# LENS-LESS MICROSCOPY USING RASPBERRY PI

BTP-II report submitted to
Indian Institute of Technology Kharagpur
In partial fulfilment for the award of the degree of
Bachelor of Technology
In
Electronics and Electrical Communication Engineering

By

Sudarshan Biswas
(18EC35052)

Under the supervision of
Dr. Sarang Pendharker

Department of Electronics and Electrical Communication Engineering

Indian Institute of Technology Kharagpur

Spring Semester 2021-22

April 2022

DECLARATION

I certify that

I. The work contained in this report is original and has been done by me under the guidance of my supervisor.

II. The work has not been submitted to any other Institute for any degree or diploma.

III. I have followed the guidelines provided by the Institute in preparing the report.

IV. I have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.

V. Whenever I have used materials (data, theoretical analysis, figures, and text) from other sources, I have given due credit to them by citing them in the text of the thesis and giving their details in the references. Further, I have taken permission from the copyright owners of the sources, whenever necessary.

Sudarshan Biswas

# CERTIFICATE

This is to certify that the report entitled, "**Lensless Microscopy using Raspberry Pi**" submitted by **Sudarshan Biswas (18EC35052)** to the Indian Institute of Technology Kharagpur, India, is a record of bonafide project work carried out by him under my supervision and guidance towards partial fulfilment of requirements for the award of degree of Bachelor of Technology in Electronics and Electrical Communication Engineering.

Date: 23/4/2022 Signature of the Supervisor

(Dr. Sarang Pendharker)

# Contents

# Chapter 1

# Introduction

Lens-free digital in-line holographic microscopy is a fast developing computational imaging method that enables the building of exceedingly compact and high-throughput microscopes. It is made possible by the consumer electronics industry's ongoing developments and improvements in microscopy and image reconstruction methods, as well as image sensor technology and processing capacity. Its current implementations can achieve giga-pixel level space-bandwidth-products, by employing cost-effective and field-portable imaging hardware.

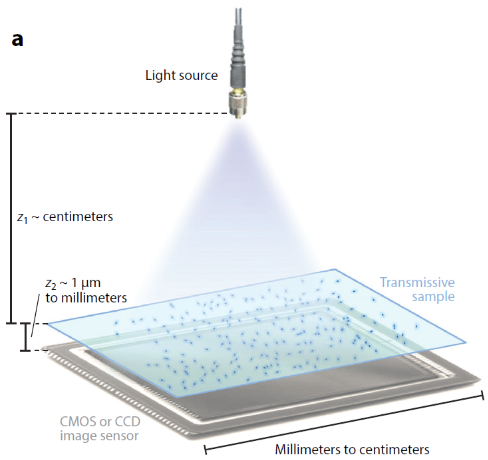



Figure 1.1: On-Chip Lensless Microscopy[1]

## 1.1 Imaging Techniques

Lens-free imaging techniques[1] can be mainly categorised into 2 types (i)contact based shadow imaging (ii) diffraction based imaging.

### 1.1.1 Contact based Shadow Imaging

When the distance between the sample and sensor array is less than 1µm hence diffraction is significantly reduced and they effectively capture the shadow of objects. Images obtained in this approach are not as precise as in the (ii) method but it can be

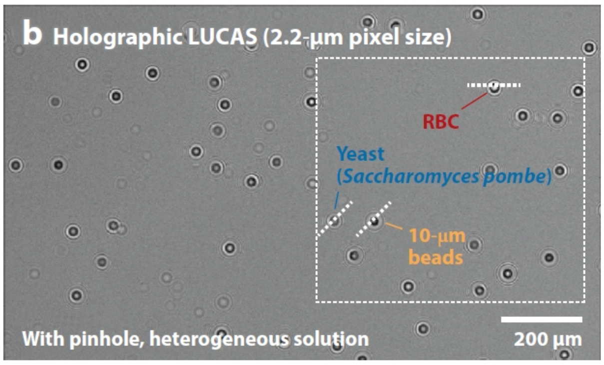



Figure 1.2: Lucas Shadow imaging of RBCs, yeast cells and 10µm beads[1]

useful in application where high precision images are not required as in the case of LUCAS (Fig 1.2) where the count of objects is the primary objective.

### 1.1.2 Diffraction Based Imaging

In the case of diffraction based imaging, the scattered light from each object interferes with itself and with the un-scattered background to form an interference pattern which needs to be digitally reconstructed to obtain the objects. My work mainly focuses on diffraction based imaging.

In order to keep the imaging setup as compact as possible, the on- chip holographic image acquisition platform employs an in-line holography geometry, where the scattered object field and the un-scattered reference beam co-propagate in the same direction, and the intensity of the interference pattern between these two beams is recorded by a digital image sensor-array(Fig 1.1).

Lens-free imaging is a viable option for any investigation that requires statistically meaningful population estimates based on the analysis of a small sample. A complete blood count and the study of hundreds of sperm trajectories to find uncommon forms of motion are two examples[5]. Imaging techniques with a large space–bandwidth product and depth of focus enhance tests developed for samples with huge ranges in number density (concentration), as it is possible to detect a few objects or many thousands to millions of objects in a single image. Tests of air and water quality are two instances of tests where the numerical density might vary significantly.

## 1.2 Digital In-Line Holography

Holography is a two-step process. First a hologram must be recorded and second, the reconstruction must yield an image of the object i.e. the intensity of the wavefront of the object. In DIHM(Digital in-line holographic microscopy) the hologram is detected by the CMOS sensor array and is transferred to the computer to be numerically reconstructed.

When the aperture of the light source is sufficiently small and monochromatic, the optical field impinging on the sample plane can be considered coherent, and the recorded shadows exhibit interference fringe patterns[1]. From these interference fringes, one can infer the optical phase of the scattered light, making it possible to reconstruct the optical field that is transmitted through the objects on the sample and thereby providing an in-focus "image" of the sample. The interference fringe pattern recorded on the sensor is an inline hologram, which is the intensity pattern generated by the interference between light scattered from an object on the sample and a reference wave that passes undisturbed through the transparent substrate. At the plane where this light interacts with the sample, we can describe its electric field as the sum of the reference wave at the sample, $E_{R,S}$ and the object wave at the sample $E_{O,S}$.

$$E = E_{R,S} + E_{O,S} = A_R + A_O(x_S, y_S)e^{i\phi(x_i,y_i)} \tag{1.1}$$

$A_R$ is the amplitude of the spatially uniform plane reference wave, $A_O(x_s,y_s)$ is the spatially varying amplitude (transmittance) of the object. $\varphi(x_s,y_s)$ is the spatially varying phase (optical thickness) of the object. The goal of digital holographic reconstruction is to recover $A_O$ and $\varphi_O$ on the basis of measurement(s) of the light intensity further downstream at the image sensor plane. From these downstream measurements, it is possible to reconstruct the amplitude and phase of the object by using the angular spectrum approach.

### 1.2.1 Angular Spectrum Approach

This approach consists of computing the Fourier transform of the captured hologram, multiplying it by the transfer function of free space, and then inverse Fourier transforming.

$$E_r = \mathscr{F}^{-1}[\mathscr{F}[E_i(x,y)]H(f_x,f_y)] \tag{1.2}$$

Er is the reconstructed optical field of the object. $E_i(x,y)$is the captured hologram $H(f_x, f_y)$is the transfer function of free space ($n = 1$):

$$H(f_x,f_y) = \begin{cases} e^{ikz\sqrt{1-(2\pi f_x/k)^2-(2\pi f_y/k)^2}} & f_x^2+f_y^2 < \frac{1}{\lambda^2} \\ 0 & f_x^2+f_y^2 \geq \frac{1}{\lambda^2} \end{cases} \tag{1.3}$$

Here, $\lambda$ is the wavelength of the light, $k = 2\pi/\lambda$, $f_x$ and $f_y$ are spatial frequencies, and z is the same sample-to-sensor distance shown in Fig 1.1. Often z may not be precisely known before the capture of the hologram, and some computational “refocusing” can be performed to estimate it and provide the sharpest reconstructions.

To reconstruct the object using this approach we need $A_R$ which can be obtained by recording the CMOS data without the slide which will give the amplitude of the reference wave. We can also get the $A_o(x_s,y_s)$ from the intensity values observed by the CMOS sensor when we use the slide. The other parameters such as λ ,z can be obtained from measurements of the experimental setup. The only unknown required to reconstruct the object is the phase information at sensor plane $\varphi(x_s,y_s)$. Since the sensor can only capture the intensity of light we cant obtain the phase information directly. We need to take multiple images by varying one or multiple parameters such as the distance between sensor plane and object plane, angle of laser source, distance between object plane and source.

My work revolves around retrieving the phase information from the recorded intensity patterns and after getting the required phase information, reconstructing the image. In the following chapter I first discuss the previously attempted method of Phase retrieval and then, the currently implemented one.

# Chapter 2

# Phase retrieval algorithms

As mentioned in the previous chapter, retrieving phase information plays a very big part in proper reconstruction of the object sample. Without the phase information we get visual artefacts like twin-images[5] and rippling. One way to preserve phase information in using off-axis holography but, since we are limited by the amount of space to dedicate to the setup with the Raspberry Pi, In-Line holography is the way to go. Hence, the immediate need for phase retrieval algorithms. One of the earlier attempts I made was using TIE to find the phase.The following section briefly explains the solution.

## 2.1 Universal Solution to Transfer of Intensity Equation

In 1982, Teague first established the quantitative relationship between the longitudinal intensity variation and phase of a coherent beam with use of a second-order elliptic partial differential equation, so called TIE[2].

$$-k\frac{\partial I(r)}{\partial z} = \nabla\dot{[}I(r)\nabla\phi(r)] \tag{2.1}$$

Now, assuming that the in-focus intensity is uniform (I'), it can be pulled directly out of the gradient operator to give something which directly boils down to a Poisson Equation:

$$-k\frac{\partial I(r)}{\partial z} = I'\nabla^2\phi(r) \tag{2.2}$$

Following that, if we secondly assume that the in-focus image intensity is is uniform with a constant value $I_{max}$ where $I_{max}$ is the maximum value of the in-focus image intensity. Then, the solution of the phase takes the following form:

$$\phi(r) = \frac{-k}{I_{max}}\nabla^{-2}[\frac{\partial I(r)}{\partial z}] \tag{2.3}$$

But it should be noted that the maximum intensity assumption $I_{max}$ used here is obviously "unreasonable," and the phase

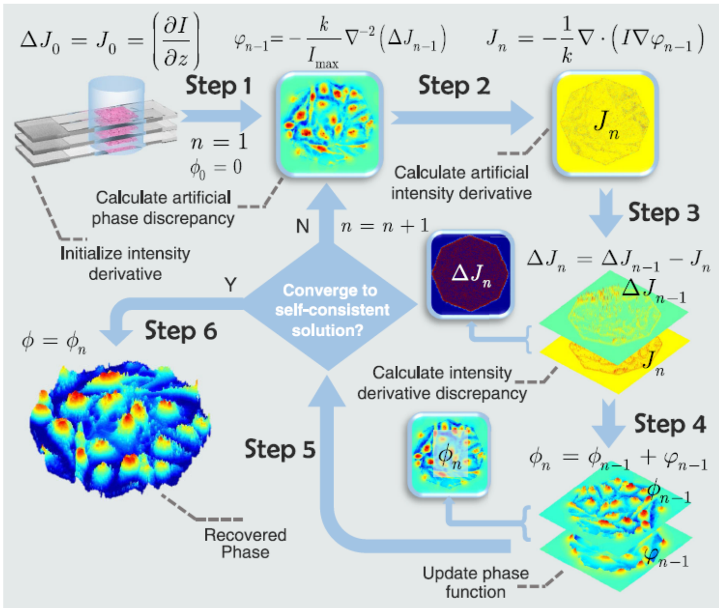



Fig 2.1: Flow chart of the US-TIE method[2]

calculated by Eq. (2.3) is usually inaccurate. Thus, the next step is to treat the inaccurate phase $\varphi_0(r)$ as an initial solution, and substitute it back to Eq. (2.1). The inaccurate solution results in inconsistency between the calculated intensity derivative $J_n$ and the real measurement value, which is treated as the error signal for another round of phase reconstruction. The solution $\varphi_0(r)$ is also taken as the "correction term," which is added back to $\varphi_0(r)$ to get an updated phase estimate. This completes one iteration of the reconstruction algorithm, and the procedure is iteratively repeated until convergence as shown in Fig. 2.1.

The approach is already proven to work quite well[2] but the problem with this is the implementation. With the cost-effective setup I plan on building, there seems to be no low-cost way/ compact way of having a setup in which the sample plane can be moved up or down by a few micrometers. Hence, an alternative approach is to be found. One of the better ways to do the is by using different illumination wavelengths by using a tuneable laser or by using angled reflection surfaces to change the laser wavelengths. In the following section I discuss phase retrieval through different illumination wavelengths.

## 2.2 Phase retrieval using double sequence intensity patterns (PRDS)

This method[3] works by iterative front and back propagation of diffraction patterns recorded at different intensities and updating the object surface profile each time.

### 2.2.1 PRDS iterative algorithm

First, an intensity stack is made by illuminating the sample with differing wavelengths. Then, the stack is divided into two groups t and q where t = 1,2,..T and q = 1,2,.....Q with wavelengths $\lambda_t$ and $\lambda_q$. The same pattern can be used twice as long as same wavelength's intensity pattern is not used by both the groups in the same iteration cycle.The iterative steps are mentioned below:

(1) Recreate the two wavefronts by using the square root of the recorded intensities (amplitude) with the calculated phases.

$$U_t = \sqrt{I_t} e^{i\phi_t(x,y)} \tag{2.4}$$

$$U_q = \sqrt{I_q} e^{i\phi_q(x,y)} \tag{2.5}$$

If it is the first iteration, two initial guessed phases are used.

(2) Back-propagate these two wavefronts to the object plane.

(3) Calculate the object surface profile using the obtained back-propagated phases

$$h(x', y') = \frac{\Lambda}{n_{obj} - n_{air}} \frac{| \phi_t(x', y') - \phi_q(x', y') |}{2\pi}, \Lambda = \frac{\lambda_t \lambda_q}{| \lambda_t - \lambda_q |}, \lambda_t \neq \lambda_q \quad (2.6)$$

where h is the object profile recalculated at every iteration and x',y' represent the coordinated in the object plane. Λ is the synthetic wavelength.

(4) Combine the average of the back-propagated square rooted intensities of the wavefront and phases at a different wavelength using the object profile calculated in the above iteration.

$$U_{t+1} = \frac{\sqrt{I_t(x', y')} + \sqrt{I_q(x', y')}}{2} exp(2\pi i \frac{h(x', y')}{\lambda_{t+1}}) \quad (2.7)$$

$$U_{q+1} = \frac{\sqrt{I_t(x', y')} + \sqrt{I_q(x', y')}}{2} exp(2\pi i \frac{h(x', y')}{\lambda_{q+1}}) \quad (2.8)$$

(5) Front-propagate the calculated wavefronts in (4).

(6) Repeat steps (1) till (5)

The iterations end when either the mean-square error of the obtained wavefront is below a pre-determined value or the maximum number of iterations is reached.

This second approach is very appropriate for my implementation because in this case, there are no moving parts. The wavelength can be tuned to get about 10-20 different images and then they can be used to obtain the phase. In the next chapter I discuss my attempts at phase retrieval using PRDS.

# Chapter 3

# PRDS implementation using MATLAB

PRDS or Phase Retrieval using Double Sequence is used to get the phase information through the iterative process mentioned in 2.2.1. There are various stages to this, the first is making the object to be illuminated; following that, using ASP (Angular Spectrum) approach to propagate/back-propagate the object wavefront and finally working on PRDS algorithm to obtain the phase information.

## 3.1 Designing the Sample Object Wavefront

The intensity of the sample object was made using Procreate® as shown in Fig 3.1(a). For the phase, the scaled down version of the intensity plot was used by multiplying it with $2\pi/256$. A problem which shows itself when the phase is zero for a considerable amount of space in the object is getting spurious phases in reconstruction. This happens because there is no way to calculate the phase difference between two zero valued functions. To prevent this problem, a radial function was added to the phase and intensity whose maximum value is at the centre and gradually becomes zero near the edges to make sure the zero phase problem doesn't occur. Additionally, the phase was blurred using a Gaussian filter to take care of any sharp gradients. Fig 3.1(b) and Fig 3.1(c) show the intensity and phase of the sample object.

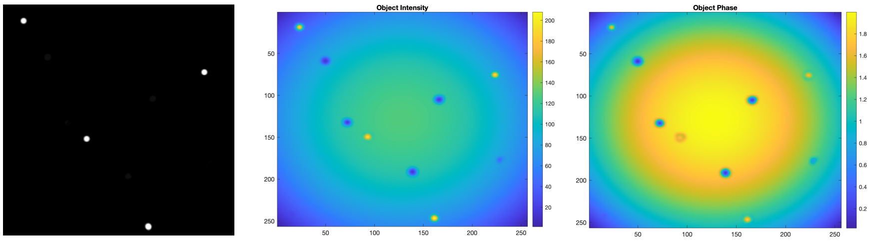


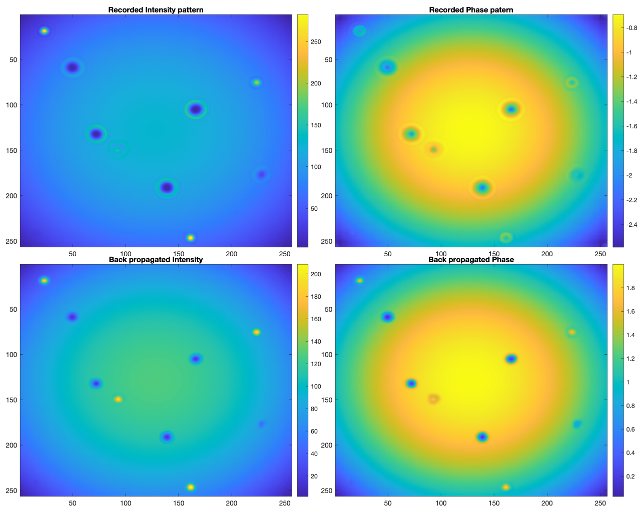

Fig 3.1: (a)Source Image (b)Object Intensity (c)Object Phase

## 3.2 Front and Back Propagation Functions

For the front propagation ASP was used as mentioned in section 1.2. As for the back propagation, the transfer function is the complex conjugate of H in Eq(1.3)

$$E_i = \mathscr{F}^{-1}[\mathscr{F}[E_r(x, y)]H^*(f_x, f_y)] \tag{3.1}$$

Where $E_i$ is the reconstructed object wavefront and $E_r$ is the recorded object wavefront. To check the sanctity of the numerical front and back propagation functions, the wavefront




from Fig 3.1 was used to do both. Fig 3.2 shows that the numerical propagation functions are very accurate.

Fig 3.2: (a)Front Propagated Intensity (b)Front Propagated Intensity
(c)Back Propagated Intensity (d) Back Propagated Phase

# 3.3 PRDS

## 3.3.1 Setup

First the intensity stack needs to be made. For that the wavelengths of light used were 800nm, 805nm, 810nm,......850nm to create 11 different intensity patterns.Then, the grouping was done such that $\lambda_t$ = 800, 805,....845nm and $\lambda_q$ = 805, 810,....850nm. This way, the two sequences have a constant wavelength difference of 5nm. The initial phase assumption was made to be the initial two intensity images.Following that, the algorithm in 2.2.1 was run for about 50 iterations; the front and back propagation functions used were the ones mentioned in 3.2.

## 3.3.2 Results

The retrieved phase is shown in Fig 3.3(a) the Mean-squared error is also shown in Fig 3.3(b).

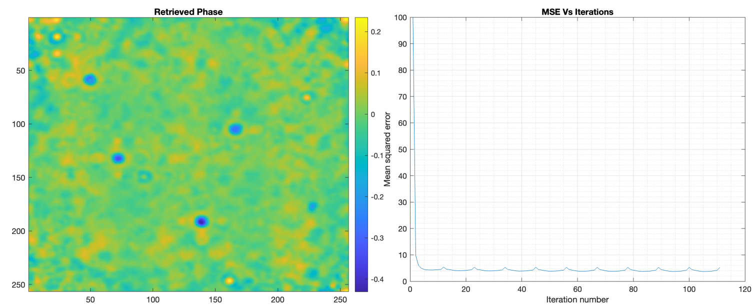



Fig 3.3: (a)Retrieved Phase using PRDS (b) MSE vs Iterations

The results are reminiscent of Contrast-phase imaging. Because while the algorithm doesn't seem to be able to recognise gradual shifts in phase, it is more than capable of detecting sudden changes in phase. In Fig 3.3(a) it can be noticed that while the gradual phase information when compared to Fig 3.1(c) is lost, the points of interest are retained. The MSE is also a very good measure of the correctness of the phase information. For example, when an object with a very gradual phase shift is illuminated, the phase image obtained might be quite deviated from the actual object profile and the MSE will provide that information. The initial guessed phase doesn't seem to affect the iteration number much, this is a very good outcome. When the initial guessed phase is set to be all zeroes or fully random noise, the retrieved phase is the same.

# Chapter 4

# Conclusion and Summary

The work is about figuring out a way to carry out lensless imaging on a Raspberry Pi. The first hurdle is reconstructing the image with just the image intensity diffraction patterns. To assuage the problem, first TIE equation approach was considered, while it does yield the required phase information, the physical implementation of it in a cost-effective/ compact way seems unfeasible. Hence ,an alternate approach was to be considered. PRDS algorithm was thus considered to correctly estimate the phase. While it does fail to catch gradual phase shifts, it was observed to give excellent phase-contrast imagery. The fact that the algorithm also calculates the correctness of the measured phase helps in knowing whether the estimated phase is accurate enough for consideration or not.

Future work shall focus on physically implementing the setup on a Raspberry Pi. While there are some physical constraints which need to be worked upon, the software part of the setup is done. The code is written in MATLAB which can be easily ported to Python using matplotlib. And since RPi natively supports GPIO control through Python code, transferring the code to a physical setup is very easy.

# References

[1]Aydogan Ozcan and Euan McLeod, "Lensless Imaging and Sensing," Annual Review of Biomedical Engineering, 2016

[2]J. Zhang, Q. Chen, J. Sun, L. Tian, and C. Zuo, "On a universal solution to the transport of intensity equation," Optics Letters, 2020.

[3]Bao, Peng & Pedrini, Giancarlo & Osten, Wolfgang, "Optical surface profile measurement using phase retrieval by tuning the illumination wavelength", Optics Communications, 2012.

[4]Yan Li, Wen Xiao, Feng Pan,"Phase retrieval method from multiple-wavelength holograms for in-line holography", Optik,2016.

[5]Bao, Peng & Zhang, Fucai & Pedrini, Giancarlo & Osten, Wolfgang, "Phase retrieval using multiple illumination wavelengths", Optics letters, 2008.

[6]Y. Rivenson, Y. Wu, H. Wang, Y. Zhang, A. Feizi, and A. Ozcan, "Sparsity based multiheight phase recovery in holographic microscopy," Scientific reports, 2016.

[7]Chao Zuo, Jiaji Li, Jiasong Sun, Yao Fan, Jialin Zhang, Linpeng Lu, Runnan Zhang, Bowen Wang, Lei Huang, Qian Chen, "Transport of intensity equation: a tutorial", Optics and Lasers in Engineering, 2020.

[8]GitHub Repository with all the worked upon code and references.