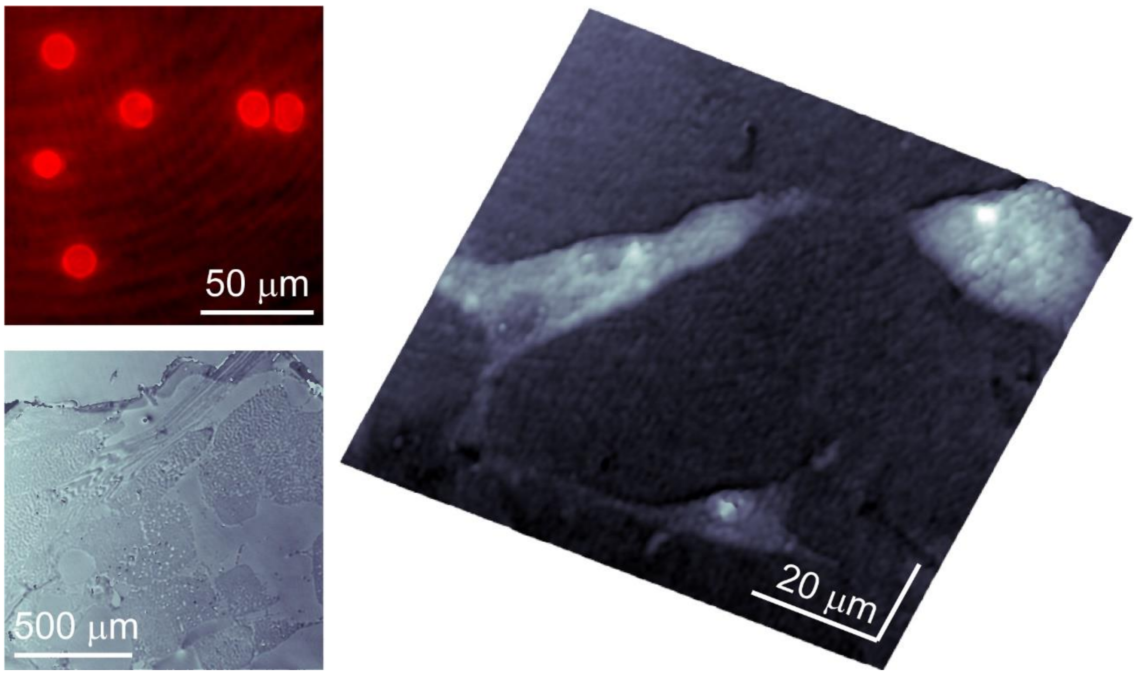
User Manual for The Universal DHM (tuDHM) reconstruction algorithm



The Universal DHM reconstruction algorithm (tuDHM) – Version 1.0

Raúl Castañeda and Ana Doblas

October 2021

Contact: rcstdqnt@memphis.edu, adoblas@memphis.edu, opticalimagingresearchlab@gmail.com

OIRL, Department of Electrical and Computer Engineering, University of Memphis, Memphis T.N.

Content

1. About tuDHM

tuDHM is a Matlab implementation developed by the Optical Imaging Research Laboratory from the Department of Electrical and Computer Engineering at the University of Memphis. tuDHM App is a free user-friendly tool, providing a universal strategy for reconstructing in telecentric, single-shot digital holographic microscopy (DHM). This implementation is based on a fast and automatic method based on the minimization of a cost function that finds the best numerical conjugated reference beam to compensate the filtered object information, eliminating any undesired phase perturbation due to the tilt between the reference and object waves. The novelties of the proposed approach are accurate reconstructed phase images, and reduced processing time.

1.1 Credits

- tuDHM is developed in MATLAB. (2020), version 7.10.0 (R2020a), Natick, Massachusetts: The MathWorks Inc.

- For the unwrapping step, tuDHM implements the code developed by Herráez *et.al*. [1,2]

1.2 Citation

If using tuDHM for publication, please kindly cite the following:

- R. Castaneda and A. Doblas, "Fast-iterative automatic reconstruction method for quantitative phase images with reduced phase perturbations in off-axis digital holographic microscopy," Applied Optics xx, xxx (2021).

1.3 Support

If you use tuDHM and find a bug, please contact us via email and we will address the problem. Our emails are opticalimagingresearchlab@gmail.com, rcstdqnt@memphis.edu, and adoblas@memphis.edu.

2. Running tuDHM

Currently, there are two implementations of tuDHM. The first one is the raw MATLAB code and the second one corresponds to a MATLAB GUI user-friendly version.

2.1 Raw version

This version is addressed to developers or imaging researchers with MATLAB knowledge. The version is downloaded from https://oirl.github.io/tuDHM/. The file downloaded **tuDHM.zip** should be unzip, the folder contains seven files, where the main function is *tuDHM.m.* To operate it, the function *tuDHM.m* must be open with MATLAB.

2.1.1 Read the hologram

For read the hologram in the file *tuDHM*.m the code line number 48 contain the variable filename, the user should punt inside ‘ ’, the name of the hologram with the extension of the image. The hologram must be in the same folder where was unzip **tuDHM.zip**. Below is the snapshot

```
%% load hologram
filename = 'holotele10x.tif'; %insert the name of the hologram to be reconstructed
```

2.1.2 Input parameters

The input parameters correspond to the parameters used for recorded the hologram: [**lambda**, and **dxy**].

- **Lambda→** Corresponds to the wavelength used for record the hologram, in micrometers.
- **dxy→** This are the dimension of the pixel size of the digital camera used to record the hologram, in micrometers.

These parameters are found in *tuDHM.m* code line numbers 53 and 54. Below is the snapshot

```
%% parameters of reconstruction (this parameters depend of the hologram recorded)
lambda = 0.532;% source's wavelength in microns
dxy = 3.75;% size of pixel in microns along the vertical and horizontal direction
```

Once the input parameters have been set, the algorithm could be run. There are two functions that allow to modify a set of arguments *holo_filter* and *phase_reconstruction*.

- In the file *tuDHM*.m the code line number 58
  The fourth argument of the function *holo_filter* is for active or deactivate the visualization of the FFT of the hologram and the spatial filter, use ‘yes’ or ‘not’, for the visualization available or not available, respectively.

```
%% spatial frequency filter
%filter the +1 term from the Fourier transform of the hologram
[holo_filter,fx_max,fy_max] = holo_filter(holo,M,N,'not');
```

- In the file *tuDHM*.m the code line number 60-63
  We define the cost function *J* as a function of the values of ($f_{x,max}$, $f_{y,max}$), the source’s wavelength (lambda), the pixel size (dxy), the size of the hologram (M and N), and the filtered hologram. The arrays *m* and *n* provide the meshgrid based on the hologram size to compute the reference wave. The estimation of the cost function requires the binarization of the reconstructed phase image. The binary image is obtained by applying the MATLAB built-in function *imbinarize,* which uses Otsu's method, and its input parameter is the reconstructed phase image.

```
%% cost function
% Cost function based on the Eq. (8)
seed_maxPeak = [fx_max - 1,fy_max - 1];
J = cost_function(seed_maxPeak,lambda,dxy,M,N,m,n,holo_filter);
```

- In the file *tuDHM*.m, the code line number 65-75
  It is found the code for minimizing the cost function. In line 66, we define the optimization parameters (maximum number of iterations) and the tolerance. The minimization of the cost function is perfomed using the MATLAB built-in function *fminunc* (line 67-68). The *fminunc* function finds a local minimum of an unconstrained multivariable function using a quasi-Newton algorithm. The input parameters of the *fminunc* function are the cost function, the integer position of the maximum peak in the filtered hologram spectrum, ($f_{x,max}$, $f_{y,max}$) , and the set of optimizations options defined in line 66. The output of the *fminunc* function is the optimal value of ($f_{x,max}$, $f_{y,max}$), and the final cost function.

```
%% minimization
options = optimset('Display','iter', 'MaxIter', 30,'TolX',1e-3);
[MaxPeaks, J] = ...
    fminunc(@(t)(cost_function(t,lambda,dxy,M,N,m,n,holo_filter)), seed_maxPeak, options);

fx_max_best = MaxPeaks(1,1);
fy_max_best = MaxPeaks(1,2);

fprintf('Cost at peaks found by fminunc: %f\n', J);
fprintf('fx: %f\n', fx_max_best);
fprintf('fy: %f\n', fy_max_best);
```

- In the file *tuDHM*.m (lines 77-80), we find the reconstruction of the phase images using the optimal ($f_{x,max}$, $f_{y,max}$) obtained after minimizing the cost function. Firstly, we estimate the optimal reference wave (line 78) via the user-defined function *reference_wave*, and later we reconstruct the phase map via the user-defined function *phase_reconstruction.* The third argument of the function *phase_reconstruction* is for active or deactivate the unwrapping step. Use 'yes' or 'not', for the available or not available, respectively. If unwrapping is used, we use the method implemented via Kasim [2].

```
%% Reconstruction, best phase distribution
[ref_wave] = reference_wave(M,N,m,n,lambda,dxy,fx_max_best,fy_max_best);
[phase] = phase_reconstruction(holo_filter,ref_wave,'not');
```

2.2 GUI version

This version contains a GUI friendly that is focus for anyone that required obtain the numerical phase reconstruction of holograms without knowledge of MATLAB.

2.2.1 Installation

To install tuDHM GUI version, download the .exe from the webpage https://oirl.github.io/tuDHM/ and execute it by clicking.

2.2.2 Running tuDHM GUI version

We have divided the graphical interface in eight panels, Figure 1: (1) selection of the hologram to be reconstruct by the user; (2) parameters of the recorded hologram introduced by the user; (3) central frequency of the spectrum of the real image, which is automatically estimated; (4) unwrapping step; (5) save output option in which the user can select the reconstructed amplitude and phase maps without and with applying unwrapping; (6) reconstruction button; (7) visualizations of the hologram and its spectrum as well as the filtered object spectrum and the raw phase image without optimal compensation of the reference wave. Panel (8) shows the final reconstructed images to be chosen between the reconstructed amplitude and phase maps without and with applying unwrapping.

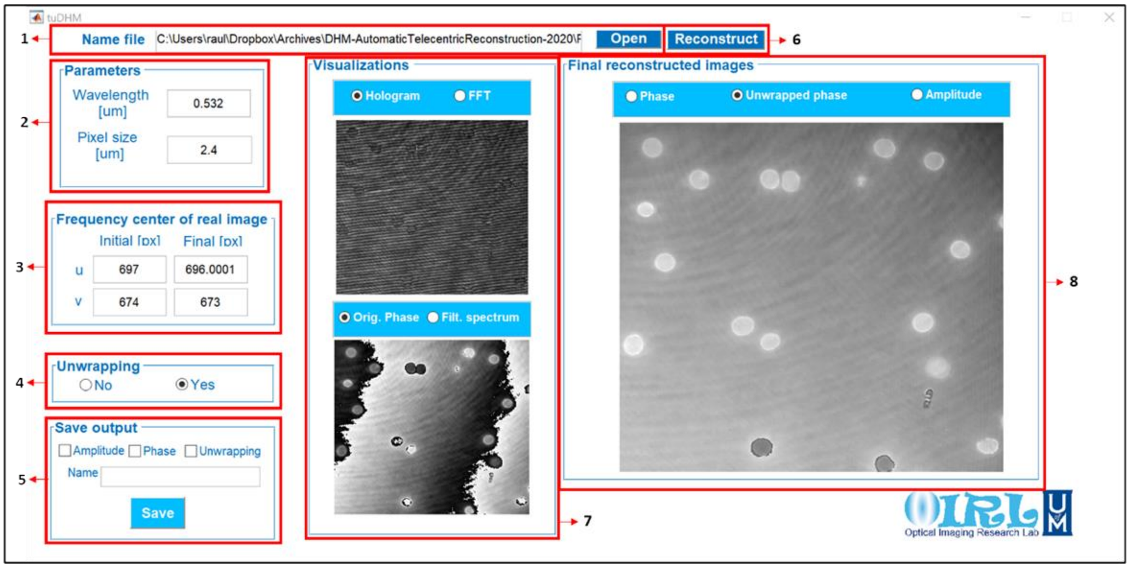


Figure 1. Snapshot of the tuDHM GUI version.

1. Choose hologram: This panel contains the button **open** that allows opens a dialog box. It enables the user to select a hologram in the local computer. Once the hologram is open, the directory folder appears in the label text (Name file).

2. Parameters: In this panel, the user must insert the values of the parameters used to record the hologram: *Wavelength*, which is the source's wavelength in microns (µm), and *Pixel size*, which is the size of the square pixel in microns (µm).

3. Central Frequency [px] of the spectrum of the real image: This panel displays the values found for positions of the maximum peak in the spectrum of the real image (e.g., position of the +1 term) before (e.g., integer position found as initial/seed value) and after applying the proposed method. (e.g., the optimal non-integer position obtained by our approach).

4. Unwrapping: The user can select the estimation and display of the unwrapped phase image for samples in which wrapping is observed. The unwrapping approach is performed through the method proposed by Herráez *et. al.* [1,2] and implemented by Kasim [2].

5. Save results: In this panel, the user has the option to save the results shown in the reconstruction panel as a figure .jpg, and .mat. The user can save the amplitude, wrapped phase and the unwrapped phase images by selecting the corresponding checkboxes (amplitude, phase, and unwrapping, respectively). In the text label name, the user can type the name of the saved images. The figure files will be named xxx_amplitude, xxxx_phase,and xxxx_unwrapped, where xxxx is the name typed on the text label name. The save button is to enable the saving of the figures generated by the application of the proposed automatic approach. The images will be saved in the same folder where was unzip **tuDHM.zip.**

6. Reconstruct: This panel contains the buttons Reconstruct. By pressing the *Reconstruct* Button, the user initializes the numerical phase reconstruction approach.

7. Visualizations: This section contains two panels. At the top panel, one can choose the display of the raw hologram or its Fourier transform. At the bottom panel, one can select the display of the reconstructed phase image when the reference compensation is done with the integer position of the maximum peak found in the spectrum of the real image, or the spectrum of the real image obtained by filtering the frequencies of the real image from the hologram spectrum (the last display is to verify that the filtering of the +1 term has been performed properly).

8. Final reconstructed image: In this panel, the user can select the visualization of the reconstructed wrapped phase image (phases values from -π to π), the unwrapped phase image, or the reconstructed amplitude image. The unwrapped phase image is only available if the user selects the option *yes* in the unwrapping panel.

3. Usage example

We have available two holograms with the respective parameter values in https://oirl.github.io/tuDHM/. The holograms correspond to samples of red blood cells and a transverse section of the thorax of a *Drosophila melanogaster* fly.

The parameters (wavelength, and pixel size) are: (0.532 µm, 2.4 µm) for the red blood cells, and (0.632 µm, 6.9 µm) for the *Drosophila melanogaster* fly.

Below is a step-by-step procedure for using tuDHM GUI version.

1. Open the digital hologram image file clicking the button open.
2. Introduce the input parameters
3. Chose if desire obtain the unwrapping phase reconstruction, by defect this option is unenabled.
4. Chose the output that desire saving and input the name
5. Clink on reconstruct button.
6. Select the representations of the output field

4. References

[1] M. A. Herraez, D. R. Burton, M. J. Lalor, and M. A. Gdeisat, "Fast two-dimensional phase-unwrapping algorithm based on sorting by reliability following a noncontinuous path", Applied Optics, Vol. 41, Issue 35, pp. 7437-7444 (2002).

[2] M. F. Kasim, "Fast 2D phase unwrapping implementation in MATLAB", available in https://github.com/mfkasim91/unwrap_phase/.