabel('phase(rad)','FontSize',18),
title('wrapped','FontSize',18),

subplot(2,2,2)
imagesc(x(101:900),(y(101:900)),(phase_3d_disp(101:900,101:900)));
colorbar
xlim([x(101) x(900)]),ylim([y(101) y(900)])
xlabel('x(\mum)','FontSize',18),ylabel('y(\mum)','FontSize',18),
zlabel('phase(rad)','FontSize',18),
title('3D unwrapped','FontSize',18),

subplot(2,2,3)
mesh(x(101:900),(y(101:900)),(phase_wrap_disp(101:900,101:900)));
colorbar
zlim([-2*pi 3*pi])
xlim([x(101) x(900)]),ylim([y(101) y(900)])
xlabel('x(\mum)','FontSize',18),ylabel('y(\mum)','FontSize',18),
zlabel('phase(rad)','FontSize',18),
title('wrapped','FontSize',18),
view([53 41])
```

```
subplot(2,2,4)
mesh(x(101:900),(y(101:900)),(phase_3d_disp(101:900,101:900)));
colorbar
zlim([-2*pi 3*pi])

xlim([x(101) x(900)]),ylim([y(101) y(900)])
xlabel('x(\mum)','FontSize',18),ylabel('y(\mum)','FontSize',18),
zlabel('phase(rad)','FontSize',18),
title('3D unwrapped','FontSize',18),

view([53 41])

N_int=1;
Nd=4;
figure('Position', [100 100 1400 300]);
subplot(1,4,1)
mesh(y(301:775),x(501:900),(phi_unwrap_3d(501:900,301:775,N_int)-
mean(mean(phi_unwrap_3d(501:600,301:400,N_int)))));
zlim([-2*pi 3*pi])
view([-128 47])

title(['t=' num2str((N_int-1))
'min'],'FontSize',18),xlabel('x(\mum)','FontSize',18),ylabel('y(\mum)','FontSize',18),
zlabel('phase(rad)','FontSize',18),

subplot(1,4,2)
mesh(y(301:775),x(501:900),(phi_unwrap_3d(501:900,301:775,N_int+Nd)-
mean(mean(phi_unwrap_3d(501:600,301:400,N_int+Nd)))));
zlim([-2*pi 3*pi])
view([-128 47])
title(['t=' num2str((N_int+Nd-1))
'min'],'FontSize',18),xlabel('x(\mum)','FontSize',18),ylabel('y(\mum)','FontSize',18),
zlabel('phase(rad)','FontSize',18),

subplot(1,4,3)
mesh(y(301:775),x(501:900),(phi_unwrap_3d(501:900,301:775,N_int+Nd*2)-
mean(mean(phi_unwrap_3d(501:600,301:400,N_int+Nd*2)))));
zlim([-2*pi 3*pi])
view([-128 47])
title(['t=' num2str((N_int+Nd*2-1))
'min'],'FontSize',18),xlabel('x(\mum)','FontSize',18),ylabel('y(\mum)','FontSize',18),
zlabel('phase(rad)','FontSize',18),

subplot(1,4,4)
mesh(y(301:775),x(501:900),(phi_unwrap_3d(501:900,301:775,N_int+Nd*3)-
mean(mean(phi_unwrap_3d(501:600,301:400,N_int+Nd*3)))));
```

```matlab
zlim([-2*pi 3*pi])
view([-128 47])
title(['t=' num2str((N_int+Nd*3-1))
'min'],'FontSize',18),xlabel('x(\mum)','FontSize',18),ylabel('y(\mum)','FontSize',18),
zlabel('phase(rad)','FontSize',18),

N1=520;N2=660;

W=0;
ROI_unwrap=phi_unwrap_3d(N1-W:N1+W,N2-W:N2+W,:);
mean_ROI_unwrap=permute(mean(mean(ROI_unwrap,1),2),[3 1 2]);

ROI_wrapped=phase_3d(N1-W:N1+W,N2-W:N2+W,:);
mean_ROI_wrapped=permute(mean(mean(ROI_wrapped,1),2),[3 1 2]);

t=(1:1:15);

p_wrapped_fit=polyval(polyfit(t,mean_ROI_wrapped,1),t);
p_unwrap_fit=polyval(polyfit(t,mean_ROI_unwrap,1),t);

t_m=0:1:14;

figure('Position', [100 100 800 500]);
subplot(2,2,4),
plot(t_m,mean_ROI_unwrap*730/1.44/4/pi,'ko')
hold on
plot(t_m,p_unwrap_fit*730/1.44/4/pi,'r','LineWidth',2)
set(gca,'FontSize',10), xlabel('time (min)','FontSize',14),ylabel('d(nm)','FontSize',14)
legend('measurement','fitting')
title('bottom cell:unwrapped')
ylim([-300 300])

subplot(2,2,3)
plot(t_m,mean_ROI_wrapped*730/1.44/4/pi,'ko')
hold on
plot(t_m,p_wrapped_fit*730/1.44/4/pi,'r','LineWidth',2)
set(gca,'FontSize',10), xlabel('time (min)','FontSize',14),ylabel('d(nm)','FontSize',14)
legend('measurement','fitting')
title('bottom cell:wrapped')
ylim([-300 300])

N1=740;N2=320;
W=0;
ROI_unwrap=phi_unwrap_3d(N1-W:N1+W,N2-W:N2+W,:);
mean_ROI_unwrap=permute(mean(mean(ROI_unwrap,1),2),[3 1 2]);
```

```matlab
ROI_wrapped=phase_3d(N1-10:N1+10,N2-10:N2+10,:);
mean_ROI_wrapped=permute(mean(mean(ROI_wrapped,1),2),[3 1 2]);

p_wrapped_fit=polyval(polyfit(t,mean_ROI_wrapped,1),t);
p_unwrap_fit=polyval(polyfit(t,mean_ROI_unwrap,1),t);

subplot(2,2,2)
plot(mean_ROI_unwrap*730/1.44/4/pi,'ko')
hold on
plot(p_unwrap_fit*730/1.44/4/pi,'r','LineWidth',2)
ylim([-150 150])
set(gca,'FontSize',10), xlabel('time (min)','FontSize',14),ylabel('d(nm)','FontSize',14)
legend('measurement','fitting')
title('top cell:unwrapped')

subplot(2,2,1)
plot(mean_ROI_wrapped*730/1.44/4/pi,'ko')
hold on
plot(p_wrapped_fit*730/1.44/4/pi,'r','LineWidth',2)
ylim([-150 150])
set(gca,'FontSize',10), xlabel('time (min)','FontSize',14),ylabel('d(nm)','FontSize',14)
legend('measurement','fitting')
title('top cell:wrapped')

figure
for frame_index=1:1:15

    frame_index
    mesh((x(101:900)),y(51:980),phi_unwrap_3d(51:980,101:900,frame_index));
     zlim([-250/(730/1.44/4/pi) 350/(730/1.44/4/pi)])
    xlim([x(101) x(900)]),ylim([y(51) y(980)])
    xlabel('x(\mum)','FontSize',18),ylabel('y(\mum)','FontSize',18),  zlabel('phase
(rad)','FontSize',18),
    title(['3D unwrapped phase; t=' num2str(frame_index-1) 'min'],'FontSize',18),

    view([53 41])
     F(frame_index) = getframe(gcf) ;

    drawnow

end

writerObj = VideoWriter('cell_det_0118');
writerObj.FrameRate = 3;
% set the seconds per image
% open the video writer
```

```
open(writerObj);
for i=1:length(F)
    % convert the image to a frame
    frame = F(i) ;
    writeVideo(writerObj, frame);
end

close(writerObj);
```

REFERENCE CODE 3

Code 3

```
clc, clear, close all
imag_i=sqrt(-1);
x=((1:1:1024)-512)*5.5/20;
y=((1:1:1024)-512)*5.5/20;
t=(1:1:64)*0.12;
A_s=10;
A_n=.5;

for i=1:1:1024
   for j=1:1:1024
      phixy(i,j)=A_s*pi*exp(-((i-512)^2/10000+(j-512)^2/10000));
   end
end

for k=1:1:64
   noise_matrix=A_n*randn(1024);
   phase_3d(:,:,k)=angle(exp(imag_i*(phixy+(k-1)*.2*pi)))+noise_matrix;
end

phi_unwrap_3d = phase_unwrap_3d(phase_3d,3);

for i=1:1:64
   phi_unwrap_2d(:,:,i)= phase_unwrap(phase_3d(:,:,i));
end

figure
plot(phixy(:,512),'k--','LineWidth',3)
hold on
plot(phase_3d(:,512,1),'g','LineWidth',2)
plot(phi_unwrap_2d(:,512,1)-mean(mean(phi_unwrap_2d(1:100,1:100,1))),'b')
plot(phi_unwrap_3d(:,512,1)-
mean(mean(phi_unwrap_3d(1:100,1:100,1))),'r','LineWidth',1)
xlim([1 1024])
set(gca,'FontSize',14)
legend('true phase','wrapped phase','2D unwrap','3D unwrap')
xlabel('i','FontSize',24)
ylabel('phase(rad)','FontSize',24)

P_GT_C=phixy(512,512)+((1:1:64)-1)*0.2*pi;
P_3D_C=permute(phase_3d(512,512,:),[3 1 2]);
```

```matlab
P_2D_unwrap_C=permute(phi_unwrap_2d(512,512,:)-
mean(mean(phi_unwrap_2d(1:100,1:100,1))),[3 1 2]);
P_3D_unwrap_C=permute(phi_unwrap_3d(512,512,:)-
mean(mean(phi_unwrap_3d(1:100,1:100,1))),[3 1 2]);


figure
plot(P_GT_C,'k--','LineWidth',3)
hold on
plot(P_3D_C,'g','LineWidth',2)
hold on
plot(P_2D_unwrap_C,'b')
plot(P_3D_unwrap_C,'r','LineWidth',1)
legend('true phase','wrapped phase','2D unwrap','3D unwrap')
set(gca,'FontSize',14)
xlabel('k','FontSize',24)
ylabel('phase(rad)','FontSize',24)

figure('Position', [100 0 800 600]);
subplot(2,2,1), imagesc(phixy),hcb = colorbar;hcb.Title.String = "rad";
set(gca,'FontSize',12),xlabel('i','FontSize',16),ylabel('j','FontSize',16),title('true
phase','FontSize',18)
subplot(2,2,2), imagesc(phase_3d(:,:,1)),,hcb = colorbar;hcb.Title.String = "rad";
set(gca,'FontSize',12),xlabel('i','FontSize',16),ylabel('j','FontSize',16),title('wrapped
phase','FontSize',18)
subplot(2,2,3), imagesc(phi_unwrap_2d(:,:,1)-
mean(mean(phi_unwrap_2d(1:100,1:100,1)))),,hcb = colorbar;hcb.Title.String = "rad";
set(gca,'FontSize',12),xlabel('i','FontSize',16),ylabel('j','FontSize',16),title('2D unwrapped
phase','FontSize',18)
subplot(2,2,4), imagesc(phi_unwrap_3d(:,:,1)-
mean(mean(phi_unwrap_3d(1:100,1:100,1)))),,hcb = colorbar;hcb.Title.String = "rad";
set(gca,'FontSize',12),xlabel('i','FontSize',16),ylabel('j','FontSize',16),title('3D unwrapped
phase','FontSize',18)

for index_n=1:1:10
    index_n
    A_s=10;
    A_n=.1*index_n;

    for k=1:1:64
        noise_matrix=A_n*randn(1024);
        phase_3d(:,:,k)=angle(exp(imag_i*(phixy)))+noise_matrix;
    end


    phi_unwrap_3d = phase_unwrap_3d(phase_3d,3);
```

```
for i=1:1:64
    phi_unwrap_2d(:,:,i)= phase_unwrap(phase_3d(:,:,i));
end

DR=(100*.5);

peaksnr_2d(index_n) =psnr(phi_unwrap_2d(:,:,1)-
mean(mean(phi_unwrap_2d(1:100,1:100,1))),phixy)
peaksnr_3d(index_n) =psnr(phi_unwrap_3d(:,:,1)-
mean(mean(phi_unwrap_3d(1:100,1:100,1))),phixy)

v_2d(index_n)=ssim(phi_unwrap_2d(:,:,1)-
mean(mean(phi_unwrap_2d(1:100,1:100,1))),phixy,'DynamicRange',DR)
v_3d(index_n)=ssim(phi_unwrap_3d(:,:,1)-
mean(mean(phi_unwrap_3d(1:100,1:100,1))),phixy,'DynamicRange',DR)
end

figure
plot(.1*(1:1:10),peaksnr_2d,'bo','LineWidth',2,'MarkerSize',8)
hold on
plot(.1*(1:1:10),peaksnr_3d,'ro','LineWidth',2,'MarkerSize',8)
legend('2D unwrap','3D unwrap')
set(gca,'FontSize',12)
xlabel('\sigma_n(rad)','FontSize',24)
ylabel('PSNR','FontSize',24)


figure
plot(.1*(1:1:10),v_2d,'bo','LineWidth',2,'MarkerSize',8)
hold on
plot(.1*(1:1:10),v_3d,'ro','LineWidth',2,'MarkerSize',8)

legend('2D unwrap','3D unwrap')
set(gca,'FontSize',12)
xlabel('\sigma_n(rad)','FontSize',24)
ylabel('SSIM','FontSize',24)

A_s=10;
A_n=.5;

for i=1:1:1024
    for j=1:1:1024
        phixy(i,j)=A_s*pi*exp(-((i-512)^2/10000+(j-512)^2/10000));
    end
end
```

```matlab
for k=1:1:64
    noise_matrix=A_n*randn(1024);
    phase_3d(:,:,k)=angle(exp(imag_i*(phixy+(k-1)*.2*pi)))+noise_matrix;
end

phi_unwrap_3d = phase_unwrap_3d(phase_3d,3);

for i=1:1:64
    phi_unwrap_2d(:,:,i)= phase_unwrap(phase_3d(:,:,i));
end

figure('Position', [100 100 600 500]);

for frame_index=1:1:64

    frame_index

    subplot(2,2,2)
    set(gca,'FontSize',12)
    mesh((phase_3d(1:800,101:900,frame_index)));view([-32 54])
    zlim([-2*pi 25*pi])
    xlabel('i','FontSize',12),ylabel('i','FontSize',12),  zlabel('phase(rad)','FontSize',12),
    title(['wrapped phase k=' num2str(frame_index)],'FontSize',12),
    %view([136 46])

    subplot(2,2,1)
    set(gca,'FontSize',12)
    mesh(phixy(1:800,101:900)+(frame_index-1)*0.2*pi);view([-32 54])
    zlim([-2*pi 25*pi])
    xlabel('i','FontSize',12),ylabel('i','FontSize',12),  zlabel('phase(rad)','FontSize',12),
    title(['ground truth k=' num2str(frame_index)],'FontSize',12),

    subplot(2,2,3)
    set(gca,'FontSize',12)
    mesh(phi_unwrap_2d(1:800,101:900,frame_index)-
mean(mean(phi_unwrap_2d(1:100,1:100,1))));view([-32 54])
    zlim([-2*pi 25*pi])
    xlabel('i','FontSize',12),ylabel('i','FontSize',12),  zlabel('phase(rad)','FontSize',12),
    title(['2D unwrapped phase k=' num2str(frame_index)],'FontSize',12),

    subplot(2,2,4)
    set(gca,'FontSize',12)
    mesh(phi_unwrap_3d(1:800,101:900,frame_index)-
mean(mean(phi_unwrap_3d(1:100,1:100,1))));view([-32 54])
    zlim([-2*pi 25*pi])
```

```matlab
xlabel('i','FontSize',12),ylabel('i','FontSize',12),  zlabel('phase(rad)','FontSize',12),
title(['3D unwrapped phase k=' num2str(frame_index)],'FontSize',12),

drawnow

F(frame_index) = getframe(gcf) ;

end

writerObj = VideoWriter('3d_2d_simu_0117');
writerObj.FrameRate = 9;
% set the seconds per image
% open the video writer
open(writerObj);
for i=1:length(F)
   % convert the image to a frame
   frame = F(i) ;
   writeVideo(writerObj, frame);
   %return
end

close(writerObj);
```

REFERENCES

Y. Wang, Q. Kang, Y. Zhang, and X. Liu, (2020). "Optically computed optical coherence tomography for volumetric imaging," Optics letters 45, 1675-1678.

W. Zhang, X. Zhang, C. Wang, W. Liao, S. Ai, J. Hsieh, N. Zhang, and P. Xue, (2019). "Optical computing optical coherence tomography with conjugate suppression by dispersion," Optics Letters 44, 2077-2080.

Y. Park, C. Depeursinge, and G. Popescu, (2018). "Quantitative phase imaging in biomedicine," Nature photonics 12, 578-589.

T. H. Nguyen, C. Edwards, L. L. Goddard, and G. Popescu, (2014). "Quantitative phase imaging with partially coherent illumination," Optics Letters 39, 5511-5514.

N. Lue, W. Choi, G. Popescu, T. Ikeda, R. R. Dasari, K. Badizadegan, and M. S. Feld, (2007). "Quantitative phase imaging of live cells using fast Fourier phase microscopy," Applied Optics 46, 1836-1842.

B. Vakoc, S.-H. Yun, J. De Boer, G. Tearney, and B. Bouma, (2005). "Phase-resolved optical frequency domain imaging," Optics express 13, 5483-5493.

Y. Zhao, Z. Chen, C. Saxer, S. Xiang, J. F. de Boer, and J. S. Nelson, (2000). "Phase-resolved optical coherence tomography and optical Doppler tomography for imaging blood flow in human skin with fast scanning speed and high velocity sensitivity," Optics letters 25, 114-116.

D. Huang, E. A. Swanson, C. P. Lin, J. S. Schuman, W. G. Stinson, W. Chang, M. R. Hee, T. Flotte, K. Gregory, and C. A. Puliafito, (1991). "Optical coherence tomography," science 254, 1178-1181.