University of South-Eastern Norway
Faculty of Technology, Natural Sciences and Maritime Sciences
–
Master's Thesis
Study program: Master of Science in Micro and Nano Systems Technology
Spring 2023

Huy Duong Gia

High-resolution Fourier Ptychography Microscopy using ultra-violet light and infrared light.

University of South-Eastern Norway
Faculty of Technology, Natural Sciences and Maritime Sciences
Department of Microsystems.
Raveien 215
NO-3184 Borre, Norway

http://www.usn.no

Summary

This thesis focuses on implementing the Fourier Ptychography Microscopy (FPM) technique using Ultra-violet (UV) and infrared (IR) light.

In this report, the introduction provides a brief overview of FPM, its history, and the motivation for using UV and IR light.

Chapter 2 explains the foundational knowledge required for understanding the FPM algorithm, including plane waves, object light fields, and imaging system.

Chapter 3 presents a detailed description of the FPM algorithm, and the FPM model is presented in detail. This chapter also provides an in-depth analysis of each step involved in the FPM algorithm to recover a high-resolution image, including the algorithms to correct uncertainties in the practical setup and reduce background noise.

In Chapter 4, all methods employed in this thesis to implement FPM with UV and IR light are presented. This chapter describes our software, firmware, and hardware, which are combined to make the system work well. The chapter also includes the proposed recovery process. The samples that are used in experiments are described as well.

Chapter 5 presents the results of implementing FPM with UV and IR light, followed by a comprehensive discussion of the outcomes.

Chapter 6 discusses the challenges encountered during the implementation of FPM with UV and IR light. Chapter 6 also offers recommendations for future research.

Finally, Chapter 7 summarizes the thesis and its findings, providing suggestions for future research.

In the Appendix, we present the conference paper that was submitted during thesis work, the custom drive electronics design, the firmware coding, the software coding, and the recovery process programming, which are used to implement FPM with UV and IR light.

Overall, this thesis offers valuable insights into the potential applications of FPM with UV and IR light and provides a foundation for future research in this area.

Preface

I would like to begin by expressing my deepest gratitude to my supervisor, Muhammad Nadeem Akram, for providing me with an incredible project in the field of optics and physics. His extensive knowledge and experience guided me throughout this research, offering valuable insights and ensuring the project's success.

I am also thankful to my co-supervisor, Dag Werner Breiby, whose invaluable comments and suggestions have greatly enriched this project. His expertise and guidance have been crucial roles in the outcomes of my research.

Thanks to Mahdieh Gholamimayani, a Ph.D. candidate, who provided support and assistance during the initial stages of this project. Her valuable advice on experimental techniques and the provision of interesting biological samples have been helpful in the advancement of my research.

I am acknowledged to the Norwegian Research Council for their funding through the FRINATEK (project number 275182) and NANO2021 (project number 272148) programs, and the Centre of Excellence funding scheme (project number: 262644). Their financial support has been crucial in enabling the successful execution of this research.

I am also grateful to the Micro and Nano System Technology department at the University of South-Eastern Norway for their technical assistance and provision of essential equipment. Their resources and expertise have been important in conducting experiments and gathering valuable data.

Lastly, I would like to acknowledge the PaproNoVi+ project, which has provided me with a full scholarship for a two-year master's program and living expenses in Norway. Their support helps me focus on my studies and research, and I am truly grateful for this opportunity.

With the collective support and contributions of these individuals, institutions, and organizations, this master's thesis on Fourier Ptychography Microscopy with UV and IR light has been finished successfully. I am truly thankful for their unwavering support throughout this academic journey.

Borre, Horten, Norway. Date: 18-05-2023

Huy Duong Gia

Contents

1 Introduction

Fourier Ptychography Microscopy (FPM) is a computational imaging method to surpass the physical limitation of the camera [1]. Because of the limited size of the optics pupil, the resolution of the optics is limited. Therefore, when an image is captured, the camera blocks all the high-frequency components of the light field, and the phase of the light field is lost as well. By illuminating the sample at different angles, the camera captures different spatial frequency bands of the object. FPM recovers the extra high-frequency spectrum and the phase by combining all the sub-spectrum iteratively. In general speaking, this method effectively increases the Numerical Aperture (NA) of the optics [2] and retrieves the phase of the object as well [3].

Fourier Ptychography Microscopy is a quickly developing imaging technique that was first introduced in 2013 by Zheng [4]. Since then, several variations of the method have been developed for use in different applications. For instance, FPM has been employed for phase imaging in [3], long working distance in [5], high-speed imaging using LED and camera multiplexing in [6], aberration correction with Embedded Pupil Function Recovery in [7], 3D imaging in [8], and high-angle illumination with dome structure in [9]. FPM has shown significant potential as a promising imaging application. Typically, visible light is used in FPM. However, in this thesis, we explore the use of UV and infrared light for implementing Fourier Ptychography Microscopy. The shorter wavelength of UV light provides higher resolution and can be used for high-contrast imaging of biological samples. On the other hand, infrared light has a long wavelength and low energy that can penetrate through materials such as Silicon or GaAs wafers. In packaging applications, particularly in silicon wafer dicing, conventional optical microscopy techniques are not capable of seeing through silicon material. Therefore, the super-resolution FPM method combined with IR can help to solve this problem.

2 Foundation theory of Fourier Ptychography Microscopy

2.1 Wave vector and plane wave

Suppose there is a point source located at point $A(x_n, y_n)$ in the coordinate system shown in Figure 2-1. The vector representing the propagation of the light from the source to the sample has the same unit vector direction as $\overrightarrow{AO'}$ and magnitude of the wave propagation constant in air is $k_0 = 2\pi/\lambda$, where λ is the wavelength of the light [10, Ch. 11]. This vector is denoted by $\vec{k}$ and is illustrated in Figure 2-1.

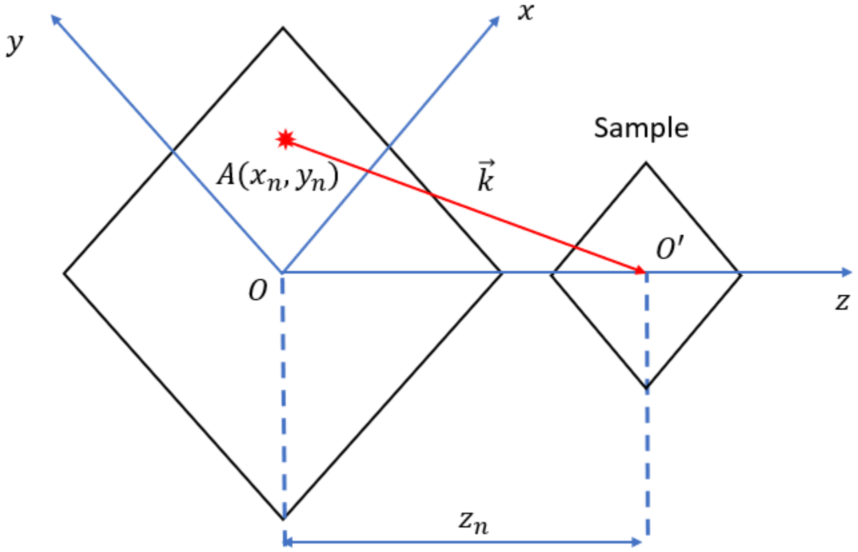


Figure 2-1 Wave vector of the light propagation point to sample.

Therefore, the $\vec{k}(k_{x_n}, k_{y_n}, k_{z_n})$ coordinate is shown in the equations (2-1) to (2-3)

$$k_{x_n} = k_0 \frac{x_n}{\sqrt[2]{{x_n}^2 + {y_n}^2 + {z_n}^2}} \tag{2-1}$$

$$k_{y_n} = k_0 \frac{y_n}{\sqrt[2]{{x_n}^2 + {y_n}^2 + {z_n}^2}} \tag{2-2}$$

$$k_{z_n} = k_0 \frac{z_n}{\sqrt[2]{{x_n}^2 + {y_n}^2 + {z_n}^2}} \tag{2-3}$$

Assume that the sample is a thin layer, the light source is positioned far away from the sample relative to the sample's size, and the illumination can be modeled as a uniform plane wave. In this case, the electromagnetic field E_s of the wave at the sample plane can be represented by equation (2-4) [10, Ch. 12]:

$$E_s = E_0\, exp(-i\vec{k}.\vec{r}) = E_0\, exp(-i(k_{x_n}x + k_{y_n}y)) \tag{2-4}$$

Where $\vec{k}$ is the wave vector with the coordinate $(k_{x_n}, k_{y_n}, k_{z_n})$, $\vec{r} = x\overrightarrow{\boldsymbol{a}_x} + y\overrightarrow{\boldsymbol{a}_y}$ is the position vector, E_0 is the constant amplitude of the wave. Thus, after passing through the sample or object, the electromagnetic field after the object E_r is shown in equation (2-5).

$$E_r = E_{object}(x, y)\, exp(-i(k_{x_n}x + k_{y_n}y)) \tag{2-5}$$

Where $E_{\mathrm{object}}(\mathrm{x}, \mathrm{y})$ represents the object function or complex amplitude of the object.

2.2 Forward imaging model of a coherent system under on-axis illumination.

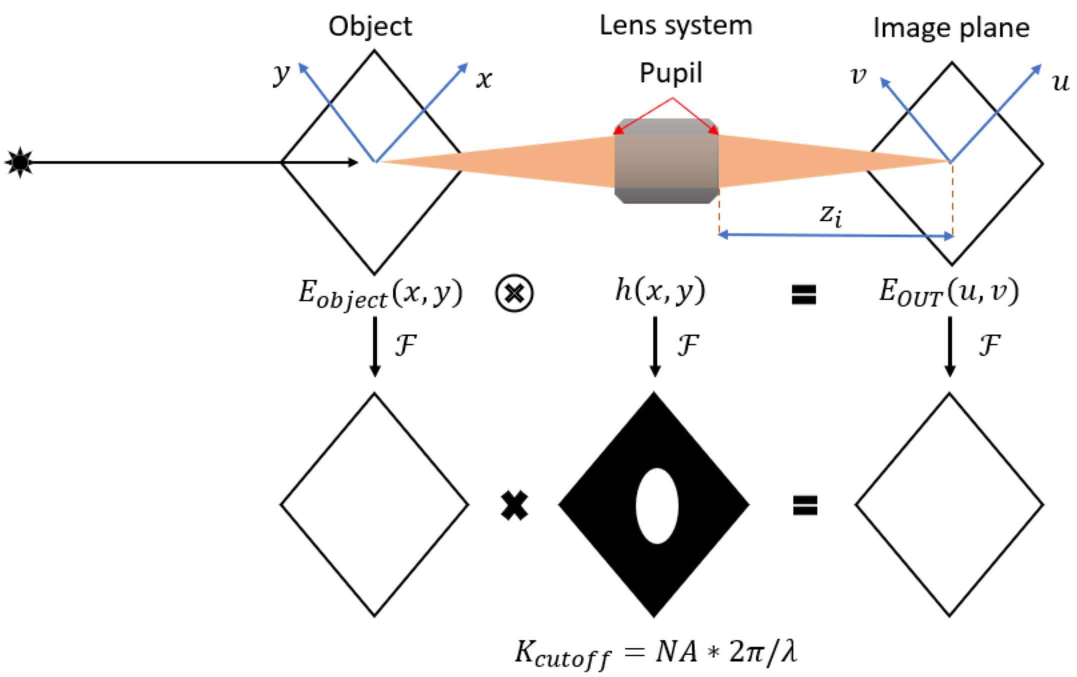


Figure 2-2 Forward imaging model of coherent system under on-axis illumination.

The forward imaging model of a lens system is a mathematical model of how the light of an object propagates through the system and is transformed into an image. Under monochromatic light or coherent light source, the imaging system is linear in complex amplitude [1, Ch. 1] [11, Ch. 6]. In case the illumination is along the optical axis (On-axis illumination) as shown in Figure 2-2, k_{x_n} and k_{y_n} components of the wave vector $\vec{k}(k_{x_n}, k_{y_n}, k_{z_n})$ are zeros. Then the light field after object E_r is equal to the object function $E_{object}(x, y)$, inferred from equation (*2-5*). Thus, the relationship between the light field after the object and image plane is shown in the equation (2-6):

$$E_{OUT}(u,v) = E_{object}(x,y)\, exp\big(-i(k_{x_n}x + k_{y_n}y)\big) \otimes h(u,v) \\ = E_{object}(x,y) \otimes h(u,v) \quad (2\text{-}6)$$

With $E_{object}(x,y)$ represents the object, $E_{OUT}(u,v)$ is the light field at image plane, $\otimes$ denotes the convolution, $h(u,v)$ is the coherent point spread function of the lens which is represented in equation (2-7) [11, Ch. 5–6].

$$h(u,v) = \frac{A}{\lambda z_i} \iint_{-\infty}^{\infty} P(x,y)\, exp\left\{-i\frac{2\pi}{\lambda z_i}(ux + vy)\right\} dxdy \quad (2\text{-}7)$$

where A is constant amplitude, λ is the wavelength of light source, z_i is the distance from the exit pupil to image plane, typically the image plane is placed at focal length, $P(x,y)$ is the pupil function. In general, the coherent point spread function is the scaled Fourier transform of the Pupil function. The pupil function defines the finite extent of the lens and can be represented by a circular function (2-8):

$$P(x,y) = circ\left(\frac{\sqrt{x^2 + y^2}}{w}\right) \quad (2\text{-}8)$$

where w is the pupil radius.

In the spatial domain, the calculation is with the convolution operation, which is time-consuming in a computer. Therefore, the light field is regularly calculated in the frequency domain as expressed in equation (2-9):

$$G_{OUT}(k_x, k_y) = G_{object}(k_x, k_y) \times H(k_x, k_y) \quad (2\text{-}9)$$

Where $G_{object}(k_x, k_y)$ and $G_{OUT}(k_x, k_y)$ are $\mathcal{F}\{E_{object}(x', y')\}$ and $\mathcal{F}\{E_{OUT}(x', y')\}$ respectively with $\mathcal{F}\{*\}$ denotes Fourier Transform calculation, $H(k_x, k_y)$ is the coherent transfer function, which is expressed by the following equation (2-10):

$$H(k_x, k_y) = \mathcal{F}\{h(u,v)\} \\ = \mathcal{F}\left\{\frac{A}{\lambda z_i} \iint_{-\infty}^{\infty} P(x,y)\, exp\left\{-i\frac{2\pi}{\lambda z_i}(ux + vy)\right\} dxdy\right\} \\ = (A\lambda z_i)P(-\frac{\lambda z_i}{2\pi}k_x, -\frac{\lambda z_i}{2\pi}k_y) \quad (2\text{-}10)$$

For notation convenience, $A\lambda z_i$ is set equal to one. Additionally, due to the symmetry of pupil function, the negative sign can be ignored. Thus:

$$H(k_x, k_y) = P\left(\frac{\lambda z_i}{2\pi}k_x, \frac{\lambda z_i}{2\pi}k_y\right) = circ\left(\frac{\sqrt{k_x^2 + k_y^2}}{2\pi w/\lambda z_i}\right) \quad (2\text{-}11)$$

This equation illustrates that the coherent transfer function is actually a lowpass filter that passes all frequencies smaller than the cut-off frequency without amplitude or phase changing, and zeroes out all higher frequencies. The image plane is placed at focal length-f, the cut-off frequency in rad/m is calculated by the equation (2-12). Figure 2-3 shows the coherent transfer function in 2D dimension.

$$K_{cut-off} = \frac{2\pi w}{\lambda f} = NA.k_0 \qquad (2\text{-}12)$$

Where NA is the numerical aperture of the lens system, k_0 is the wave propagation constant in free space which is depicted in previous section 2.1.

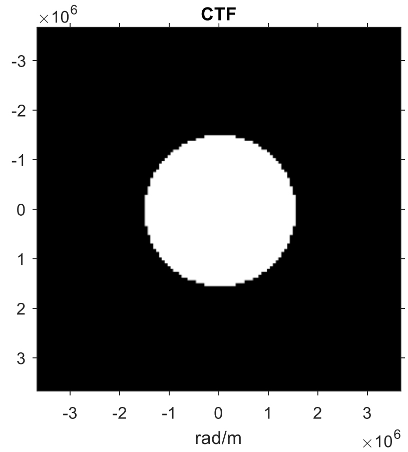


Figure 2-3 Coherent transfer function in 2D dimension, with NA = 0.1 and wavelength is 400nm.

2.3 Off-axis illumination

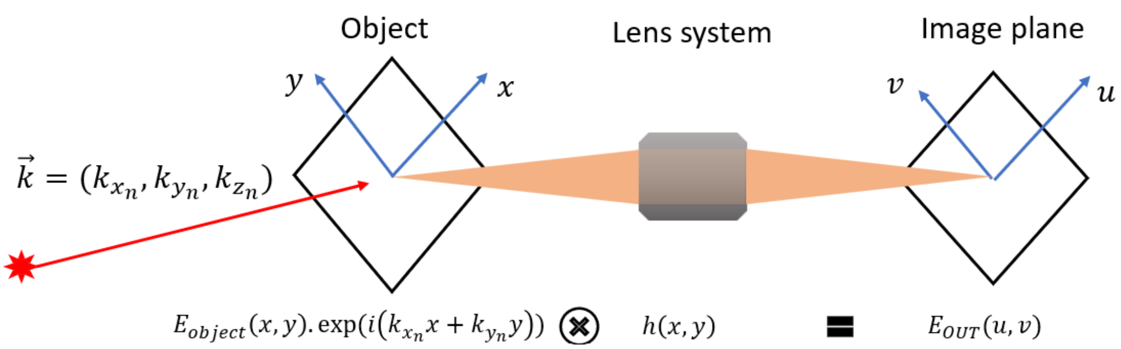


Figure 2-4 Imaging model of coherent system with incident angle of wave propagation.

According to wave propagation theory, section 2.1 and section 2.2, if the light source illuminates the object at a different angle, the light field after passing through the object is equal $E_{object}(x,y) \times \exp\left(i(k_{x_n}x + k_{y_n}y)\right)$, with $E_{object}(x,y)$ is the complex amplitude of the object, and $(k_{x_n},\ k_{y_n},\ k_{z_n})$ is the wave propagation vector $\vec{k}$. The imaging model now is represented as equation (2-13):

$$E_{OUT}(x', y') = [E_{object}(x, y) \times exp\ (i(k_{x_n}x + k_{y_n}y))] \otimes h(x, y) \qquad (2\text{-}13)$$

In the frequency domain the imaging model is presented as equation (2-14):

$$G_{OUT}(k_x, k_y\ ) = G_{object}(k_x - k_{x_n}, k_y - k_{y_n}) \times H(k_x, k_y) \qquad (2\text{-}14)$$

Where $G_{object}(k_x - k_{x_n}, k_y - k_{y_n})$ is shifted Fourier Transform of the object. Figure 2-4 shows the coherent imaging model with incident illumination.

2.4 CMOS (CCD) sensor

A Charge-couple device is a type of sensor that converts incoming photons into electron charges, allowing it to measure the light intensity [12]. Typically, A CMOS sensor is combined with the lens system to measure the light intensity at the image plane after the light has passed through the lens system. However, the CMOS sensor can only measure the intensity and is unable to capture the phase of the light field. Therefore, the phase information is lost in the imaging process.

2.5 Imaging resolution

According to the defined cut-off frequency in section 2.2, the full-pitch resolution of optics is expressed in the following equation (2-15):

$$\delta_f = \frac{1}{f_c} = \frac{\lambda}{NA} \qquad (2\text{-}15)$$

Additionally, we have a definition of half-pitch resolution, which is $\delta_h = \frac{\delta_f}{2} = 0.5\lambda/NA$. If we consider a periodic pattern of black and white bars with the same width, as shown in Figure 2-5, the full-pitch resolution corresponds to the width of a pair of black and white bars. On the other hand, the half-pitch resolution is the width of only one black or white bar.

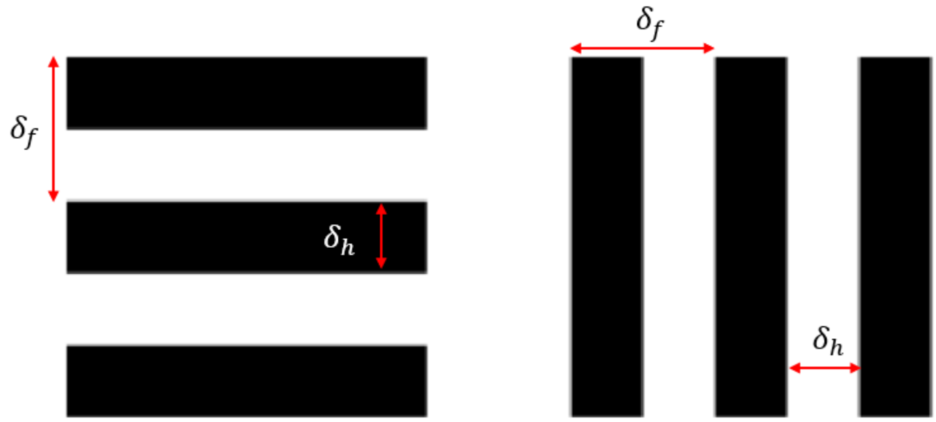


Figure 2-5 The demonstration of full-pitch resolution and half-pitch resolution.

2.6 Nyquist-Shannon sampling theorem

The Nyquist-Shannon sampling theorem is a fundamental concept in discrete-time signal processing. When the spectrum of a signal has a maximum frequency of f_{max}, the signal should be sampled at a frequency of at least the Nyquist frequency, which is $2f_{max}$. This means that the sampling interval should be $\Delta x \leq 1/(2f_{max})$. If the signal is sampled under the Nyquist frequency, the original structure of the signal cannot be reconstructed [13].

In two-dimensional sampling, the non-zero spectrum is in bandlimit $(-B_x, B_x)$ in x-direction and $(-B_y, B_y)$ in y-direction. Following the Nyquist-Shannon sampling theorem, the sampling grid $(\Delta x, \Delta y)$ should satisfy the following equation (2-16) [14, Ch. 2]:

$$\Delta x < \frac{1}{2B_x};\ \Delta y < \frac{1}{2B_y} \tag{2-16}$$

If the sampling grid does not satisfy the theorem, the result exhibits imperfections known as aliasing [14, Ch. 2]. Applying in the coherent imaging system, f_c defined the non-zero limit of light field spectrum at image plane. To avoid aliasing problem, the pixel size of CMOS camera $(\Delta x, \Delta y)$ is stated as the equation (2-17):

$$\Delta x = \Delta y < \frac{1}{2f_c} = \delta_h \tag{2-17}$$

In this case, the half-pitch resolution δ_h represents the maximum pixel size of the coherent imaging system should be, including the magnification M of the optics.

3 Fourier ptychography algorithm

As described in section 2.3, under an incident angle of wave propagation, the high-frequency signal is shifted to low frequency, which allows them to pass through the Coherent Transfer Function. Therefore, the lens system in Fourier Ptychographic Microscopy captures the high-frequency signal content of the object's light field in the frequency domain by illuminating the samples under the incident angle of wave propagation. The idea of Fourier ptychography is to capture the light field information under different angles of illumination to obtain as much high-frequency information as possible as demonstrated in Figure 3-1. The spectrums of obtained low-resolution images are then iteratively stitched until the recovery error is satisfied. Thereby, a wider frequency range complex spectrum of the object is recovered, which means the resolution of the recovered complex sample will be higher. Finally, the inverse Fourier transform is applied to produce an image with higher resolution and more detail due to the additional frequency-domain information captured by the technique.

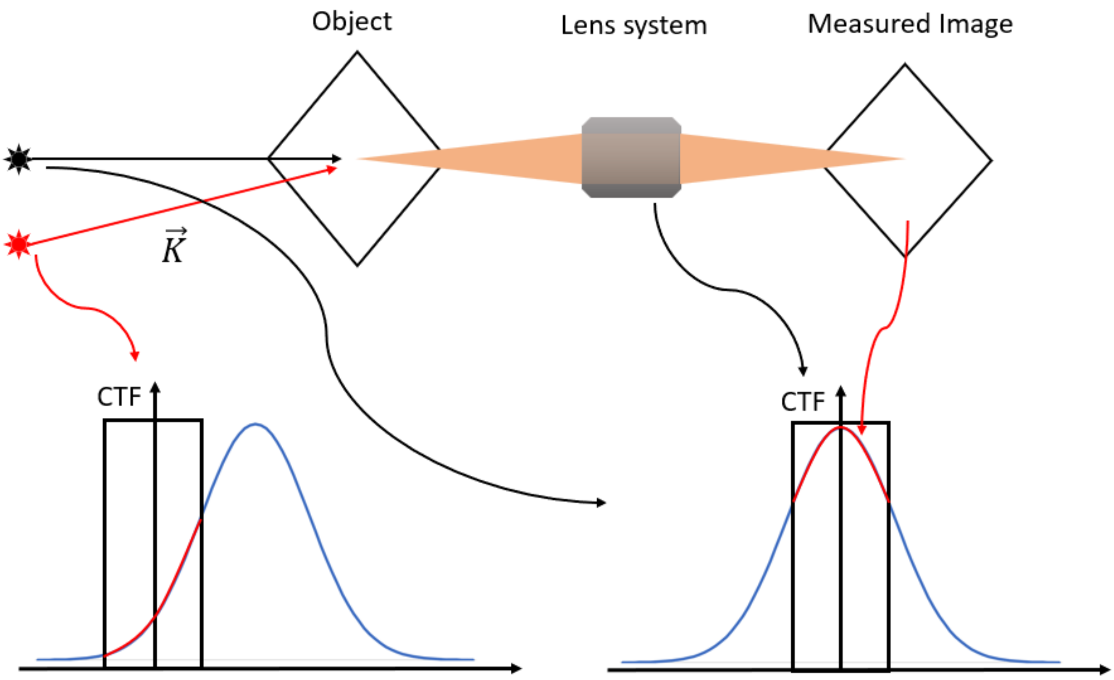


Figure 3-1 Demonstration of Fourier Ptychography Microscopy.

3.1 Fourier Ptychography model

In the Fourier ptychography algorithm, an LED matrix is employed to illuminate the sample from multiple angles. In some cases, a dome-shaped configuration is utilized to increase the angle of illumination [15]. The light intensity of the sample is captured using a lens system and CMOS camera, as shown in the model schematic of FPM, Figure 3-2. It is important that the LED pattern is uniform. Additionally, the LED gap and LED-sample distance are designed such that the optimum overlap area in the frequency domain of neighbor LEDs should be 60%, as discussed in section 3.2. The LED matrix pattern is shown in Figure 3-3.

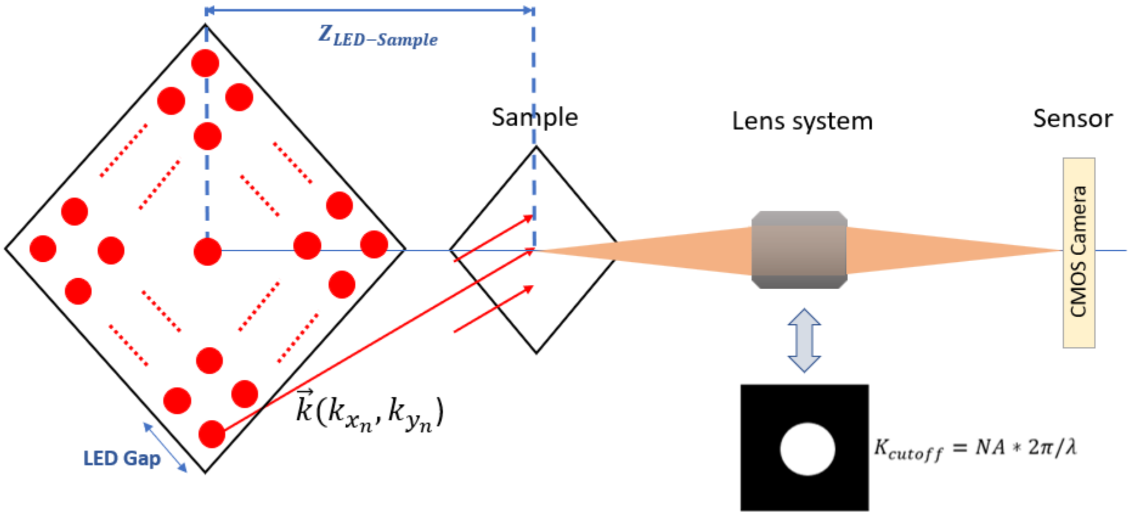


Figure 3-2 Schematic model of Fourier ptychography microscopy.

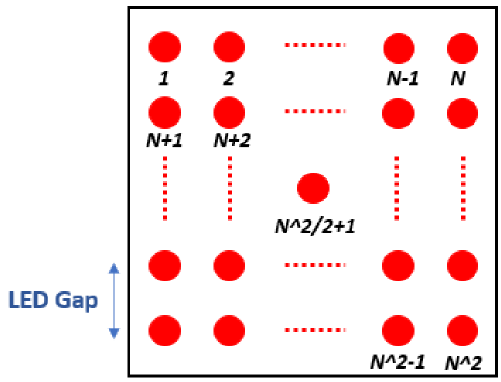
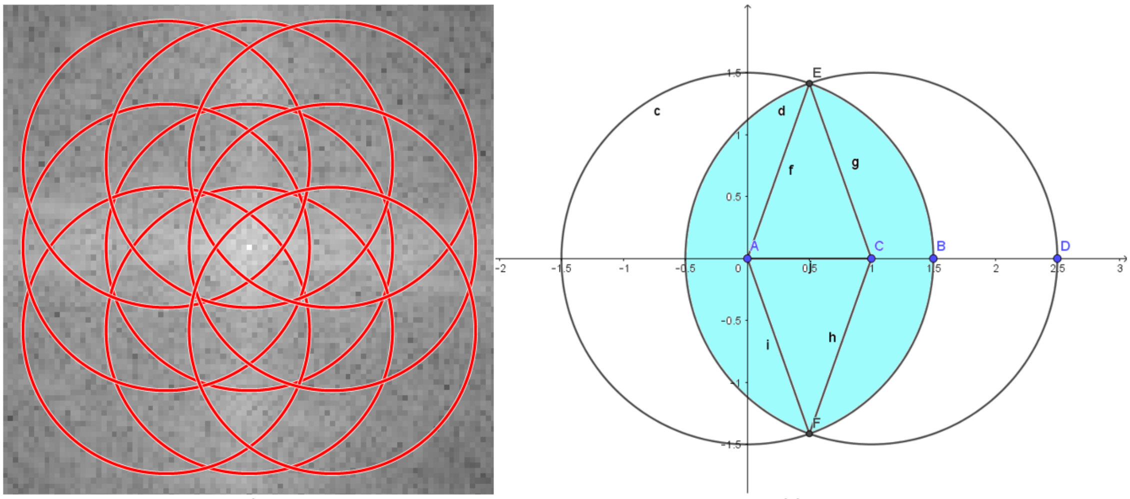


Figure 3-3 LED matrix pattern with the size of NxN.

3.2 Overlap in Fourier domain

To achieve high-quality recovery in Fourier Ptychography, sufficient overlap between the Fourier domain of adjacent LED illuminations is important. This overlap area ensures that the information from each illumination angle can be effectively covered and combined in the Fourier domain. Therefore, the spectrum can be formed completely and accurately [16]. Figure 3-4 shows the overlap in the frequency domain.

The amount of overlap required depends on the specific FPM system being used. However, according to Bunk [17] and Liu [18], the optimum percentage of the spectrum overlap is about 60%. Using basic geometry, the spectrum overlap is calculated by dividing the light-blue area in Figure 3-4b by the total circle area.



a) b)

Figure 3-4 a) The overlapping in the frequency domain, under illumination of 3x3 LEDs matrix, LED gap = 6mm, LED sample distance is 100mm, NA = 0.1, and wavelength is 400nm. b) the overlapping model of two neighbor LEDs in the frequency domain.

3.3 Fourier ptychography resolution

Fourier Ptychography Microscopy method effectively increases the numerical aperture (NA) of the system by using a large field of illuminating angles. The numerical aperture of FPM is calculated using the following equation (3-1)

$$NA_{syn} = NA_{obj} + NA_{ill} \quad (3\text{-}1)$$

Where NA_{syn}, NA_{obj} and NA_{ill} are synthetic, objective and illumination numerical aperture, respectively. And the illumination NA is calculated based on the highest angle of illumination used in the FPM system, equation (3-2):

$$NA_{ill} = n.sin\,(\theta_{ill\_max}) \quad (3\text{-}2)$$

Where n is the refractive index in object space, θ_{ill_max} is the maximum angle of illumination. Thereby, the resolution of FPM is finally shown in equation (3-3).

$$\delta = \frac{0.5\lambda}{NA_{syn}} \quad (3\text{-}3)$$

3.4 Recovery process using EPRY method.

The ERPY (Embedded Pupil Function Recovery) method is a specialized algorithm utilized in Fourier Ptychography to recover both the Fourier spectrum and complex pupil function of an imaging system. The EPRY method involves iteratively refining the reconstructed high-resolution image by minimizing the difference between the low-resolution images and their corresponding estimates in the Fourier domain. This process employs a non-linear optimization algorithm that considers the errors in each iteration, progressively minimizing errors to produce a more precise reconstruction [19]. The ERPY-FPM method further improves this process by recovering complex pupil function, resulting in reduced aberration error. Additionally, this method exhibits robustness to noise and achieves faster convergence compared to traditional FPM recovery techniques [7]. The following steps below describe in detail the EPRY-FPM algorithm. The flow chart of the EPRY-FPM algorithm is shown in Figure 3-5.

- **Step 1:** Initialization of the EPRY algorithm begins with an initial estimate of the complex-valued pupil function. This initial guess can be made to speed up the recovery process. In EPRY, the ideal coherent transfer function is typically used as the initial pupil function estimate. Additionally, the initial object estimate is obtained by taking the Fourier Transform of an upscaled low-resolution image acquired under on-axis illumination.
- **Step 2:** For each wavevector number $i = 1{:}J$, where J is the total number of raw images, the EPRY algorithm calculates the un-updated spectrum $\Phi_i(u)$ by multiplying the current estimate of the complex-valued pupil function $P_i(u)$ with the spectrum of the current reconstructed object $O_i(u - U_i)$, shifted by the corresponding propagation vector U_i. This can be expressed as the equation (3-4):

$$\Phi_i(u) = P_i(u)O_i(u - U_i) \tag{3-4}$$

- **Step 3:** The un-updated low-resolution image $\phi_i(r)$ corresponding to the i^{th} raw image is obtained by computing the inverse Fourier transform $\mathcal{F}^{-1}\{*\}$ of the un-updated spectrum $\Phi_i(u)$. This can be written as equation (3-5):

$$\phi_i(r) = \mathcal{F}^{-1}\{\Phi_i(u)\} = \sqrt{I_i(r)}\frac{\phi_i(r)}{|\phi_i(r)|} \tag{3-5}$$

  Where $\sqrt{I_i(r)}$ is the amplitude of the i^{th} un-updated image

- **Step 4:** The amplitude of the updated image $\phi_i'(r)$ is updated by replacing it with the measured amplitude $\sqrt{I_{im}(r)}$ from the i^{th} raw image:

$$\phi_i'(r) = \sqrt{I_{im}(r)}\,\frac{\phi_i(r)}{|\phi_i(r)|} \tag{3-6}$$

- **Step 5:** The updated spectrum $\Phi_i'(u)$ is obtained by computing the Fourier transform of the updated image $\phi_i'(r)$:

$$\Phi_i'(u) = \mathcal{F}\{\phi_i'(r)\} \tag{3-7}$$

- **Step 6:** Using the difference between the updated and un-updated spectrum, the EPRY algorithm corrects the reconstructed object's spectrum for the i^{th} raw image the following equation (3-8) [20]:

$$O_{i+1}(u - U_i) = O_i(u - U_i) + \alpha\,\frac{P_i^*(u)}{|P_i^*(u)|_{max}^2}\,[\Phi_i'(u) - \Phi_i(u)] \tag{3-8}$$

The object spectrum correction is obtained by dividing the difference between the two exit waves by the current pupil function. This correction is then incorporated into the current object spectrum, with a weight proportional to the intensity of the current pupil function estimate. The step size of the update is controlled by the constant α [7].

- **Step 7:** The EPRY algorithm also corrects the estimate of the pupil function using a similar approach. The corrected pupil function $P_{i+1}(u)$ is obtained by adding a correction term to the current estimate, with a weight proportional to the intensity of the current estimate of the reconstructed object's spectrum. The correction term is given by (3-9):

$$P_{i+1}(u) = P_i(u) + \beta\,\frac{O_i^*(u-U_i)}{|O_i^*(u-U_i)|_{max}^2}\,[\Phi_i'(u) - \Phi_i(u)] \tag{3-9}$$

- **Step 8:** Steps 2 through 7 are repeated for all the raw low-resolution images until all the images are corrected.
- **Step 9:** Steps 2 through 8 are repeated until the recovery object's spectrum converges.

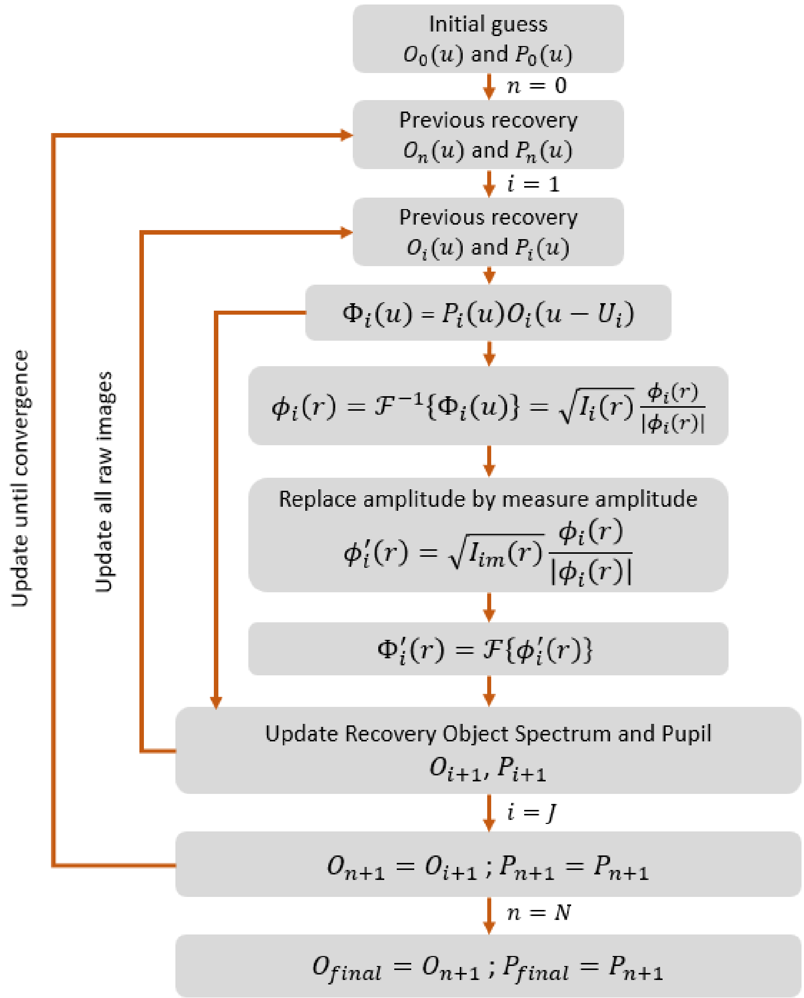


Figure 3-5 The flow chart of EPRY-FPM algorithm.

3.5 Intensity correction

In a real experiment, the power of the LEDs is uncertain as the manufacturer cannot produce all LEDs with the same power or relative luminous intensity. Therefore, if we illuminate the sample with two different LEDs at the same position, the sensor will receive slightly different intensities. To address this issue, Bian [21] introduced an intensity correction method for a robust Fourier Ptychographic Microscopy (FPM) approach, which aims to reduce the intensity uncertainty error.

This method corrects the intensity by using a correction factor $c_i = \sum_{x,y} I_i \, / \sum_{x,y} I_{im}$ to minimize error between recovery amplitude and measurement amplitude which is expressed as $\sum_{x,y} abs(\sqrt{I_i} - \sqrt{I_{im}})$. The entire process is depicted in Figure 3-6. However, it is crucial to know that running the process with a high loop may eliminate high-frequency signals, leading to a decline in the reconstructed image's resolution.

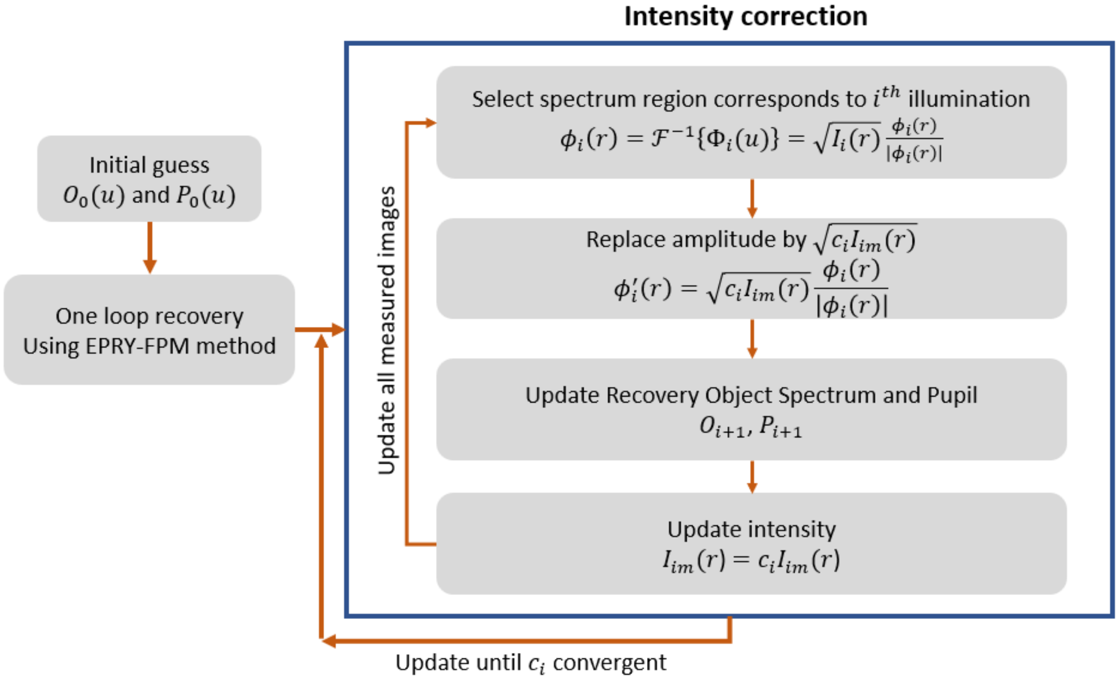


Figure 3-6 Flow chart for intensity correction process.

3.6 Background noise reduction

When using a high angle of illumination, the resulting images are dark-field images. Such images have a low signal-to-noise ratio (SNR) due to the presence of background noise, including stray light, white noise, electronic noise, and other forms of interference. The presence of background noise can significantly impact the performance of Fourier Ptychographic Microscopy (FPM).

3.6.1 Thresholding method

The thresholding method involves selecting multiple uniform sub-regions and computing their average intensity to determine the thresholding level of noise, as shown in equation (3-10).

$$\varepsilon_i = \langle \sum_n \langle I_i^n(r) \rangle \rangle \tag{3-10}$$

Where the $I_i^n(r)$ is the intensity of the n^{th} subregion of the i^{th} dark-field image, $\langle * \rangle$ represents the average value, ε_i is the noise thresholding level. The noise is then reduced by subtracting the thresholding level from the dark-field images, as demonstrated in equation (3-11).

$$I_i^u(r) = I_{im}(r) - \varepsilon_i \tag{3-11}$$

Where the $I_{im}(r)$ is the i^{th} measured dark-field image, $I_i^u(r)$ is the i^{th} filtered image. To maintain the realistic intensity values, any negative pixels are zeroed out [22]. Figure 3-7 shows the images before and after filtering using the thresholding method. Compared Figure 3-7 b1-c1 with b2-c2, the intensity of the uniform sub-regions after filtering varied in a smaller range, which illustrates that the noise was reduced.

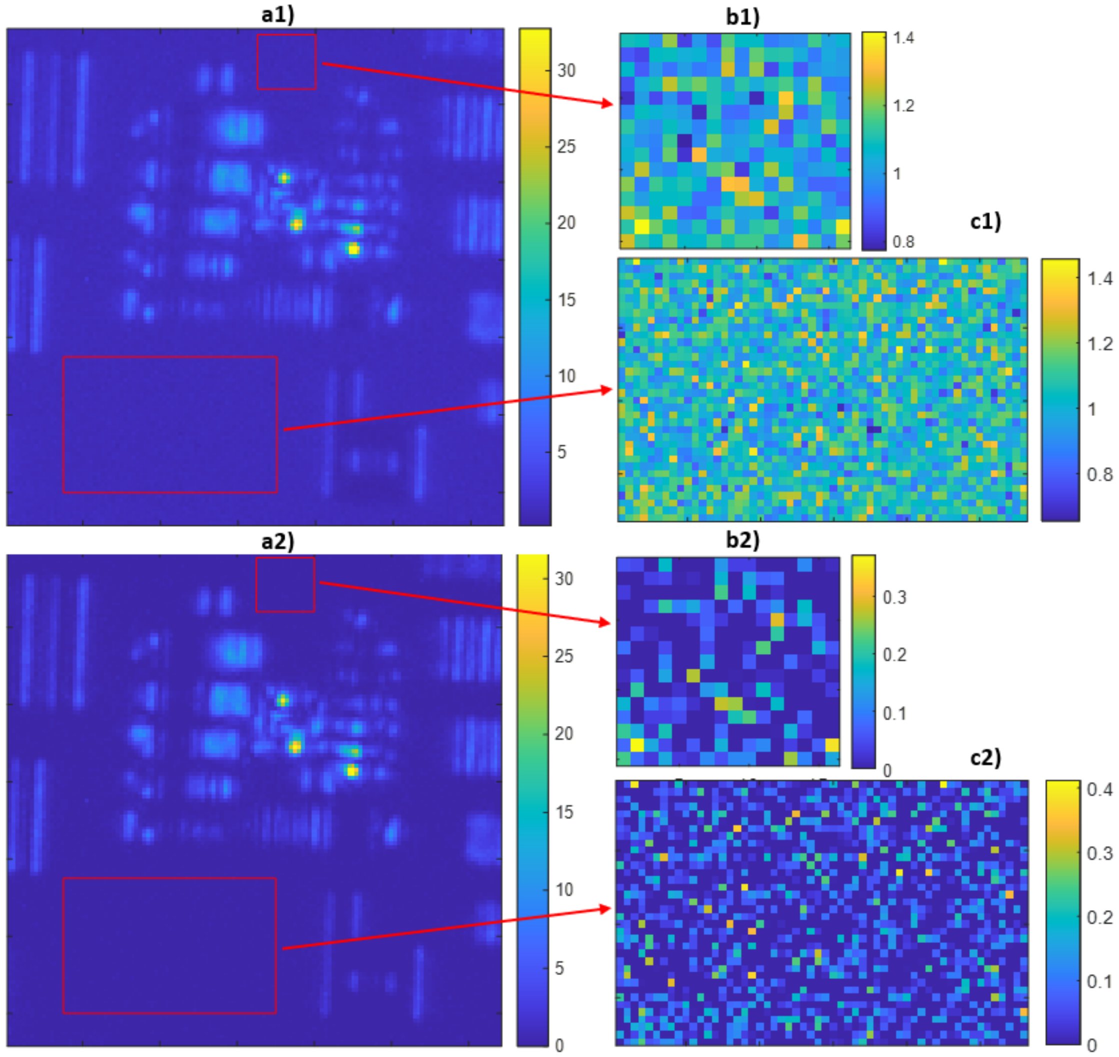


Figure 3-7 a1) and a2) are the dark-field images before and after filtering using the thresholding method. b1) and b2) are subregions 1 before and after filtering using the thresholding method. b1) and b2) are subregions 2 before and after filtering using the thresholding method.

3.6.2 Improved thresholding method.

During experiments with real samples, selecting blank subregions can be challenging and unsuitable in some cases. The improved thresholding method introduced by Lou [22] offers a solution to reduce background noise without requiring blank subregions.

In the recovery process of Fourier Ptychography Microscopy (FPM), the reconstruction begins with bright field images that have a high Signal-to-Noise Ratio (SNR) in both the spatial and frequency domains. When the process comes to recovering the dark-field image, the intersection of the bright field region generates a target dark-field image with better SNR than the measured image. The thresholding value is then determined using the following equation (3-12):

$$\varepsilon_i = \langle I_{im}(r)\rangle - \langle I_i(r)\rangle \tag{3-12}$$

Where the ε_i is the thresholding level, $I_{im}(r)$ is the i^{th} measured image, $I_i(r)$ is the target dark-field image, $\langle * \rangle$ denotes the average value. Thereby, the filtering intensity is then calculated in equation (3-13):

$$I_i^u\,(r) = I_{im}(r) - \alpha.\varepsilon_i \tag{3-13}$$

Where $I_i^u\,(r)$ is the dark-field intensity after filtering. α is the weight factor to balance noise reduction and FPM performance. Choosing a large α reduces noise but can also eliminate high-frequency signals. In this study, we use $\alpha = 1$. Figure 3-8 shows the background noise reduction process using the improved thresholding method. Noted that running the process with a high loop count may result in the elimination of high-frequency signals, which can reduce the resolution of the reconstructed image.

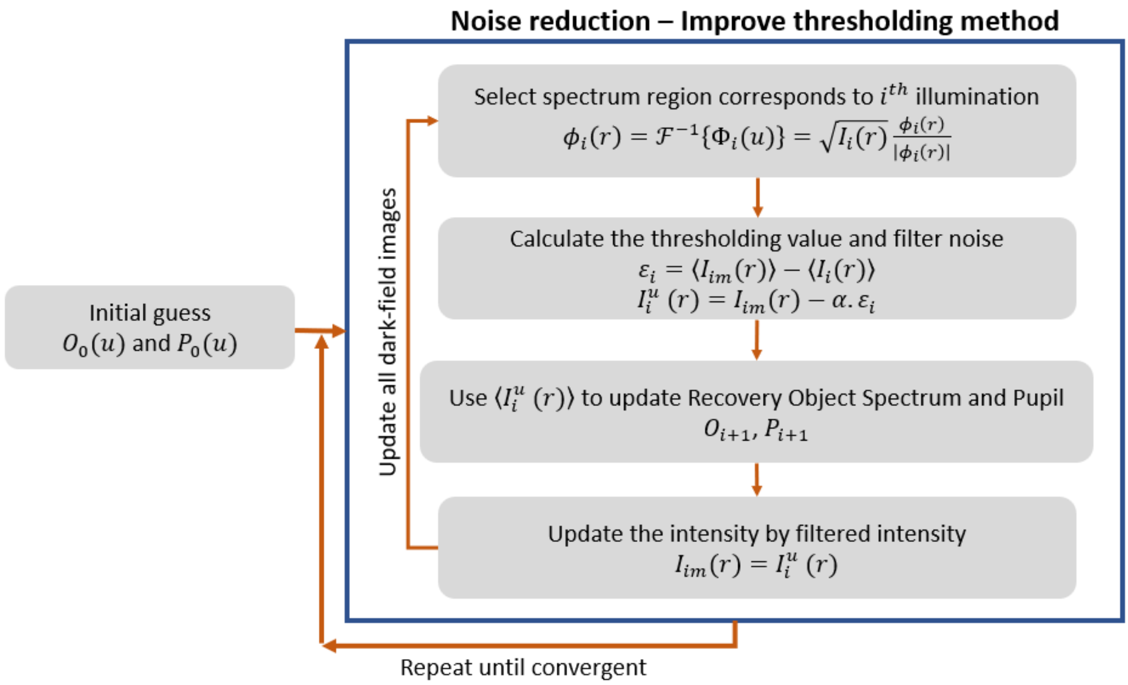


Figure 3-8 Flow chart for background noise reduction process using improved thresholding method.

3.7 Convergence index

The convergence index is a metric used to evaluate the convergence or quality of an iterative process. It represents how well the algorithm is converging toward an optimal solution [23]. Thereby, the convergence index can be applied in Fourier Ptychography to calculate the reconstruction error between the measured images and the estimated images at each iteration of the optimization process. Bian [21] presents a specific formulation for the convergence index, given by equation (3-14):

$$Convergence\ index = \sum_i \frac{mean(\sqrt{I_{im}(r)})}{\sum_{x,y} abs(\sqrt{I_{im}(r)} - \sqrt{I_i(r)})} \tag{3-14}$$

When the $I_{im}(r)$ and $I_i(r)$ are the measured image and estimated image, respectively. $\sum_{x,y}(*)$ denotes the summation of all pixels, $\sum_i(*)$ denotes the summation of all measured images. The lower error between the measured images and the estimated images results in a higher convergence index. Monitoring the convergence index allows us to evaluate how well the recovery process's effectiveness. Thus, some uncertainties in the FPM setup can be optimized by monitoring the convergence index, especially LED-Sample distance. Figure 3-9 presents the flow chart to optimize the LED-Sample distance by monitoring the convergence index.

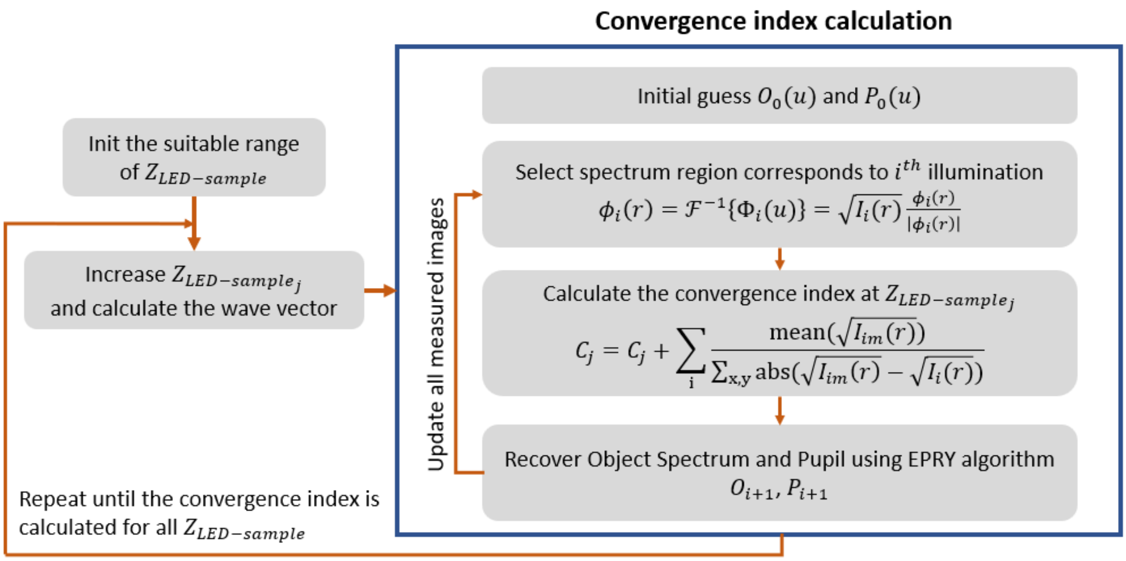


Figure 3-9 Flow chart of monitoring the convergence index by the LED-Sample distance changing.

4 Method

4.1 System architecture

4.1.1 Setup

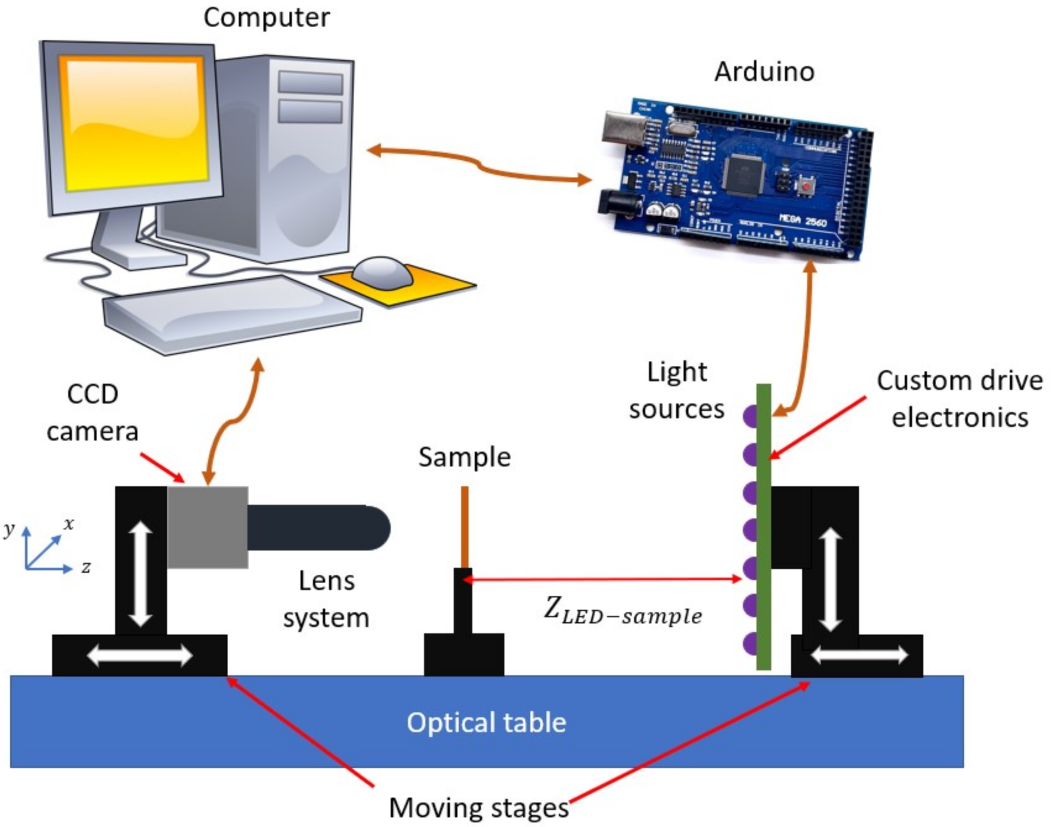


Figure 4-1 Schematic of the experiment set-up. The computer sketch was taken from Wikipedia (open source).

The experiment setup, which is shown in Figure 4-1, includes a CMOS camera, lens system, sample, control computer, Arduino board, custom drive electronics, UV or IR LEDs, moving stages, and optical table.

To begin with, the UV or IR LEDs are soldered onto the custom drive electronics to create a light source matrix. The custom drive electronics is designed to receive commands from Arduino GPIO pins to turn the LED on one by one so that the LED can illuminate the sample. Moreover, the custom drive electronics helps place the LEDs at the precise position on the matrix, which is very important, as discussed in section 4.1.2.

Secondly, the CMOS camera and lens system are combined to measure the light intensity at image plane. The computer is used to capture raw images from the CMOS Camera and control the LED matrix via the Arduino board. The computer is connected to the camera by USB cable 3.0, and communicates with Arduino by UART protocol. Software is developed on the computer with two primary functions. The first function is to calibrate the central LED, refer to section 4.2.1. The second function is to take measurements, refer to section 4.2.2, which sequentially turns the LED on and captures raw images from the CMOS camera.

Thirdly, the CMOS camera, the sample, and the custom drive electronics are mounted on the moving stages to facilitate focusing on the sample and calibrating the central LED location exactly on the optical axis.

Finally, these stages are mounted on the optical table to minimize the vibration during experiment. With optics experiments, small vibrations can make a big bad impact on results. The experimental setup is shown in Figure 4-2.

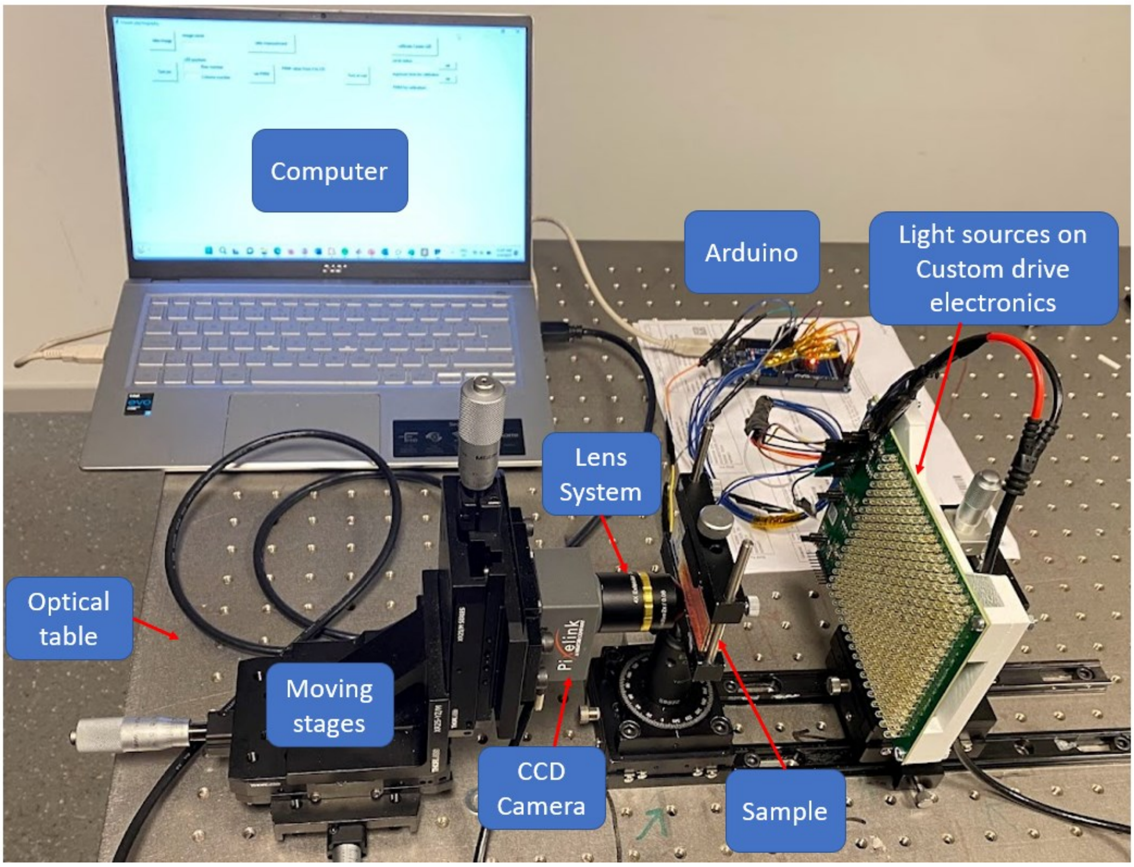


Figure 4-2 The practical experiment setup.

4.1.2 Custom Drive Electronics.

There are three reasons for developing custom drive electronics. The first reason is that a large number of LEDs are required to implement the FPM algorithm for illuminating the sample to obtain high illumination NA. For example, a 17x17 LED matrix requires 289 LEDs in total. Second reason, these light sources need to be constant in light intensity. In the market, LEDs are typically controlled by a scan method, which periodically turns on LEDs at a high frequency that cannot be recognized by human eyes. This is not suitable for optical experiments. Therefore, a custom driver is crucial for easily controlling a large number of constant light sources. The third reason is that the exact position of the LEDs in the matrix is important and significantly affects the recovery result. Hence, a PCB with a precise design is needed to maintain the LED position accuracy in the desired matrix shape.

Custom drive electronics uses three STLED524 Integrated Chips (IC) [24] to control 289 LEDs arranged in a 17x17 matrix. Each STLED524 IC is capable of controlling up to 120 LEDs individually and can adjust the LED current from 0 to 35mA in 256 levels. The communication between the microcontroller and the STLED524 ICs is established through the SPI communication protocol. An Arduino board is used as the microcontroller to receive commands from the computer and send control commands to the STLED524 ICs. The firmware is provided in “Appendix B: Arduino Firmware”. The complete schematic of custom drive electronics is shown in “Appendix C: Custom drive electronic schematic design”.

The PCB design for maintaining LED position accuracy is done by designing with 2 holes that fit two legs of the LED very well, which avoids movement during soldering. The diameter of the used LED leg is 0.71mm, and the holes are designed with a diameter of 0.75mm. The distance between two neighboring LEDs, also known as the LED gap, is 6mm. Therefore, the error in LED position in the matrix is $0.02\text{mm}/6\text{mm} \approx 0.3\%$.

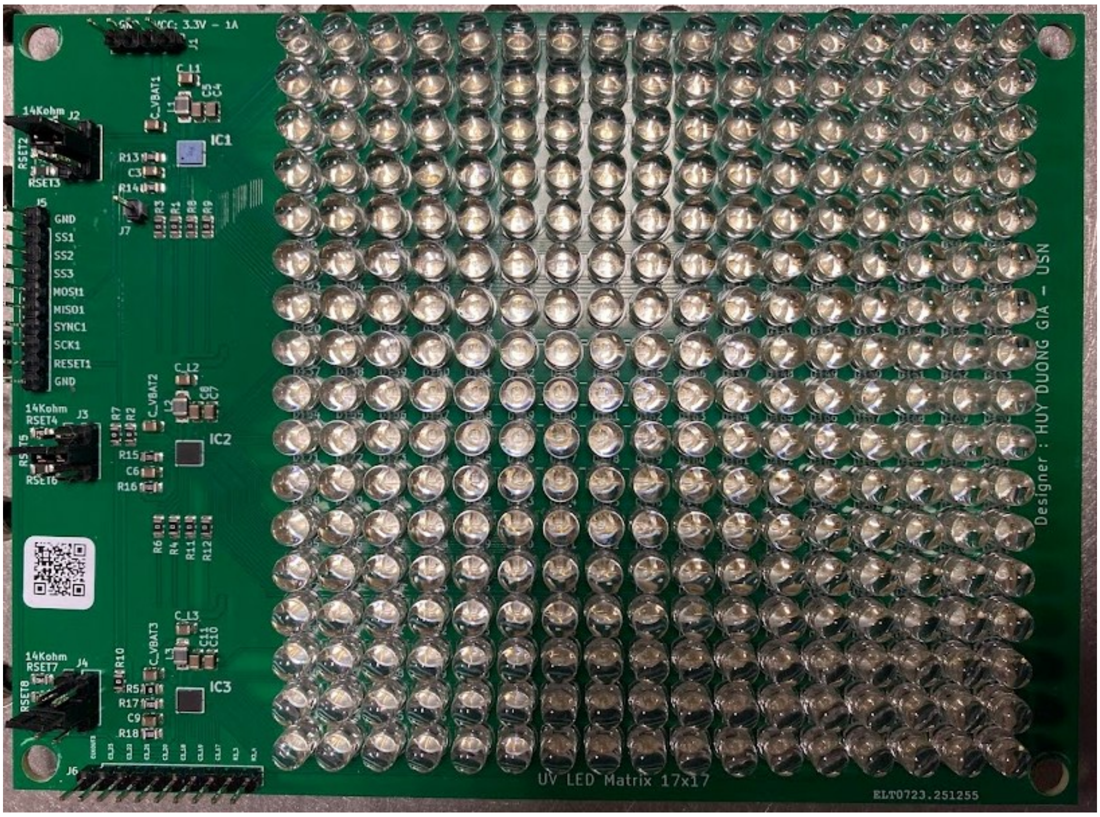


Figure 4-3 Our complete PCB of custom drive electronics with 17x17 UV LED matrix. The PCB size is 15x11cm. Four corner holes are designed for mounting.

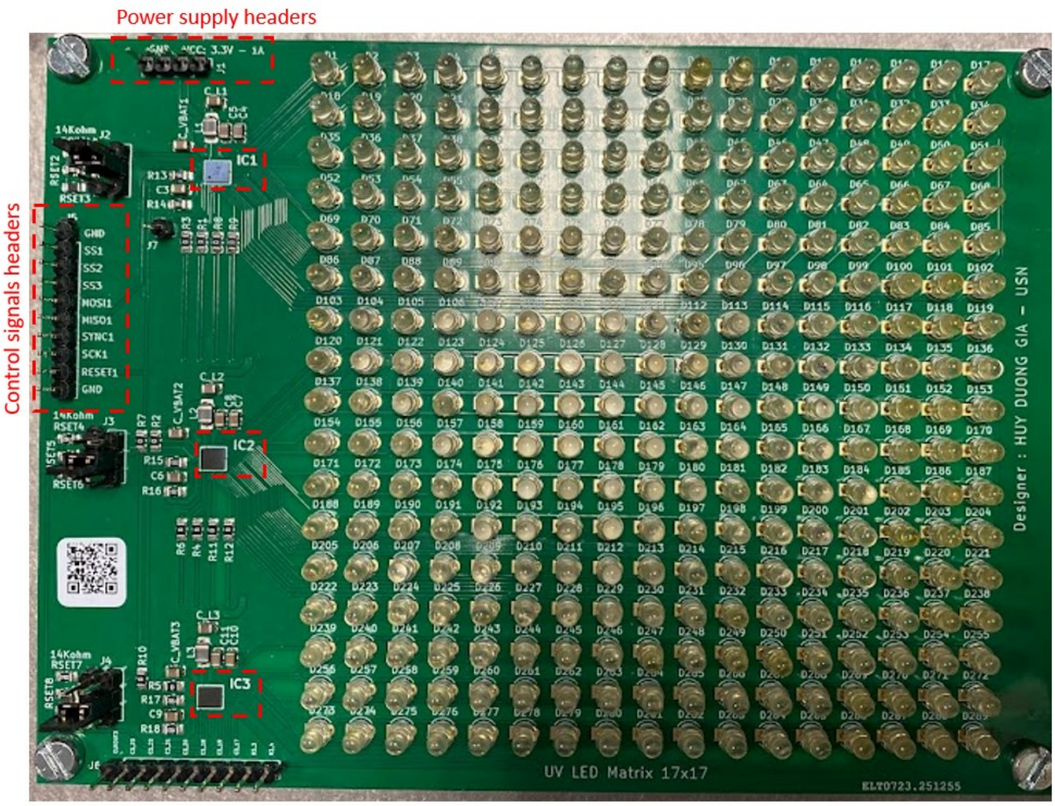
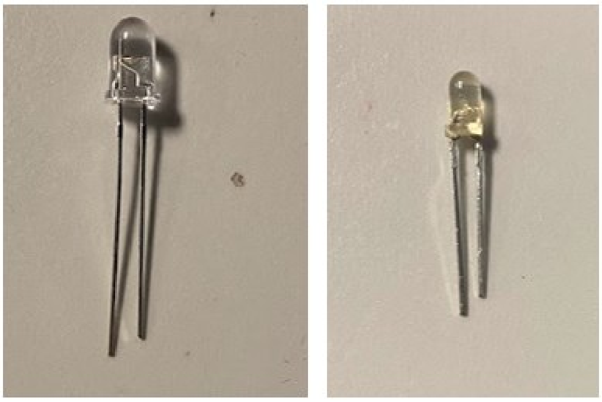


Figure 4-4 Our complete PCB of custom drive electronics with 17x17 IR LED matrix.

4.1.3 The UV and IR light sources

In this thesis, the UV LED is the MT0380-UV-A from Marktech Optoelectronics [25], while the IR LED is the OP265A from TT Electronics [26]. UV and IR LEDs are shown in Figure 4-5. Table 4-1 displays the parameters for both LEDs.



a) b)

Figure 4-5 a) MT0380-UV-A UV LED. The dome is 6mm in diameter. b) OP265A IR LED. The dome is 3mm in diameter.

Table 4-1 UV and IR LED parameters.

	UV LED	IR LED
Wavelength (nm)	400±15	890±40
Forward current (mA)	20	10
Forward voltage (V)	3.3	1.8

4.1.4 Lens system and CMOS camera

The Lens system consists of a 2x Objective lens [27] and a 4x extension tube [28]. These two equipment are manufactured by Edmund Optics. This combination creates the 4x magnification finite-conjugate lens system. The Numerical Aperture of this objective lens system is 0.1.

Figure 4-6 Objective lens, extension tube, and CMOS camera

The lens system is then mounted on the PL-D799 Pixel link camera [29], which has a pixel size of 3.45x3.45 µm. Due to the 4x magnification of the lens system, 3.45 µm on image plane represents 3.45/4 µm on object plane. Therefore, the actual pixel size that the camera samples from the object is 0.8625x0.8625 µm.

According to the Nyquist-Shannon theorem, the pixel size of the camera that is calculated using equations (*2-17*) and (*2-15*) is $\Delta_x = \frac{0.5\lambda}{NA} = 2\mu m$, where the $\lambda = 0.4\mu m$ is the wavelength of UV LED, $NA = 0.1$ is the Numerical Aperture of lens system. Therefore, the pixel size of the camera satisfies the Nyquist-Shannon theorem. Table 4-2 shows the parameters of the combination used lens system and used camera.

Table 4-2 Parameters of the lens system and camera.

NA	0.1	Bit depth	12-bit
Magnification	4x	Resolution	4096x2016 pixels
CMOS Pixel size	3.45x3.45 µm		

Additionally, the Quantum Efficiency of the camera at 400nm and 890nm are about 55% and 14%, respectively, as illustrated in Figure 4-7. Quantum Efficiency is the percentage of converted incoming photons to converted signal in the camera. This indicates that the camera can still function effectively at wavelengths of 400nm and 890nm.

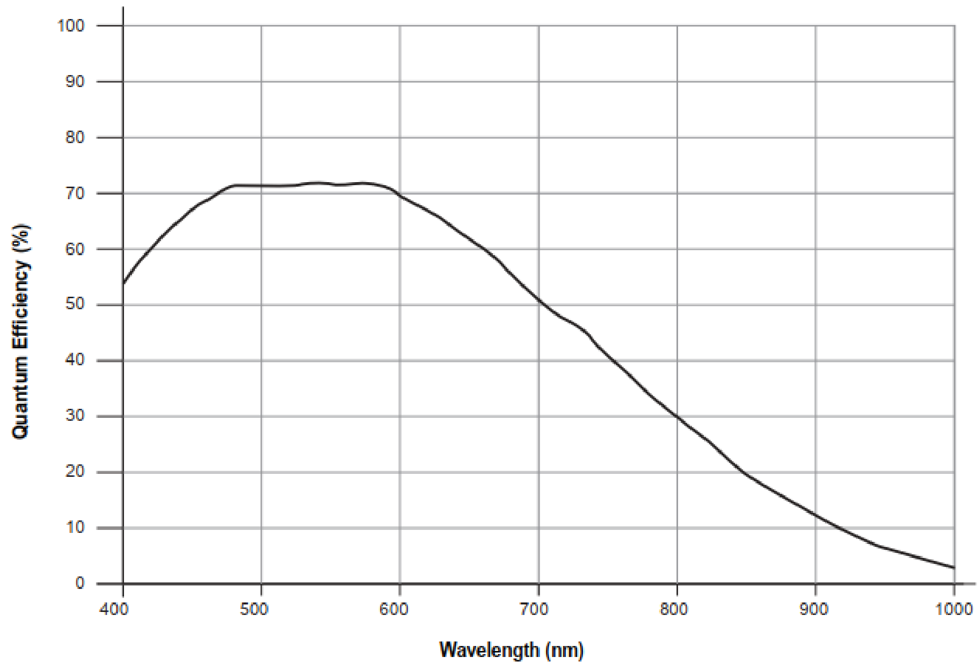


Figure 4-7 Quantum efficiency versus wavelength of PL-D799 PixelLink camera. Quantum Efficiency at 400nm and 890nm are 55% and 14%, respectively. The graph is taken from the datasheet of the PL-D799 PixelLink camera.

4.1.5 The XYZ moving stages

The camera is attached to a moving stage that can adjust its position in the x, y, and z directions. This allows the camera to move in the z-direction to focus on the sample and in the x-y direction to align the desired sample area.

The custom drive electronics are also mounted on the x-y-z moving stages, enabling the light source matrix to move in the z-direction to ensure the optimum spectral overlap between captured images. The central LED can also move in the x-y direction to align with the optical axis of the lens system.

To ensure that the sample is parallel to the LED matrix surface, it is mounted on a rotating stage.

4.1.6 PCB holder and Camera holder design.

Ensuring parallelism between the image plane and the LED matrix surface is important. Because this parallelism affects the LED position and illumination angle in the optical coordinate system. Failure to maintain this parallelism can increase uncertainties and errors in the system, as illustrated in Figure 4-8.

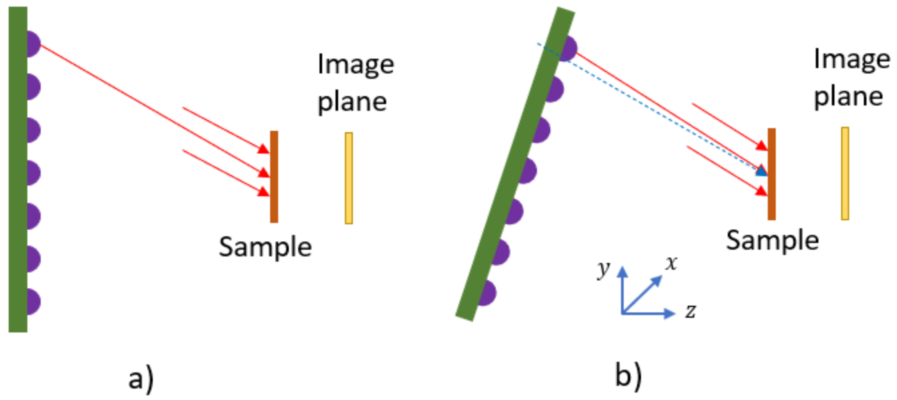


Figure 4-8 a) LED position and illumination angle when LED matrix surface and image plane are parallel. b) LED position and illumination angle when LED matrix surface and image plane are not parallel, the dash-blue vector illustrates illumination angle when LED matrix surface and image plane are parallel.

Therefore, to minimize the LED position and the illumination angle error, the PCB holder and Camera holder are designed in Solidworks and fabricated using a Prusa i3 3D printer with a resolution of 0.15mm. Figure 4-9 shows the PCB holder and Camera holder design.

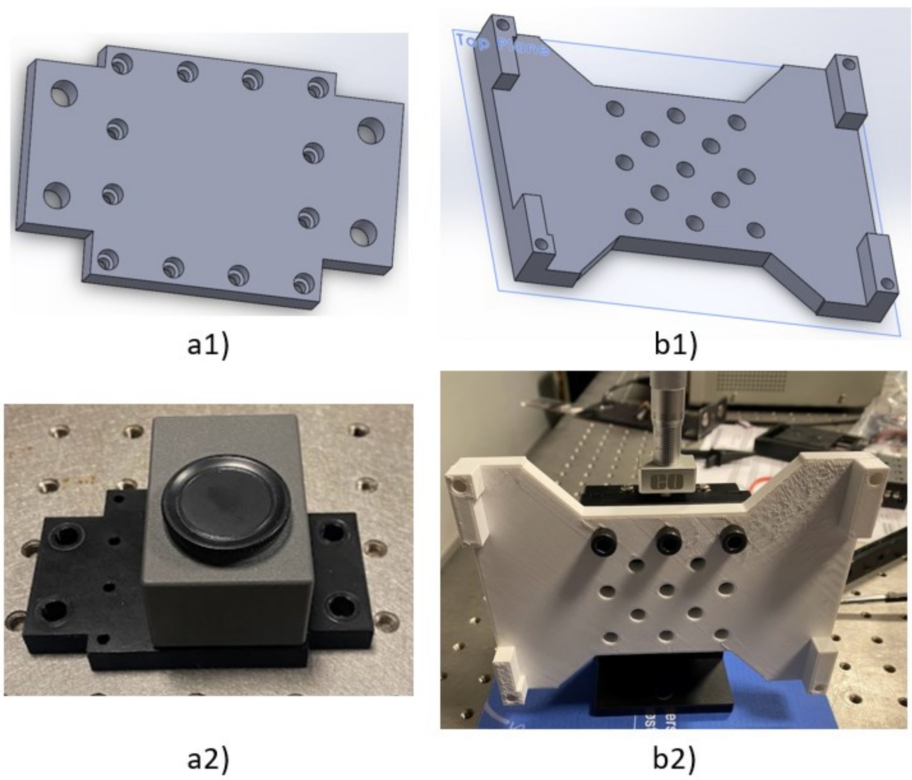


Figure 4-9 a1)-b1) camera holder and PCB holder design by Solidworks. a2) Camera is mounted on holder. b2) PCB holder is mounted on moving stage.

4.2 Software

The software was developed using Python programming and features two primary functions: calibration of the central LED (refer to section 4.2.1) and taking measurements (refer to section 4.2.2). The graphical user interface (GUI) is illustrated in Figure 4-10, while the coding is presented in the "Appendix D: Software program".

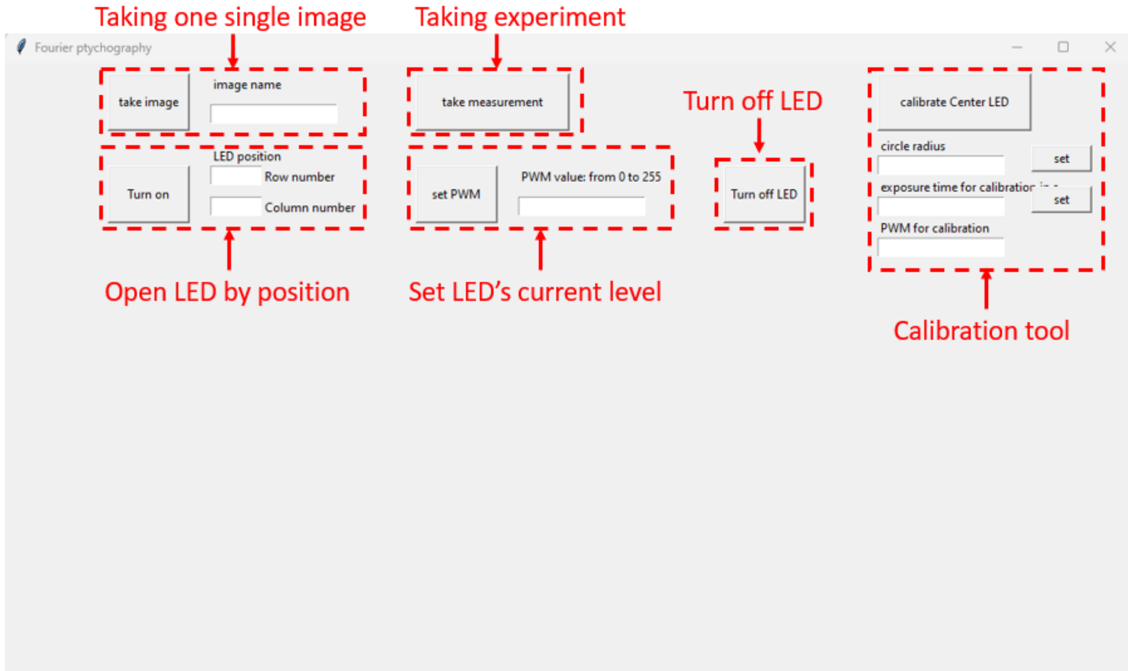


Figure 4-10 Graphic user interface of the developed software and functions description, programmed by Python language.

4.2.1 Calibration of the central LED

After adjusting the camera position to focus on the particular region of the sample that requires high-resolution recovery, the LED matrix must be properly calibrated in the optical system. As shown in Figure 3-2 of the FPM model, the central LED should be positioned along the optical axis. The accuracy of all other LED positions is maintained by the custom PCB. Hence, to ensure that the LED matrix is positioned precisely in the optical system, it is necessary to calibrate the central LED so that it lies exactly on the optical axis and the x-y LED grid is also parallel to the CMOS chip x-y axes. As illustrated in Figure 4-11, the white circle represents the LED's position. If this white circle is located at the center of the image, then the LED is correctly positioned along the optical axis.

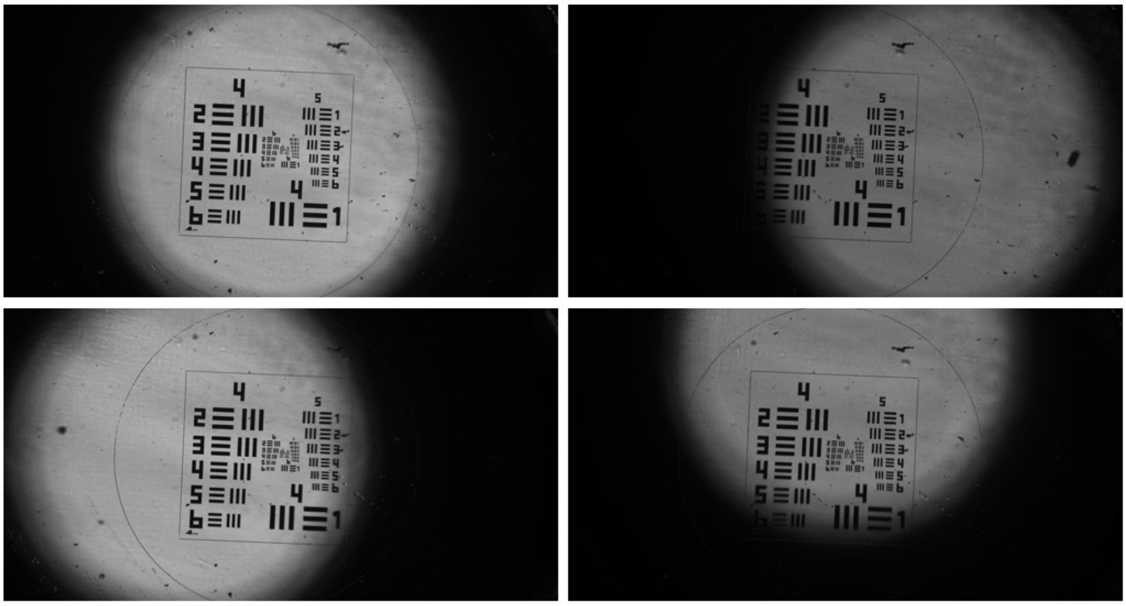


Figure 4-11 Raw images from the camera under the illumination of different LED positions.

To calibrate the central LED, firstly the central LED is turned on to illuminate the sample. Then the software reads the image from the camera and converts the image to a binary image to make the white circle with more contrast. A live video of the binary image is displayed on the software's GUI. A red circle is drawn on the video, the circle center is fixed and positioned exactly at the center of the video, which is the center of the optical system. As a result, the LED matrix can be moved by adjusting the moving stages so that the white circle matches the red circle, indicating that the central LED is positioned along the central line of the optical system. The red circle's radius can be adjusted using the software to match the white circle, as shown in Figure 4-11. Figure 4-12 shows the GUI with the video when the central LED is not calibrated, while Figure 4-13 displays the GUI with the video when the central LED has already been calibrated.

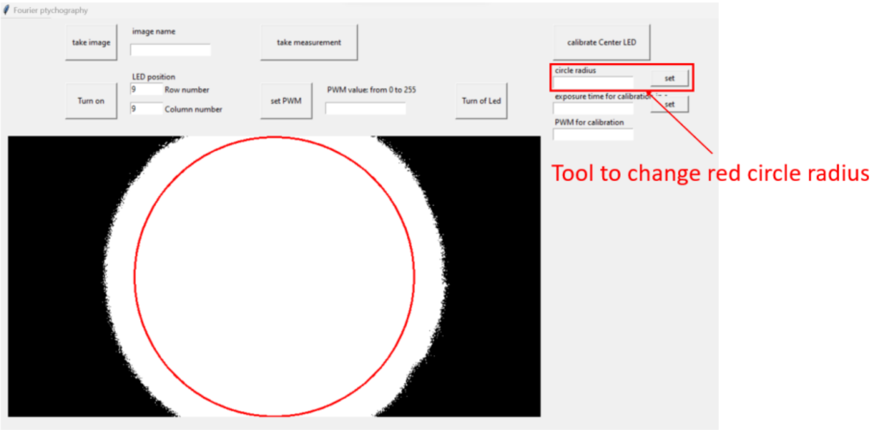


Figure 4-12 The software's GUI displays a live video of the binary image with a red circle on it. The red circle is centered precisely at the image's center, which coincides with the center of the optical system.

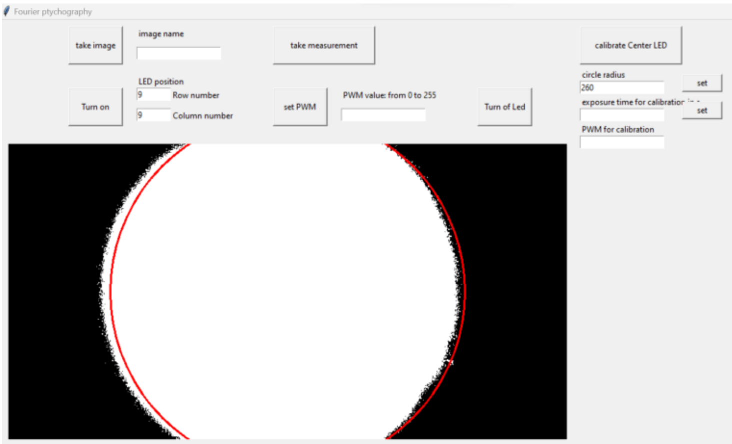


Figure 4-13 The software's GUI displays a live video when the central LED is not yet calibrated.

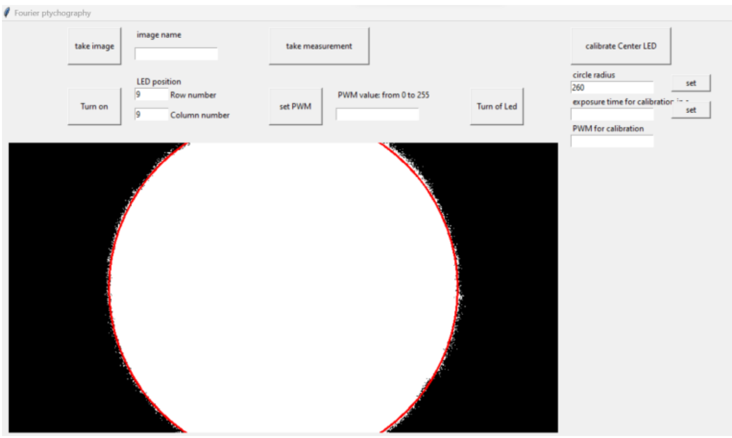


Figure 4-14 The software's GUI displays a live video when the central LED has already been calibrated.

4.2.2 Taking measurements

The taking measurements tool is programmed to take the dataset of images of all LED illuminations for recovering high-resolution image.

Step 1 involves sending a command from the software to the Arduino to turn on the LED and illuminate the sample.

Step 2, the software changes the camera exposure time for each illumination to avoid overexposure or underexposure of the image. Overexposure leads to lost information as the brightness exceeds the solvable range of the camera, while underexposure occurs when the exposure time is too short, and incoming photons are not enough to convert to electron charges. Figure 4-15 shows the images and their histogram in case of overexposure and underexposure. The histogram of the image should be within the solvable range, as illustrated in Figure 4-16.

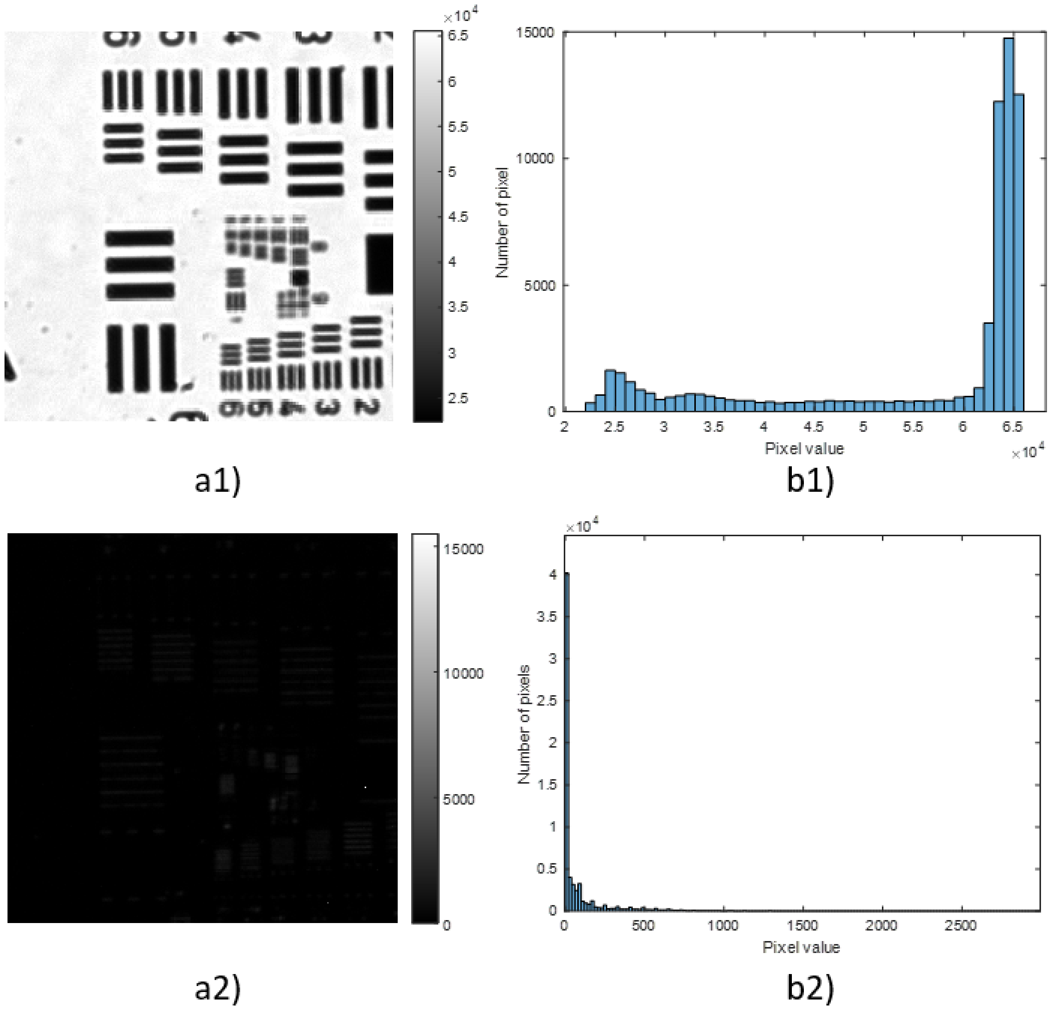


Figure 4-15 a1)-b1) the overexposure image and its histogram, respectively. A part of the image information is lost because the pixel brightness of their pixel is larger than the *maximum brightness of the camera. a2)-b2) the underexposure image and its histogram, respectively. A part of the image information is lost because the brightness of their pixel is smaller than 0.*

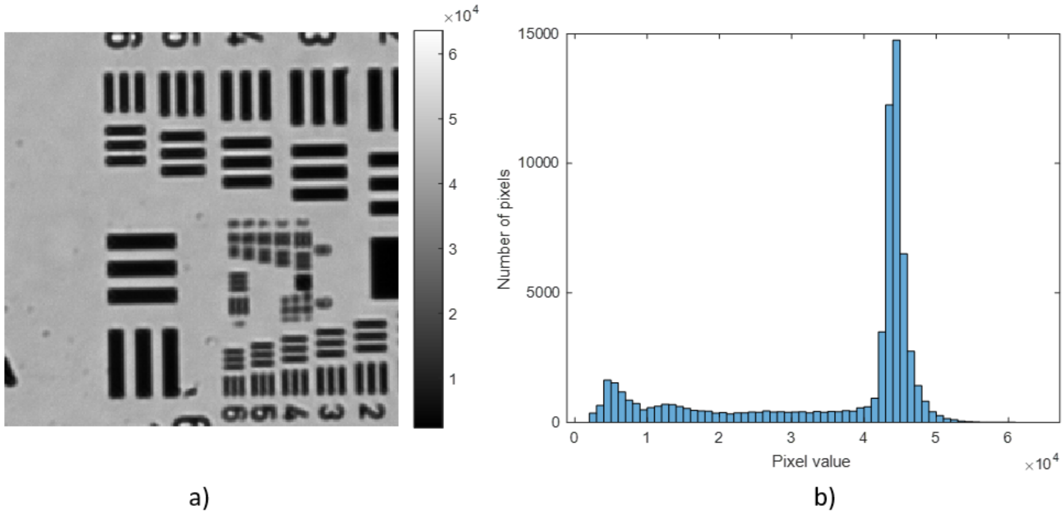


Figure 4-16 a)-b) A good image in terms of brightness and histogram. The distribution of brightness is within the brightness range of the camera, which is from 0 to 65535.

Step 3, the raw image is captured and saved. Additionally, instead of saving the entire image, a small region that needs to be recovered can be saved to minimize the dataset's space.

Step 4, repeat steps 1 to 3 until all illuminations have been implemented. The sequence of illuminations is shown in Figure 4-17.

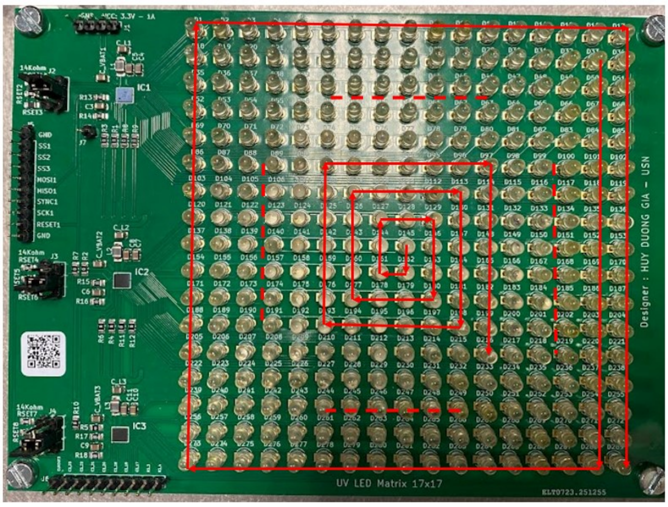


Figure 4-17 The sequence of illumination. The experiment starts from the central LED and then moves to the end in a spiral.

4.3 Proposed recovery process

After taking measurement with a dataset of images under all illuminations, we move on to the final step: recovering the high-resolution image. In this section, we propose an efficient recovery process that can successfully reconstruct high-resolution images. The proposed recovery process is shown as a flow-chart in Figure 4-18.

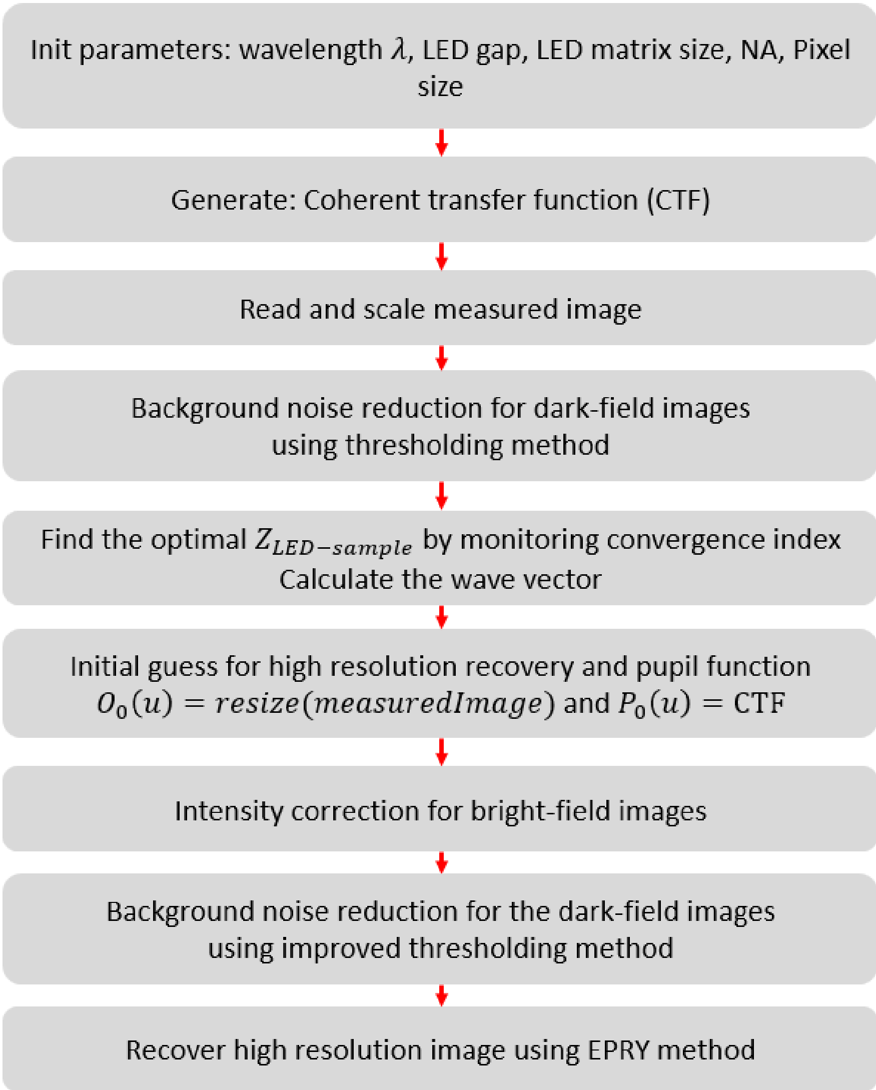


Figure 4-18 Flow chart to recover high-resolution image for the practical dataset.

Step 1: Initialize setup parameters, including:

- The LED gap is 6mm, refer to **section 4.1.2**
- Define LED-Matrix size
- Wavelength λ is 400nm for UV light or 890nm for IR light, refer to **section 4.1.3**
- NA is 0.1 and Pixel size is 0.8625 µm, as discussed in **section 4.1.4**

Step 2: Generate the Coherent transfer function, as described in **section 2.2**.

Step 3: Read and scale images from the dataset. Notice that, since the signal is scaled based on the exposure time, all images need to be scaled at the same exposure time.

Step 4: Reduce background noise for dark-field images by using the thresholding method, as discussed in **section 3.6.1.**

Step 5: Find the optimal $Z_{LED-sample}$ by monitoring the convergence index, **section 3.7**.

- Then the wave vector of all illuminations is calculated using the initial parameters of the LED gap and $Z_{LED-sample}$, with equations (*2-1*) to (*2-3*).

Step 6: Use the measured image (which is upscaled to high-resolution size) under central LED illumination as an initial guess for the high-resolution image. This step helps the recovery process converge more quickly, as the initial guess is close to the high-resolution image. Also, use the Coherent transfer function as the initial pupil function guess.

Step 7: Correct intensity for bright-field images using the intensity correction method described in **section 3.5**.

Step 8: Reduce background noise again using the improved thresholding method, as discussed in **section 3.6.2**. This step is important for samples that do not have uniform regions.

Step 9: Finally, use the EPRY algorithm to iteratively recover the high-resolution image in Fourier space until convergence, as discussed in **section 3.4**.

The MATLAB programming is provided in "Appendix E: Recovery process programming".

4.4 Sample and characteristic evaluation method

4.4.1 2015a-USAF target

The 2015a-USAF target, which is manufactured by Ready Optic, is used to test the resolution of optical systems [30]. This target consists of periodic pairs of black and white bars, as shown in Figure 4-19. The target is made of transparent glass, and the black bars are chrome implanted in the glass. The chrome bar thickness is 1000 angstroms. The widths (w) of the black and white bars can be calculated using equation (4-1), where the group number and element number are indicated in Figure 4-19 b. This width parameter represents the half-pitch resolution of the camera. If the optical system can distinguish between the black and white bars, then the half-pitch resolution of the optical system is at least w. From equation (4-1), the widths are calculated and shown in Table 4-3.

$$w = \frac{1}{2 \times 2^{Group + \frac{Element-1}{6}}} \quad (4\text{-}1)$$

With known bar widths, the 2015a-USAF target is used as the sample in the FPM setup to test and evaluate the resolution of FPM.

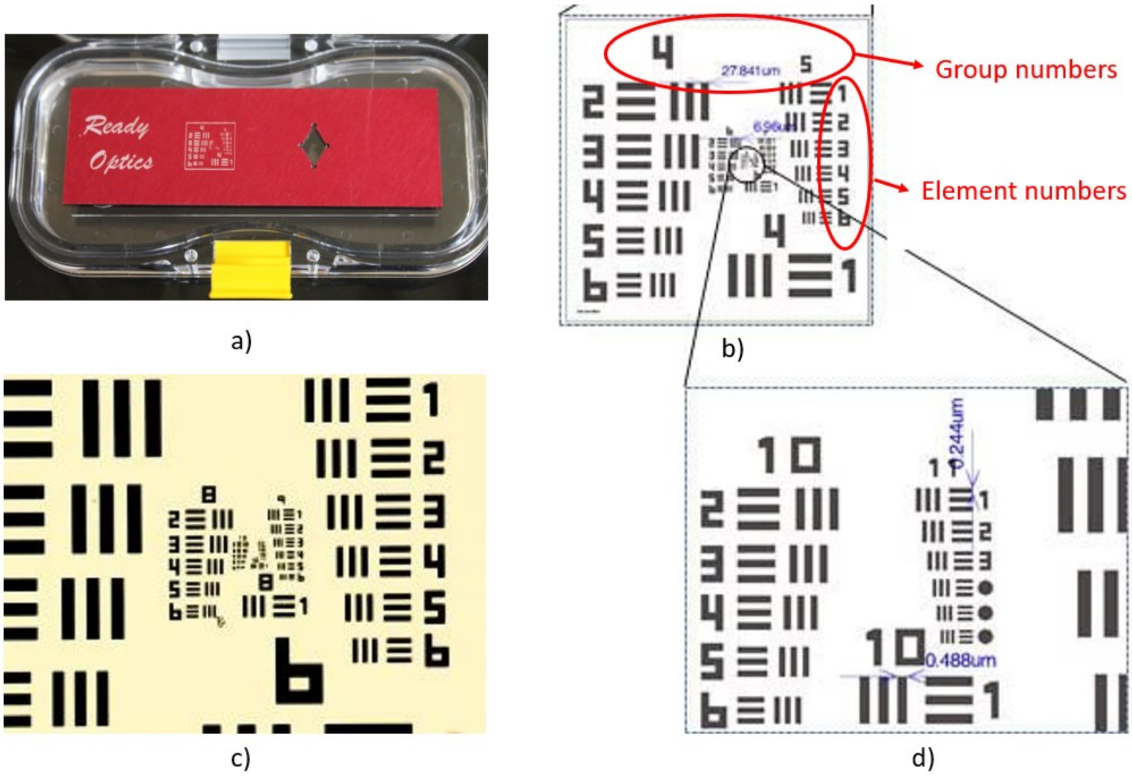


Figure 4-19 2015a-USAF target. a) Picture of the target. b) Group 4 and 5 of the target. c) Group 6, 7, and 8 of the target. d) Group 10, and 11 of the target. The images are taken from the website of the manufacturer Ready Optics [30].

Table 4-3 The Bar's width according to Group and Element number

Group number	Element number	Bar's width (nm)	Group number	Element number	Bar's width (nm)
7	1	3906.25	9	1	976.5625
7	2	3480.073118	9	2	870.0182794
7	3	3100.39268	9	3	775.0981699
7	4	2762.135864	9	4	690.533966
7	5	2460.783301	9	5	615.1958251
7	6	2192.308688	9	6	548.077172
8	1	1953.125	10	1	488.28125
8	2	1740.036559	10	2	435.0091397
8	3	1550.19634	10	3	387.549085
8	4	1381.067932	10	4	345.266983
8	5	1230.39165	10	5	307.5979126
8	6	1096.154344	10	6	274.038586

4.4.2 1951-USAF target

Similar to the 2015a-USAF target, the 1951-USAF target is used to test the resolution of an optical system. It also has periodic pairs of black and white bars. 1951-USAF target is manufactured by Edmund optics and has the maximum resolution is Group 9-Element 3 (Element 9-3) [31]. Figure 4-20 shows the 1951-USAF target.

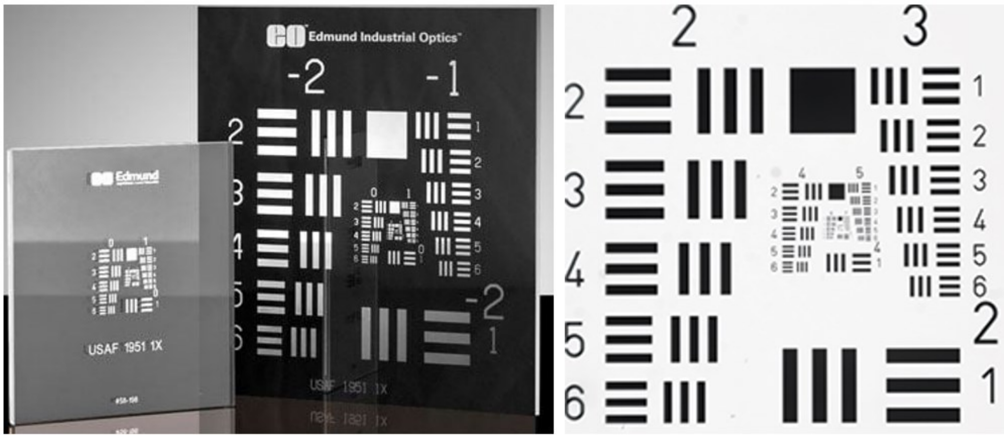


Figure 4-20 1951-USAF target. The images are from the manufacturer Edmund's website [31]

4.4.3 Biological sample

After quantifying the resolutions of FPM with UV light, various bio-samples such as blood cells and thin cartilage samples are tested using FPM with UV light. Figure 4-21 shows some biological samples that use to test with FPM.

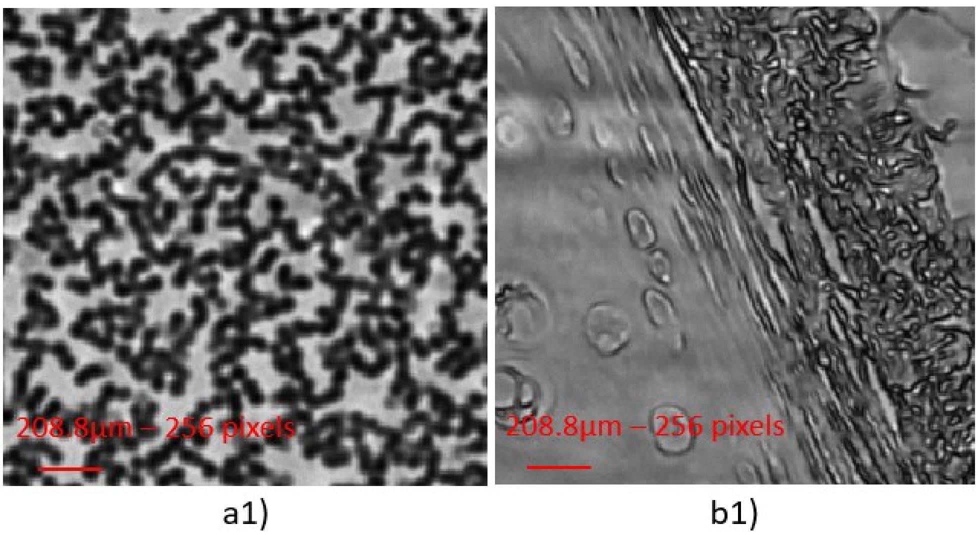
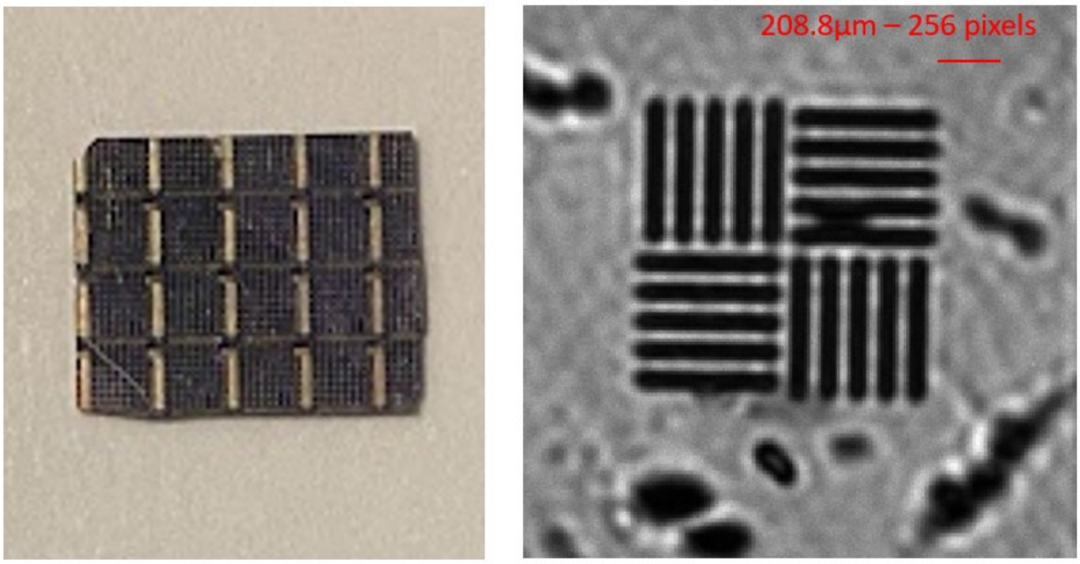


Figure 4-21 Some biological samples under UV light. a) blood cell. b) Slice of cartilage.

4.4.4 Gallium arsenide wafer sample

As mentioned in the Introduction, FPM combined with IR light has the potential to enable imaging through Silicon wafers. We have tried to image structure on a Silicon wafer with our IR light with the wavelength of 890nm. However, Silicon was not so transparent due to its high absorption. To image structure on Silicon wafer, longer wavelengths such as 1500nm are required to pass through Si. Moreover, a specialized camera such as the InGaAs camera is needed to receive the 1500nm wavelength light.

Instead of using a Silicon wafer, a Gallium arsenide (GaAs) wafer is used to test the FPM's ability when combined with IR light. GaAs is opaque under UV or visible light. However, it is transparent under IR light with wavelengths of 890nm. The GaAs wafer is 0.34 mm thick. Additionally, the wafer needs to be polished beforehand. If the wafer is rough, it can affect the light and result in poor image processing. Figure 4-22 displays the piece of the polished GaAs wafer and the structure on it.



a) b)

Figure 4-22 a) A piece of GaAs wafer. B) a structure on the wafer. The image is taken when the wafer is illuminated by IR light.

4.4.5 The sample created by combining of GaAs wafer and 1951-USAF target.

To determine the maximum resolution that FPM-IR can achieve with a GaAs wafer sample, we conducted an experiment in which a polished GaAs sample was attached to a 1951-USAF target. This approach allowed us to simulate the structure of the GaAs wafer same as the USAF sample. The side of the target with metal bars must contact the GaAs wafer surface as closely as possible. Additionally, the used GaAs wafer must be smooth and have no pre-existing structure on it. This was necessary to ensure that any observed features were solely the result of the FPM technique.

For our experiment, we used a GaAs wafer that was 0.34 mm thick. Figure 4-23 provides a visual representation of the experimental setup.

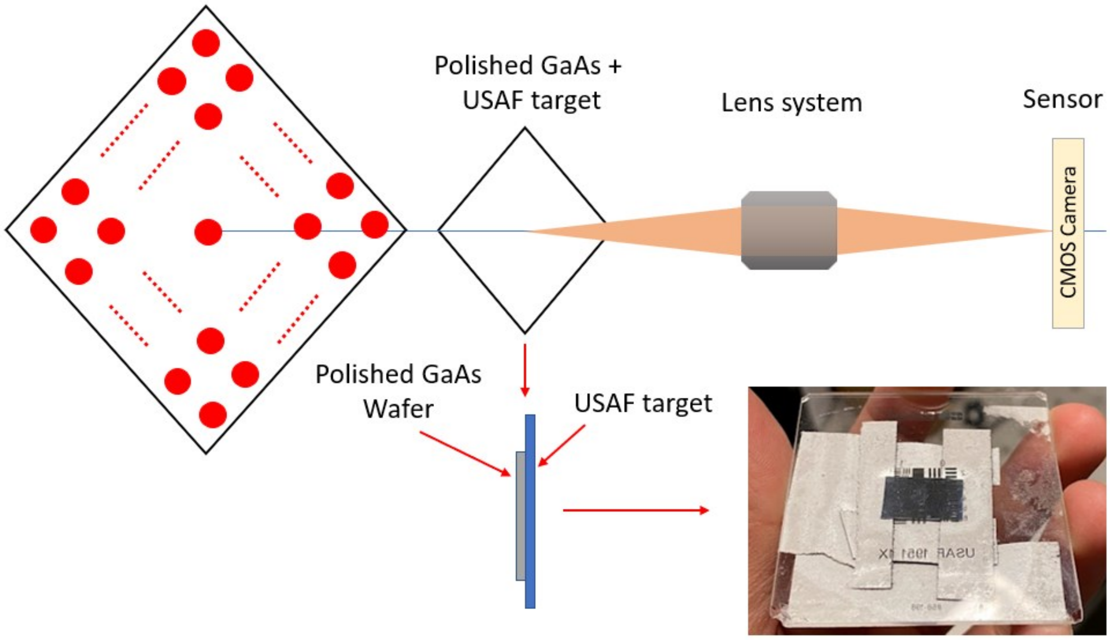


Figure 4-23 GaAs wafer combined with USAF target test case. The Polished GaAs wafer is 0.34mm thick.

5 Experimental results and discussion

5.1 Experiment with UV light

5.1.1 2015a-USAF target.

This experiment was performed on a 2015a-USAF sample. The dataset was taken under 17x17 UV lights, with the experimental parameters detailed in Chapter 4. The optimal $Z_{LED-sample}$ is 108mm, which is found by monitoring the convergence index, as illustrated in section 5.1.1.2. The calculated overlap is 65%. By using the proposed recovery process, the recovery results of the experiment are shown in Figure 5-1.

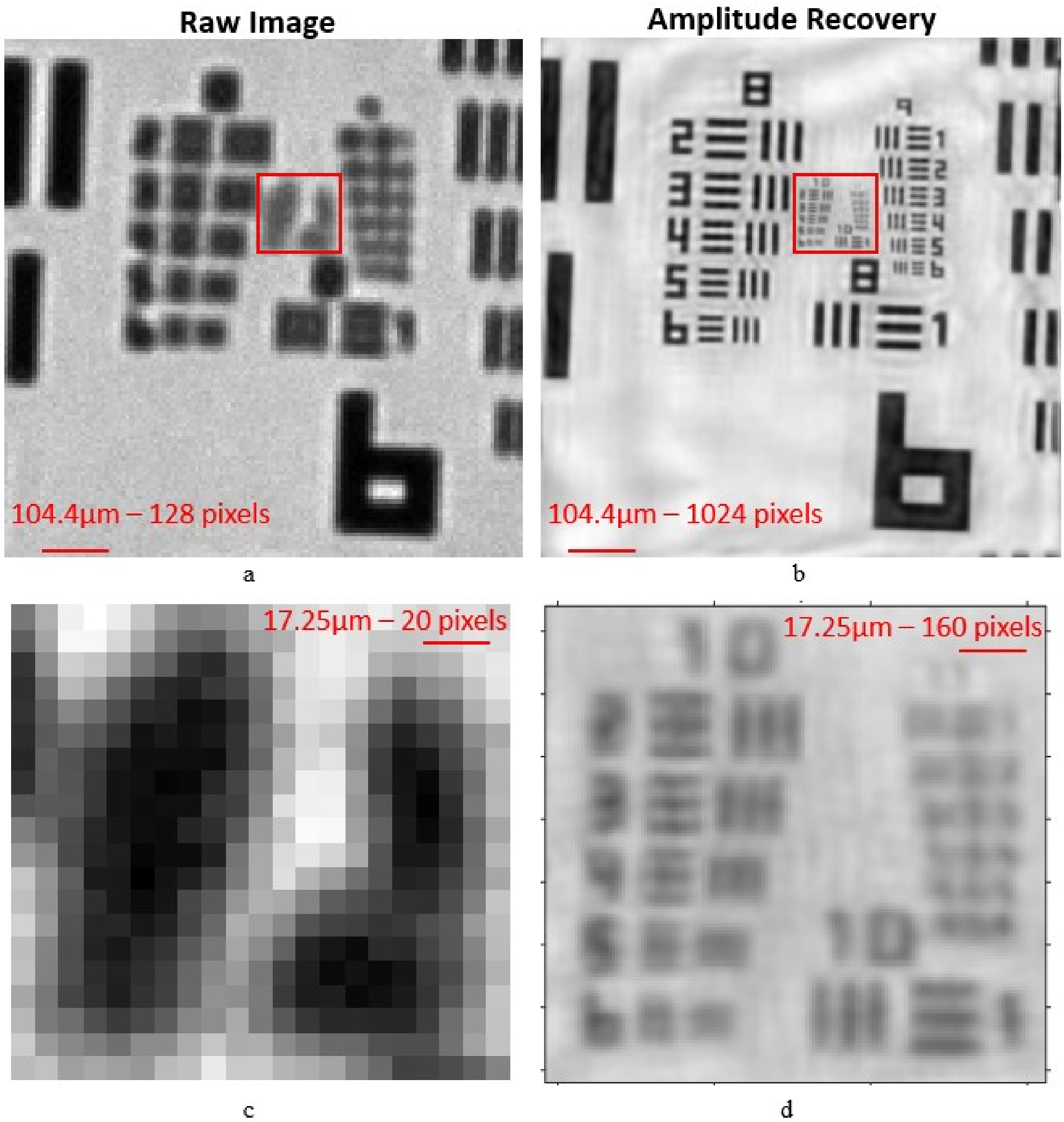


Figure 5-1 Results of the experiment with USAF sample, dataset of 17x17 UV LEDs, the optimal $Z_{LED-sample}$ is 108mm. a) Raw image under illumination of the central UV LED.

b) Amplitude of recovery high-resolution. c)-d) Zoomed images to the red-square area of raw and amplitude recovery images, respectively.

5.1.1.1 Maximum resolution of FPM with UV light

In comparison to the raw image in Figure 5-1a, the recovery image in Figure 5-1b is much clearer with higher resolution. In the raw image, it is difficult to distinguish all bars in groups 8 and 9 of the USAF target. However, in the recovery image, these bars are clearly visible. The bars of groups 8 and 9 in the recovery image appear sharply and well-define, which is consistent with the target. As shown in Figure 5-1d, the recovery resolution is at least 387nm, the element 10-3 of USAF target, according to Table 4-3. This matches the theoretical FPM resolution of 379nm, which is calculated by equation (*3-3*). This represents more than a 5-fold increase in resolution compared to traditional imaging methods. Table 5-1 provides the comparison between experimental and theoretical results. The resolution can be improved further If the dome shape is used with higher angles of illumination. In conclusion, the success of FPM combined with UV method is demonstrated with the experimental resolution matching the theoretical resolution.

Table 5-1 FPM-UV experimental results compared with theoretical resolution.

Highest USAF element resolved with FPM	Theoretical half-pitch resolution with FPM	Theoretical half-pitch resolution without FPM
10-3 (half-pitch: 387 nm)	379 nm	2000 nm

5.1.1.2 Finding the optimal LED-Sample distance by monitoring the convergence index

In this experiment, the LED-sample distance measured by ruler is about 106mm. However, the accuracy of this measurement is not guaranteed. To further investigate the relationship between the convergence index and the LED-sample distance, the convergence index was calculated for all $Z_{LED-sample}$ values range from 100mm to 115mm with a step size of 0.5mm. The findings are presented in Figure 5-2, where the convergence index is plotted against the LED-sample distance. It can be observed that the convergence index increases when the LED-sample distance increases from 100mm to 108mm. However, beyond this point, as the LED-sample distance continues to increase, the convergence index starts to decrease. Notably, the highest convergence index is observed at an LED-sample distance of 108mm. This indicates that the recovery process converges best at the LED-sample distance of 108mm. Therefore, it can be inferred that the optimal $Z_{LED-sample}$ corresponds to 108mm, based on the highest convergence achieved.

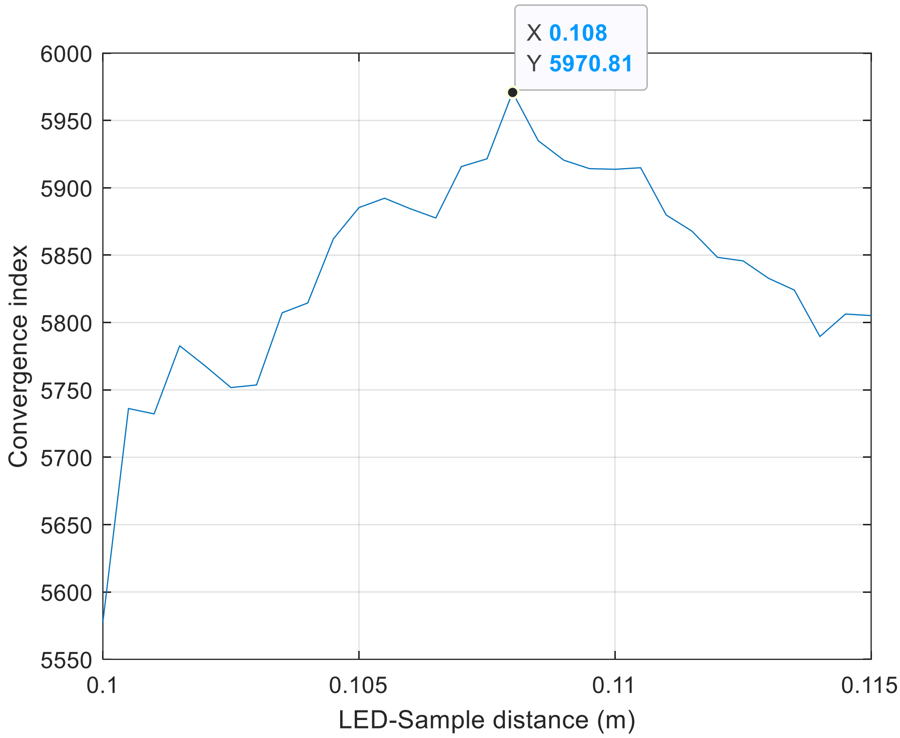


Figure 5-2 Convergence index versus LED-Sample distance of the experiment performed on a 2015a-USAF sample under the illumination of UV light.

5.1.1.3 Comparison of results between the Proposed Recovery Process and the EPRY Method-only Process

To evaluate the Proposed Recovery Process, we conducted a recovery process which uses only EPRY method. As can be seen in Figure 5-3, compare to the EPRY-only process result, the recovery image of the proposed recovery process is very smooth, which is the result of a good noise reduction method. However, it is common for images to contain some noise or artifacts, which can be caused by various factors such as stray light, optical aberrations, and imperfections in the LED matrix. In addition, the lens system used in this experiment was not specifically designed for UV wavelengths, which may have contributed to some stray light and reduced image quality. Improving the optical setup, such as by using a lens system optimized for UV wavelength to minimize the multi-reflection of UV light, can help further improve the quality of the recovery image.

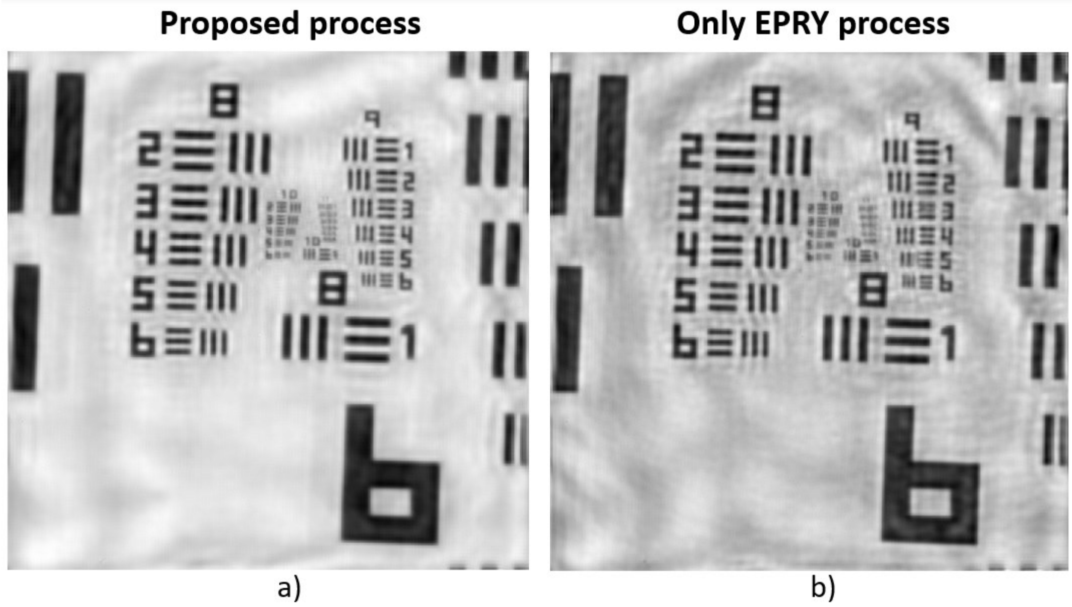


Figure 5-3 Recovery amplitudes from the same dataset of two recovery processes: a) Proposed recovery process, b) EPRY-only process.

5.1.1.4 Phase recovery

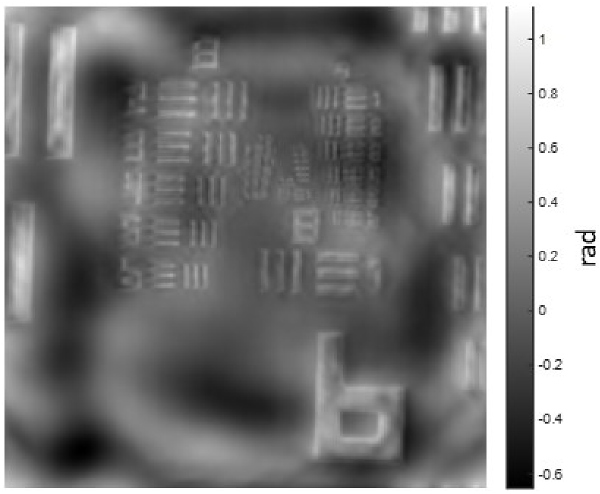


Figure 5-4 Phase recovery of the object.

Figure 5-4 shows the phase recovery of the object. The phase is almost zero for the object, which is relevant to the USAF sample as it is a very flat sample. The difference between the bar area and the background area in the phase recovery can be easily observed. This is because the bars are made of chrome metal, which blocks the electric field. As a result, the electric field after the bar is zero, and so the phase is undefined. It should be noted that the recovery method cannot reconstruct information that is not present in the raw data.

5.1.1.5 Recovery Fourier spectrum

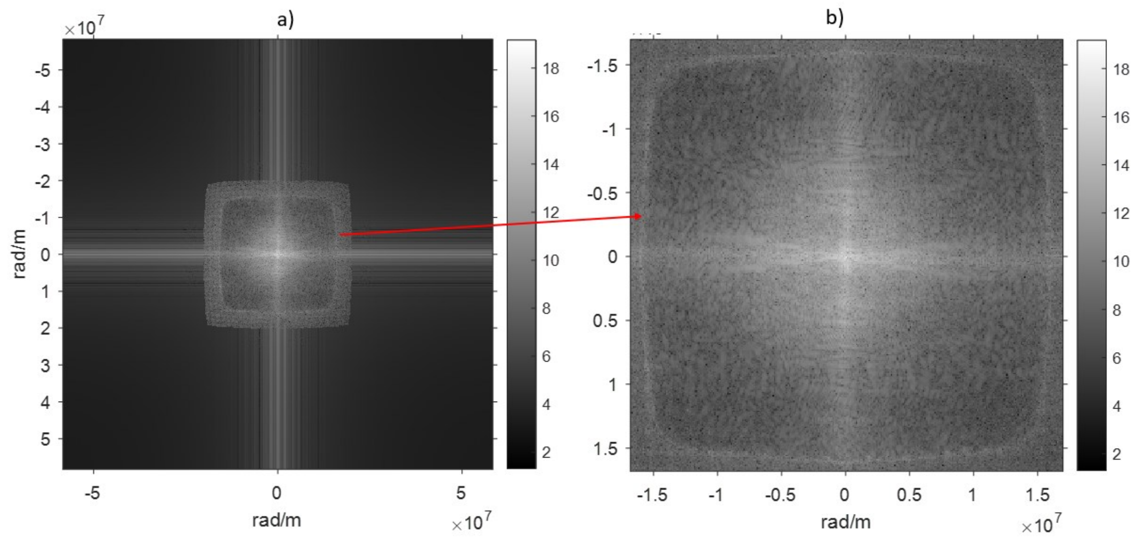


Figure 5-5 a) Recovered Fourier spectrum in log scale of the recovered complex object. b) Zoomed-in to the recovered Fourier spectrum.

In Figure 5-5, the logarithmic scale Fourier spectrum of the recovered complex object is presented. The spectrum appears smooth. This demonstrates that the algorithm effectively integrates information from all sub-regions of all illumination angles.

5.1.1.6 Pupil recovery

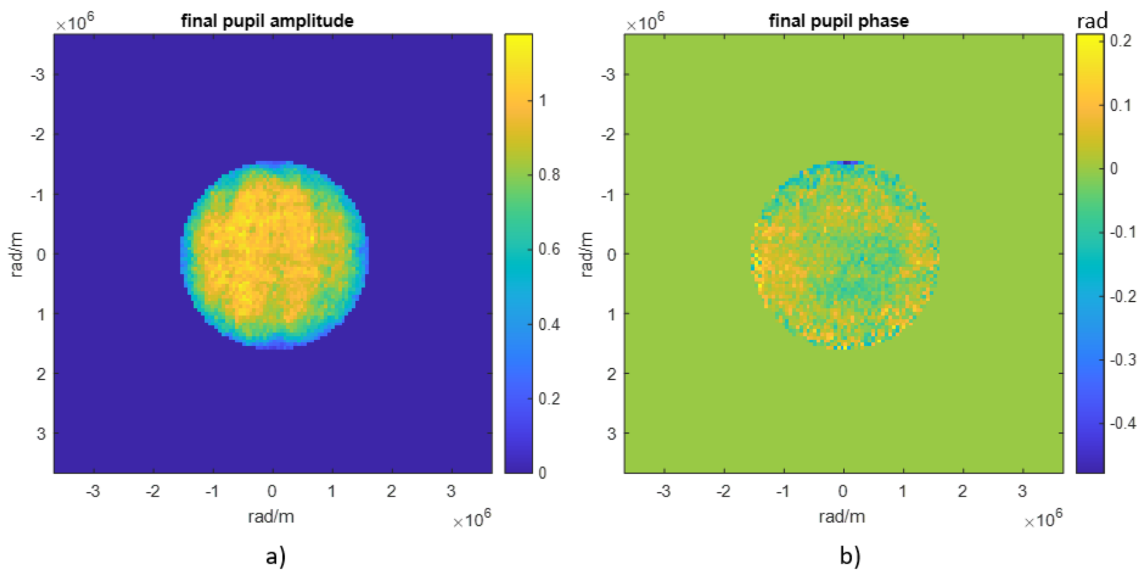


Figure 5-6 Recovery pupil function: a) Amplitude and b) Phase

In real experiments, imperfections in the lens system manifest as aberrations in the imaging system. The EPRY algorithm is capable of recovering the pupil function which represents these aberrations. Figure 5-6 illustrates the recovered pupil function of the USAF target. Notably, the pupil amplitude gradually decreases towards the edge, and the presence of spherical aberration can be noted in the pupil phase. This pupil recovery helps the high-resolution recovery converge faster by canceling the aberration error of the imaging system.

5.1.2 Bio samples

5.1.2.1 Blood cell sample

This experiment was performed on a blood cell sample, the dataset was taken under 17x17 UV lights, with the experimental parameters detailed in Chapter 4. Similar to section 5.1.1.2, the optimal $Z_{LED-sample}$ is 100.5mm, which is found by monitoring the convergence index, as illustrated in Figure 5-7. The calculated overlap is 63%. By using proposed recovery process, the experimental recovery results are shown in Figure 5-8.

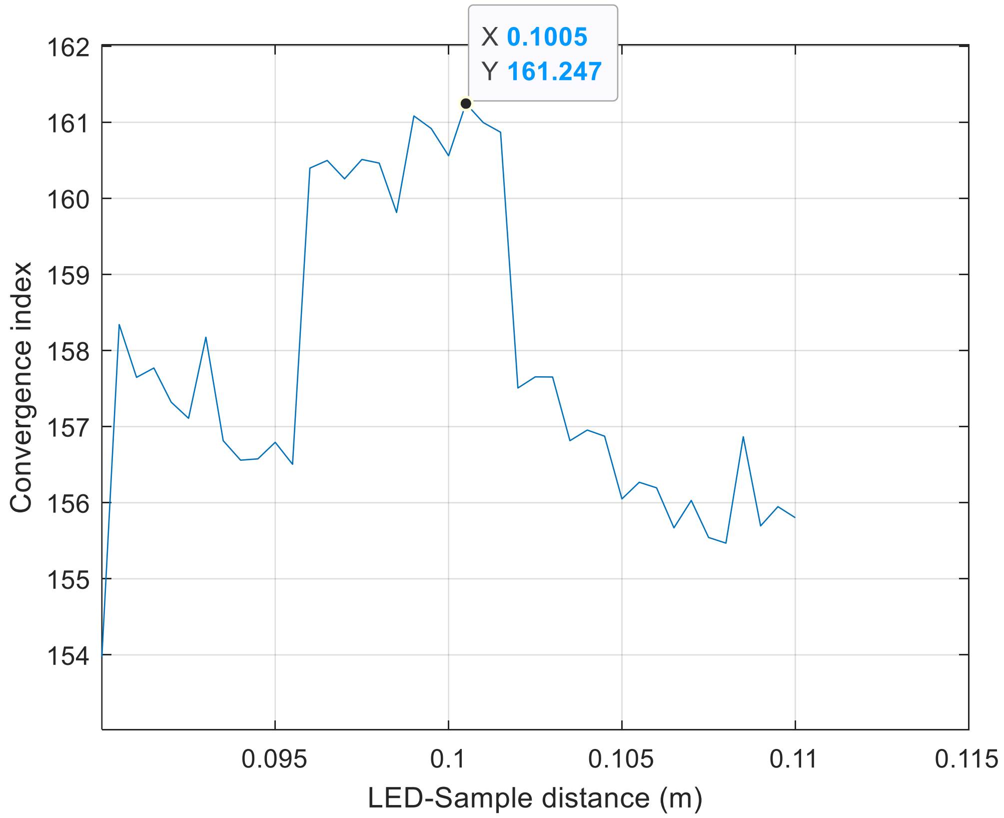


Figure 5-7 Convergence index versus LED-Sample distance of the experiment performed on a blood cell sample under the illumination of UV light.

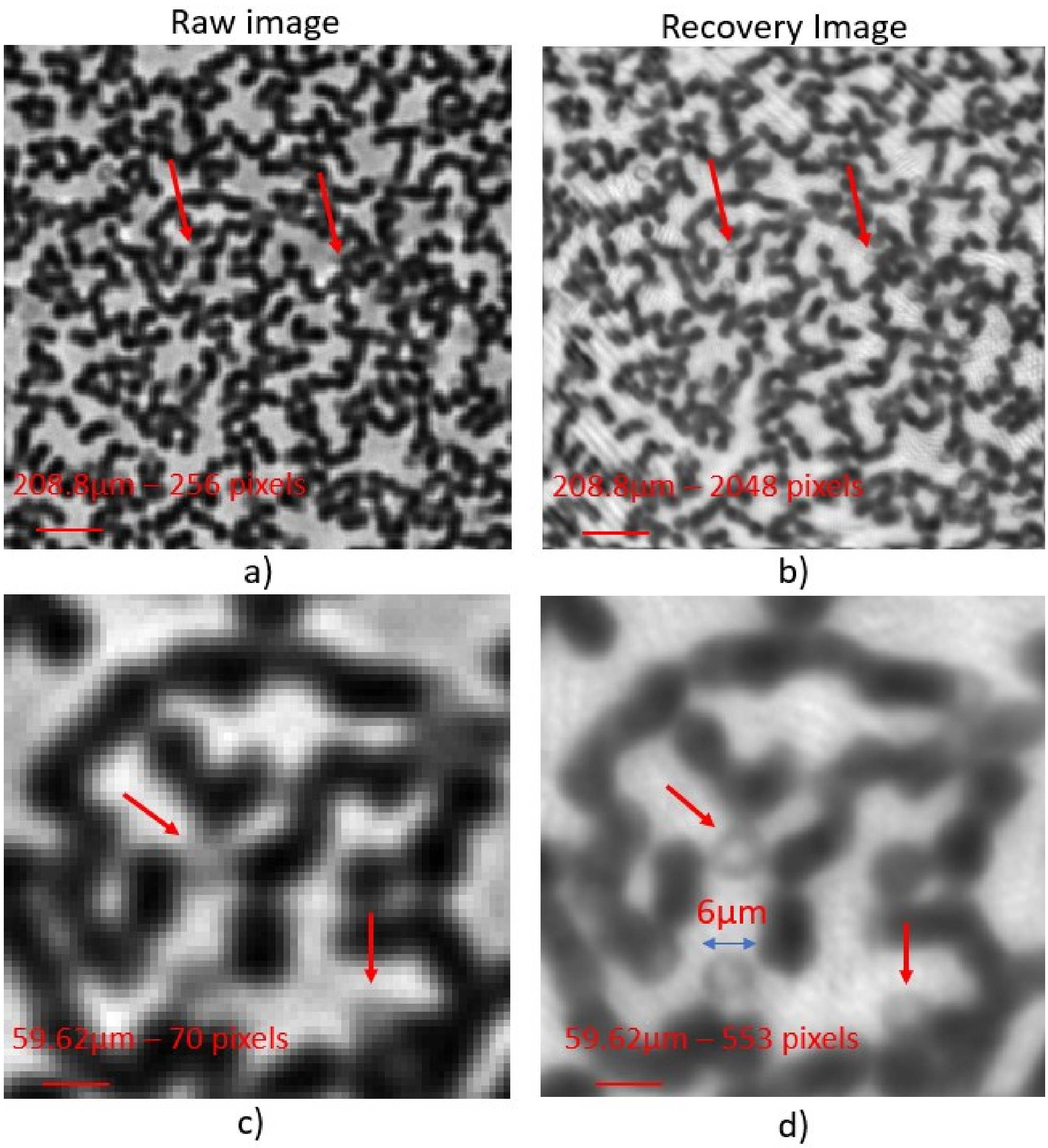


Figure 5-8 Results of the experiment with blood cell sample, dataset of 17x17 UV LEDs, the optimal $Z_{LED-sample}$ *is 100.5mm. a) Raw image under illumination of central UV LED. b) Amplitude of recovery high-resolution image. c)-d) Zoomed images of raw image and recovery image, respectively.*

Comparing the recovery high-resolution amplitude, in Figure 5-8b, with the raw image, in Figure 5-8a, the high-resolution amplitude is much clearer, with blood cells being more clearly observed. Additionally, as illustrated in Figure 5-8c-d in the area pointed by the red arrow, no blood cell is visible in the raw image, whereas a blood cell is visible in the recovery image, along with several other similar blood cells. The measured blood cell diameter is around 6µm.

As can be seen in phase recovery, Figure 5-9, the phase contrast between blood cells and background can be observed, which presents the height difference between them. Because the phase represents the height difference of the sample.

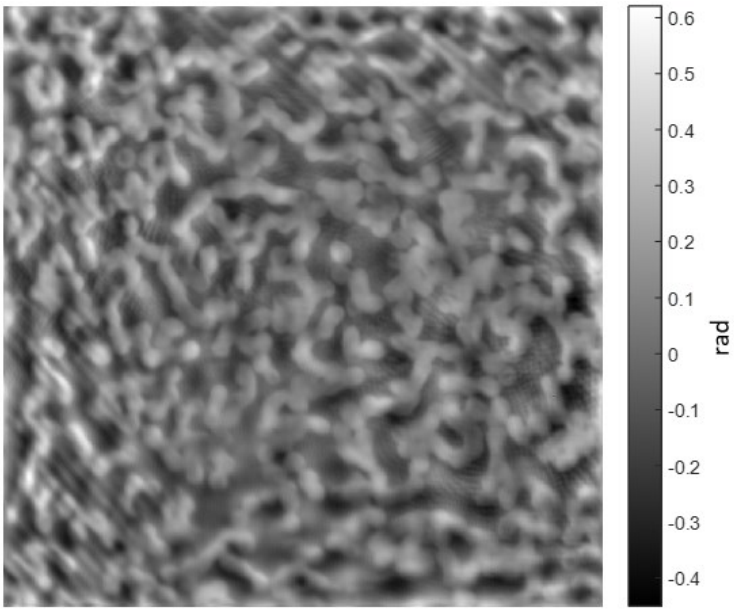


Figure 5-9 Recovery phase of the blood cell sample.

5.1.2.2 Cartilage sample

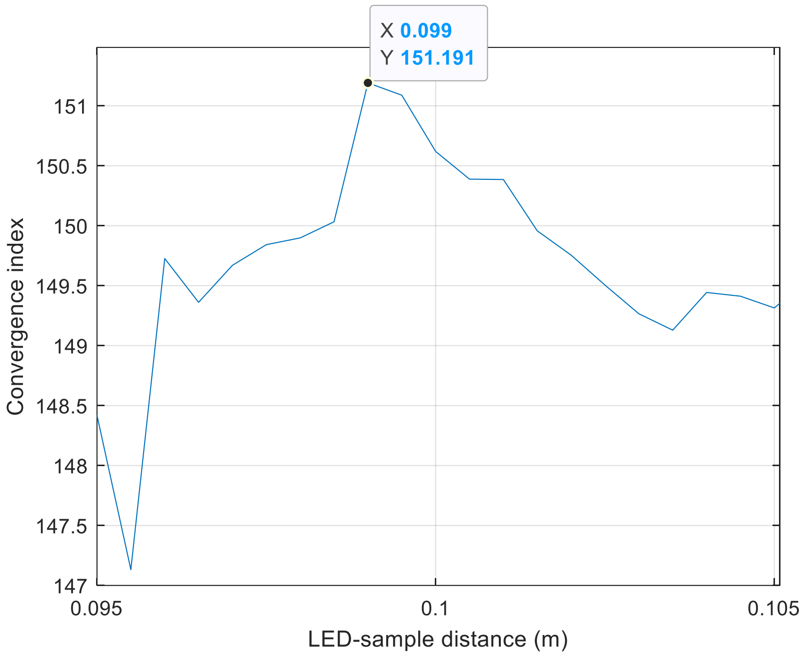


Figure 5-10 Convergence index versus LED-Sample distance of the experiment performed on a cartilage sample under the illumination of UV light.

This experiment was performed on a thin slice cartilage sample, the dataset was taken under 17x17 UV lights, with the experimental parameters detailed in Chapter 4. Similar to section 5.1.1.2, the optimal $Z_{LED-sample}$ is 99mm, which is found by monitoring the convergence index, as illustrated in Figure 5-10. The calculated overlap is 62%. By using the proposed recovery process, the recovery results of the experiment are shown in Figure 5-11.

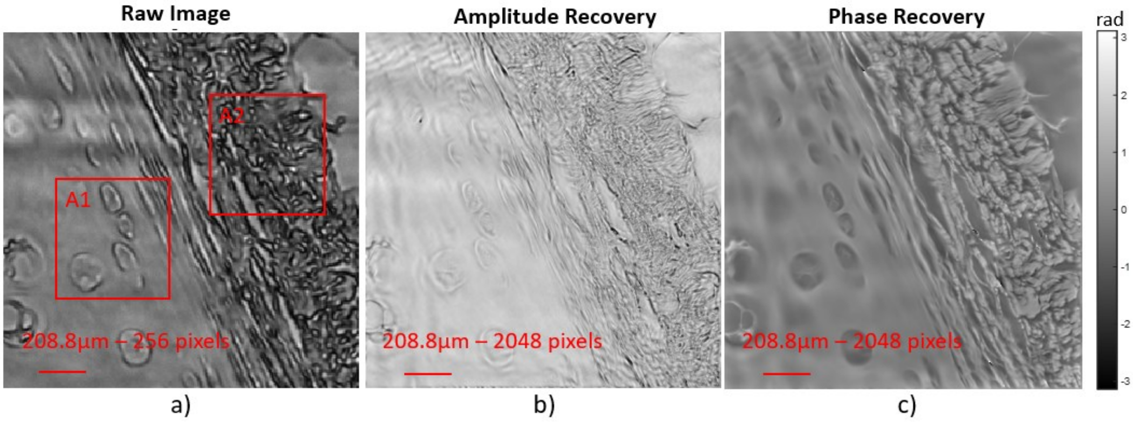


Figure 5-11 Results of the experiment with thin slice of cartilage, dataset of 17x17 UV LEDs, the optimal $Z_{LED-sample}$ is 99mm. a) Raw image under illumination of the central UV LED. b) Amplitude of recovery high-resolution. c) Recovery phase of the sample.

A thin slice of cartilage sample appears transparent under UV light due to its low absorption of UV radiation. As depicted in Figure 5-11b, the recovery of the amplitude of this transparent sample using FPM method is challenging. However, also due to transparency, the phase information is encoded in the dark-field images [1, Ch. 3]. Therefore, the phase recovery is clear and contains more information than the amplitude recovery. This is demonstrated in Figure 5-12, where the small features of cartilage are shown in the phase recovery. The phase recovery also presents the height difference with high contrast, providing a clear representation of the sample's structure.

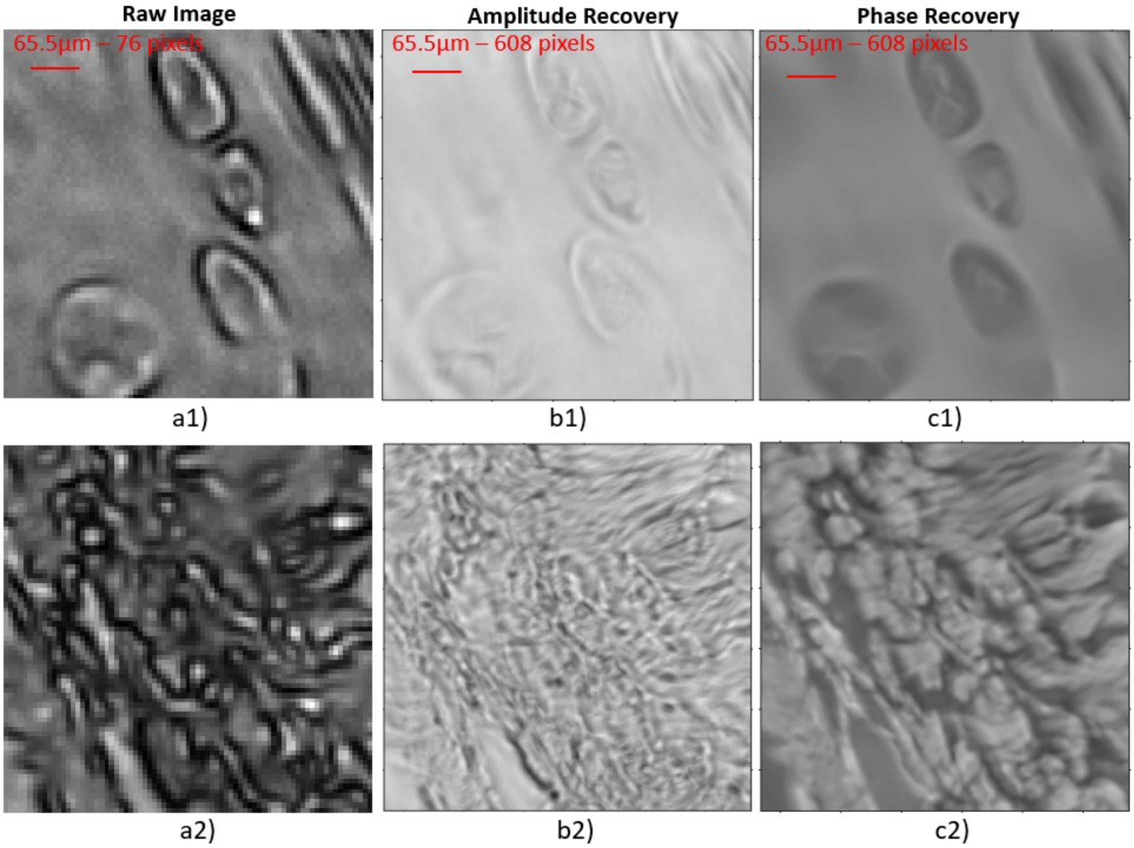


Figure 5-12 a1)-a2) are zoomed images to the red square area A1 and A2, respectively, of the raw image. b1)-b2) are zoomed images to red square area A1 and A2, respectively, of amplitude recovery image. c1)-c2) are zoomed images to red square area A1 and A2, respectively, phase recovery.

5.2 Experiment with IR light

5.2.1 2015a-USAF target sample

This experiment was performed on a 2015a-USAF sample. The dataset was taken under 17x17 IR light, with the experimental parameters detailed in Chapter 4. Similar to section 5.1.1.2, the optimal $Z_{LED-sample}$ is 104.5mm, which is found by monitoring the convergence index, as illustrated in Figure 5-13. The calculated overlap is about 64%. By using the proposed recovery process, the recovery results of the experiment are shown in Figure 5-14.

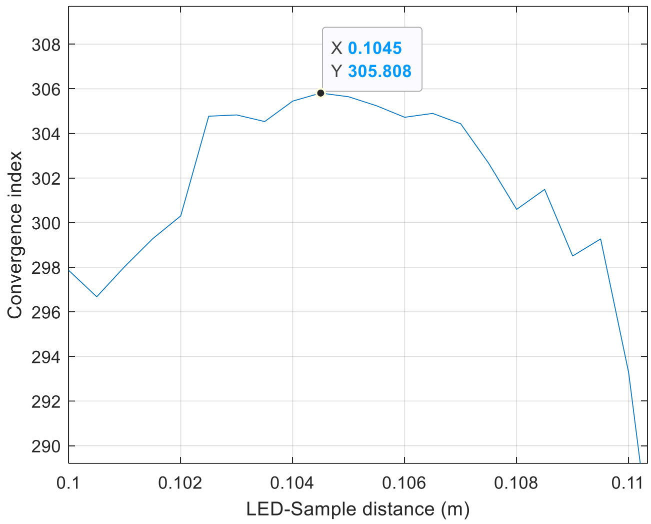


Figure 5-13 Convergence index versus LED-Sample distance of the experiment performed on USAF target under the illumination of IR light.

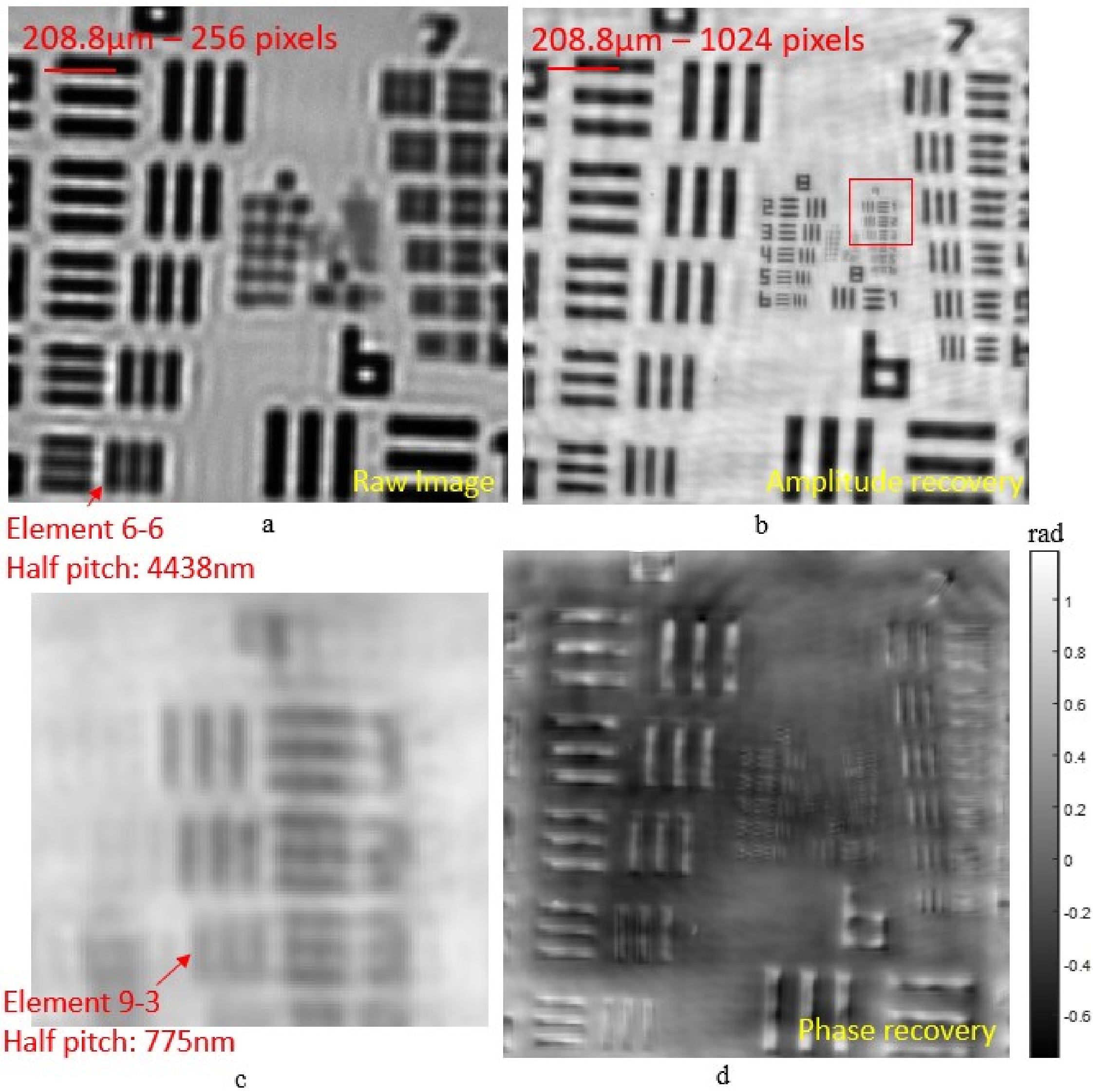


Figure 5-14 Results of the experiment with USAF sample, dataset of 17x17 IR LEDs, the optimal $Z_{LED-sample}$ is 104.5mm. a) Raw image under illumination of the central IR LED. b) Amplitude of recovery high-resolution. c) Zoomed image to the highest resolution, group 9, the red-square area in b). d) Recovery phase of the sample.

As shown in the raw image in Figure 5-14a, only the bars in group 6 are visible, while the periodic bars in groups 7 and 8 appear as blurred single squares. However, in the amplitude recovery image as shown in Figure 5-14b, the bars are clearly visible up to group 8. Furthermore, as shown in Figure 5-14c, a recovered image resolution equivalent to 775nm is achieved, with the ability to resolve USAF element 9-3. The recovery resolution is slightly better than the theoretical resolution of 863 nm, perhaps caused by experimental uncertainty. This result is significantly better than element 6-6, which is the highest resolvable element without FPM. Table 5-2 shows the comparison between the experimental result with theoretical resolution. In conclusion, the FPM combined with IR method is demonstrated with the experimental resolution matching the theoretical resolution.

Table 5-2 FPM-IR experimental results compared with theoretical resolution.

Highest USAF element resolved with FPM	Theoretical half-pitch resolution with FPM	Theoretical half-pitch resolution without FPM
9-3 (half-pitch: 775 nm)	863 nm	4450 nm

5.2.2 GaAs wafer sample

As discussed in section 4.4.4, Silicon has limited transparency under IR light with 890nm of wavelength due to its high absorption. Therefore, GaAs are used instead of Si to test the performance of FPM-IR.

This experiment was performed on a piece of GaAs wafer. The dataset was taken under 13x13 IR lights, with the experimental parameters detailed in Chapter 4. The optimal $Z_{LED-sample}$ was 100mm. The calculated overlap is about 62%. By using the proposed recovery process, the recovery results of the experiment are shown in Figure 5-15.

Figure 5-15a displays the raw image of the structure on the GaAs wafer under the central IR LED. This area of the GaAs wafer contains some periodic bars on the front side of the wafer, and some unknown structures on the back side of the wafer due to non-uniform polishing. To be clear, the back side of the wafer is the surface which is illuminated by

LED. On the other hand, front side of wafer is the surface which is captured by camera. This experiment focuses on recovering the periodic bars on the front side of the wafer.

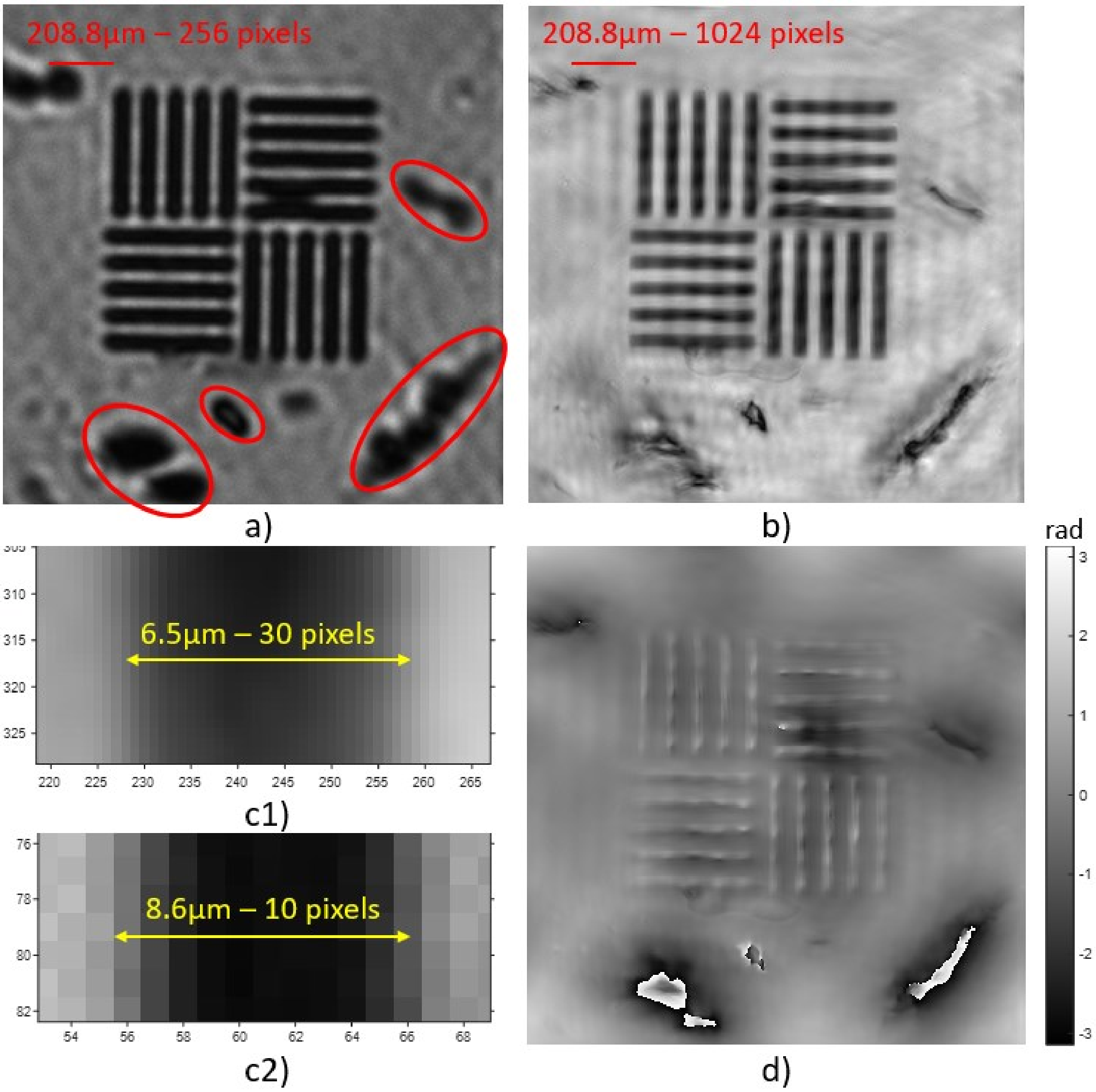


Figure 5-15 Results of experiment with a piece of GaAs wafer sample, dataset of 13x13 IR LEDs, the optimal $Z_{LED-sample}$ was 100mm. a) Raw image under illumination of the central IR LED. The areas in the red ovals are unknown structures from the back side of the wafer. b) Amplitude of recovery high-resolution. c1)-c2) Zooming images of recovery amplitude and the raw image, respectively, to measure the bar width. d) Phase recovery of the sample.

Figure 5-15b shows the recovered amplitude of the sample, where the periodic bars are observed much clearer and sharper than in the raw image. The bar's width measurement based on the recovered amplitude, as shown in Figure 5-15c1, is about 6.5µm. It is smaller than 8.6µm in width measurement based on the raw image, as shown in Figure 5-15c2.

Figure 5-15d displays the phase recovery. The phase recovery of almost the entire sample area is close to zero, except for some unknown structures from the back side of the wafer. This indicates that the front side of the GaAs wafer is flat.

This experimental result demonstrates the FPM combined with IR is applicable to see-through GaAs wafer and provide a clearer view of the features on it.

5.2.3 The sample created by combining of GaAs wafer and 1951-USAF target.

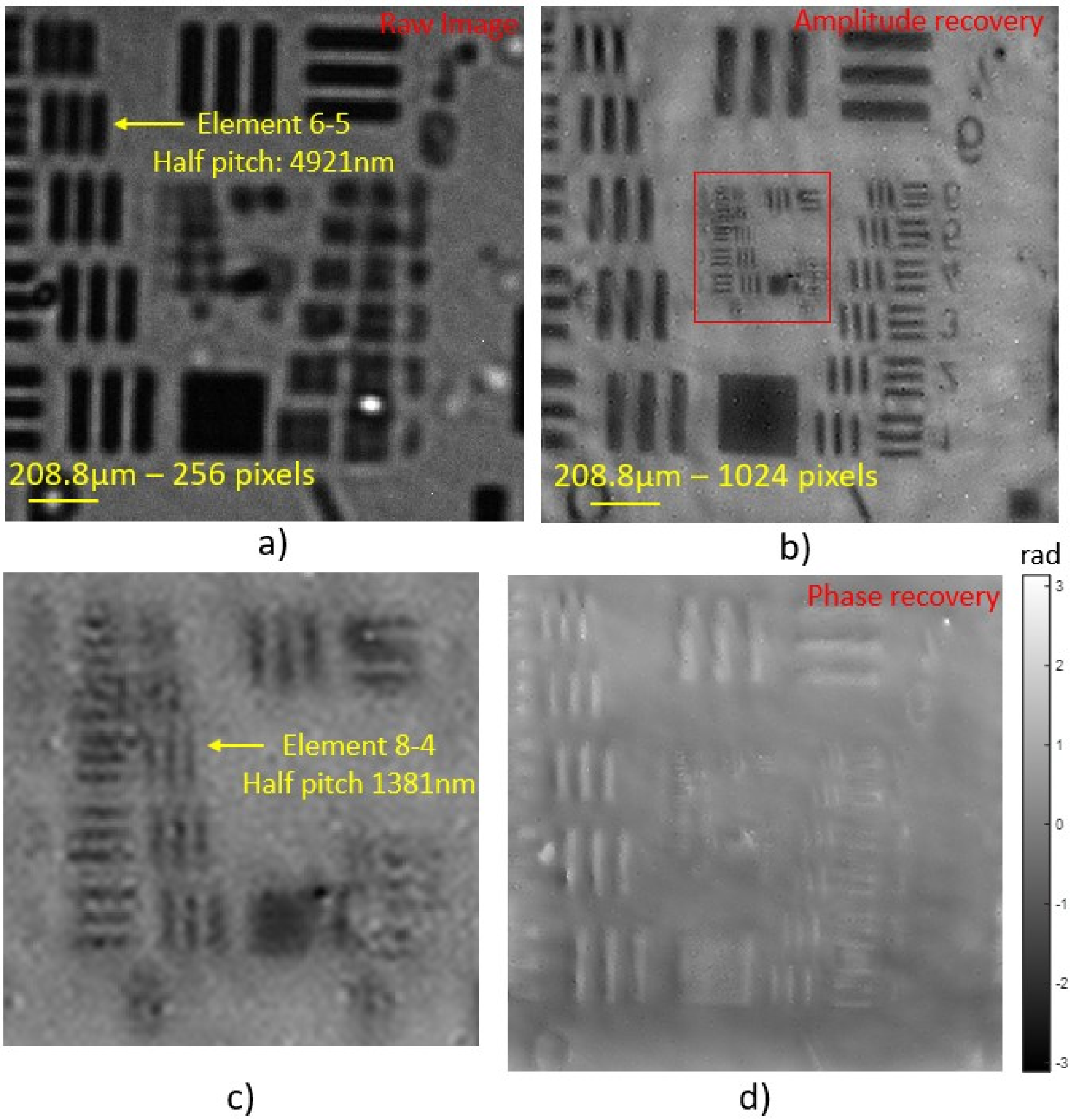


Figure 5-16 Results of the experiment with a sample created by combining of GaAs wafer and 1951-USAF target, dataset of 9x9 IR LED, the optimal $Z_{LED-sample}$ is 72mm. a) Raw image under illumination of the central IR LED. b) Amplitude of recovery high-

resolution. c) Zoomed images to red rectangle area of amplitude image, highest resolvable USAF element, element 8-4. d) Recovery phase of the sample

This experiment was performed on a sample created by combining of GaAs wafer and 1951-USAF target, as discussed in section 4.4.5. The dataset was taken under 9x9 IR light, with the experimental parameters detailed in Chapter 4. The optimal $Z_{LED-sample}$ was 72mm. The calculated overlap is about 49%. By using the proposed recovery process, the recovery results of the experiment are shown in Figure 5-16.

To optimize the signal-to-noise ratio, we used a 9x9 IR light matrix size and kept the LED-sample distance closer. Because, with a high angle of illumination the dark field light becomes very low, leading to very weak signals obtained with a high level of noise by the camera. The IR-LED matrix size of 9x9 is a reasonable matrix size with enough signal to recover. Additionally, the closer the LED is to the sample, the stronger the signal that can be obtained after passing through the sample.

The raw image under the central IR LED on the matrix is shown in Figure 5-16a. The highest USAF element that can be observed is 6-5. Figure 5-16b shows the amplitude of high-resolution recovery. As shown in the recovery image, the bars of group 7 are accurately reconstructed. The highest resolution is at least 1381nm, which is equivalent to element 8-4 of the 1951-USAF target, as shown in Figure 5-16c. This is a significant improvement compared to element 6-5 of the raw image. Once again, this experiment demonstrates that FPM-IR is applicable for seeing through GaAs wafers.

Table 5-3 Comparison of the results of the FPM-IR experiment on a sample created by combining of GaAs wafer and 1951-USAF target, with theoretical resolution.

Highest USAF element resolved with FPM	Theoretical half-pitch resolution with FPM	Theoretical-resolved USAF element with FPM
8-4 (half-pitch: 1381 nm)	1070 nm	8-6 (half-pitch: 1096nm)

Table 5-3 compares the experimental results to the theoretical results, which shows that the experimental resolution with FPM-IR is worse than the theoretical resolution. There are two reasons behind this result. First, as the illumination angle increases, the transmission coefficient decreases due to no anti-reflection coatings, leading to a weaker signal after passing through the sample. Secondly, the lens system and camera used in this experiment were not designed for the IR wavelength of 890nm. As shown in Figure

4-7 in section 4.1.4, the quantum efficiency of the camera at 890nm is only 14%, which is insufficient to obtain a good signal. These two reasons make the signal-to-noise ratio very small. Improving the lens system and camera may help to solve this problem.

6 Difficulties and tips.

6.1 Sensor thermal expansion

One challenge encountered when implementing Fourier Ptychography Microscopy (FPM) is sensor thermal expansion. The problem arises due to the extended operation time required to capture a large number of images. Additionally, the exposure time for dark-field images increases as the illumination reaches the corners of the matrix. Thus, these two reasons make the sensor operate for a long time while taking measurements, leading to thermal expansion. This problem is hardly recognized when examining the entire image at its maximum size of 14.1x7.55mm. However, it becomes noticeable when observing a smaller area (approximately 208x208µm) over 15 minutes. During this time, the captured video gradually drifts to another region of the sample. And after resting the camera, it captures the original area once again. This issue significantly impacts the results, especially considering the high-frequency waves and the expected resolution of approximately 400nm. Even a tiny change can lead to bad results.

- Tips: To address this problem, my approach is to run the camera for approximately 30 minutes before taking measurements. This allows the sensor to heat up and reach the stable state, which reduces the potential for thermal expansion.

7 Conclusion and Future work

In this work, Fourier Ptychography Microscopy (FPM) is implemented successfully with ultra-violet (UV) and infrared (IR) light. We developed custom-built drive electronics to control a planar 17x17 LED matrix using discrete UV and IR LEDs (with wavelengths of 400±15 nm and 890±40 nm, respectively). Additionally, software was developed for taking measurements and calibrating the LED's position in the optical system. The lens system used for the experiment has a NA of 0.1, 4x magnification, and a pixel size of 3.45µm. Finally, we proposed a recovery process, which combines two noise reduction methods, intensity correction, finding the optimal LED-sample distance, and the EPRY algorithm.

FPM with UV and IR are demonstrated with the experimental resolution matching the theoretical prediction. As illustrated in the experiment on the 2015a-USAF target using UV light, the resolution of the recovered image is at least 387 nm with USAF element 10-3 being resolved, which matches the theoretical resolution of 379 nm. This represents more than a 5-fold increase in resolution compared to the conventional method. As demonstrated in section 5.2.1, in the experiment on USAF target using IR light, a recovered image resolution equivalent to 775 nm is achieved, with the ability to resolve USAF element 9-3. The recovery resolution is slightly better than the theoretical resolution of 863 nm, perhaps caused by experimental uncertainty. Moreover, this result is significantly better than element 6-6, which is the highest resolvable element without FPM.

The FPM-UV and FPM-IR are also implemented on the practical samples such as biological samples for UV light and GaAs wafers for IR light. Section 5.1.2.1 illustrates the FPM-UV applicability on the high-contrast biological sample such as the blood cell sample, in which we find more blood cells than the normal microscope. Moreover, the phase recovery of FPM may help to observe a transparent sample's structure such as a thin slice of rabbit cartilage or pure phase targets.

The FPM-IR applicability of seeing through GaAs wafer is demonstrated in sections 5.2.2 and 5.2.3. This method provides a clear view of periodic bars on the GaAs surface. The highest resolution of FPM-IR when the light passes through the GaAs wafer is 1381nm, which is a big improvement compared to the 4450nm resolution of the traditional optical microscope without FPM. This is the first step for building a new set-up to see through the Silicon wafer, which is valuable in Silicon wafer dying applications.

7.1 Future work

In the near future, to improve the FPM-UV resolution, our custom drive electronics can be combined with a dome-shape LED array [15] to increase the illumination angle. Hence, we can raise the highest illumination angle to 70 degrees, resulting in an increased NA of illumination (NA_{ill}) of 0.94. This setup could enable a significant resolution improvement up to 192 nm for FPM-UV.

Another approach to enhance resolution is by utilizing a UV LED with a shorter wavelength. For instance, employing a 200 nm wavelength can yield a two-fold improvement compared to the current setup with 400 nm wavelengths. However, the 200 nm wavelength UV-LED is quite expensive, which is impractical for a matrix with 289 LEDs. To overcome this challenge, we propose mounting only one LED on a motorized stage, allowing us to control LED's position by adjusting the stage position programmatically.

In this thesis, we successfully implemented FPM on a GaAs sample using IR light with a wavelength of 890 nm and a standard camera designed for the visible range of light. To further advance FPM-IR for viewing through a Silicon wafer, a new setup needs to be developed. Since Silicon exhibits high absorption, longer wavelengths such as 1500 nm are required to pass through it. Additionally, a specialized camera (such as the InGaAs camera, which is capable of receiving light signals at 1500 nm wavelengths, should be employed to ensure compatibility with the system.

References

[1] Guoan Zheng, *Fourier ptychographic imaging: a MATLAB® tutorial*. San Rafael, California: Morgan Claypool Publishers, 2016.

[2] G. Zheng, R. Horstmeyer, and C. Yang, "Wide-field, high-resolution Fourier ptychographic microscopy," *Nature Photon*, vol. 7, no. 9, pp. 739–745, Sep. 2013, doi: 10.1038/nphoton.2013.187.

[3] X. Ou, R. Horstmeyer, C. Yang, and G. Zheng, "Quantitative phase imaging via Fourier ptychographic microscopy," *Opt. Lett., OL*, vol. 38, no. 22, pp. 4845–4848, Nov. 2013, doi: 10.1364/OL.38.004845.

[4] C. Yang, R. Horstmeyer, and G. Zheng, "Fourier ptychographic microscopy: A method for pushing standard microscope beyond its physical limitations," 2013. doi: 10.1364/acpc.2013.ath3j.4.

[5] X. Ou, R. Horstmeyer, G. Zheng, and C. Yang, "High numerical aperture Fourier ptychography: principle, implementation and characterization," *Opt. Express*, vol. 23, no. 3, p. 3472, Feb. 2015, doi: 10.1364/OE.23.003472.

[6] L. Tian, X. Li, K. Ramchandran, and L. Waller, "Multiplexed coded illumination for Fourier Ptychography with an LED array microscope," *Biomed. Opt. Express*, vol. 5, no. 7, p. 2376, Jul. 2014, doi: 10.1364/BOE.5.002376.

[7] X. Ou, G. Zheng, and C. Yang, "Embedded pupil function recovery for Fourier ptychographic microscopy," *Opt. Express*, vol. 22, no. 5, p. 4960, Mar. 2014, doi: 10.1364/OE.22.004960.

[8] L. Tian and L. Waller, "3D intensity and phase imaging from light field measurements in an LED array microscope," *Optica*, vol. 2, no. 2, p. 104, Feb. 2015, doi: 10.1364/OPTICA.2.000104.

[9] M. G. Mayani, N. Hussain, D. W. Breiby, and M. N. Akram, "Improved resolution Fourier ptychography scheme using oil-filled dome-shaped LED array," *Electronics Letters*, vol. 58, no. 20, pp. 762–764, 2022, doi: 10.1049/ell2.12596.

[10] W. H. Hayt, *Engineering electromagnetics*, 8th ed. New York: McGraw-Hill, 2012.

[11] J. W. Goodman, *Introduction to fourier optics*, Fourth edition. New York: W.H. Freeman, Macmillan learning, 2017.

[12] "Charge-coupled device," *Wikipedia*. Apr. 14, 2023. Accessed: May 06, 2023. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Charge-coupled_device&oldid=1149860070

[13] "Nyquist–Shannon sampling theorem," *Wikipedia*. Apr. 12, 2023. Accessed: May 05, 2023. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Nyquist%E2%80%93Shannon_sampling_theorem&oldid=1149432927

[14] D. G. Voelz, *Computational fourier optics: a MATLAB tutorial*. Bellingham, Wash: SPIE Press, 2011.

[15] M. G. Mayani, N. Hussain, D. W. Breiby, and M. N. Akram, "Improved resolution Fourier ptychography scheme using oil-filled dome-shaped LED array," *Electronics Letters*, vol. 58, no. 20, pp. 762–764, 2022, doi: 10.1049/ell2.12596.

[16] S. Dong, Z. Bian, R. Shiradkar, and G. Zheng, "Sparsely sampled Fourier ptychography," *Opt. Express*, vol. 22, no. 5, p. 5455, Mar. 2014, doi: 10.1364/OE.22.005455.

[17] O. Bunk, M. Dierolf, S. Kynde, I. Johnson, O. Marti, and F. Pfeiffer, "Influence of the overlap parameter on the convergence of the ptychographical iterative engine,"

Ultramicroscopy, vol. 108, no. 5, pp. 481–487, Apr. 2008, doi: 10.1016/j.ultramic.2007.08.003.

[18] Q. Liu *et al.*, “Effect of spatial spectrum overlap on Fourier ptychographic microscopy,” *J. Innov. Opt. Health Sci.*, vol. 10, no. 02, p. 1641004, Mar. 2017, doi: 10.1142/S1793545816410042.

[19] P. Thibault, M. Dierolf, A. Menzel, O. Bunk, C. David, and F. Pfeiffer, “High-Resolution Scanning X-ray Diffraction Microscopy,” *Science*, vol. 321, no. 5887, pp. 379–382, Jul. 2008, doi: 10.1126/science.1158573.

[20] A. M. Maiden and J. M. Rodenburg, “An improved ptychographical phase retrieval algorithm for diffractive imaging,” *Ultramicroscopy*, vol. 109, no. 10, pp. 1256–1262, Sep. 2009, doi: 10.1016/j.ultramic.2009.05.012.

[21] Z. Bian, S. Dong, and G. Zheng, “Adaptive system correction for robust Fourier ptychographic imaging,” *Opt. Express*, vol. 21, no. 26, p. 32400, Dec. 2013, doi: 10.1364/OE.21.032400.

[22] L. Hou, H. Wang, J. Wang, and M. Xu, “Background-noise Reduction for Fourier Ptychographic Microscopy Based on an Improved Thresholding Method,” *Current Optics and Photonics*, vol. 2, no. 2, pp. 165–171, Apr. 2018, doi: 10.3807/COPP.2018.2.2.165.

[23] P. G. BERGAN and R. W. CLOUGH, “Convergence Criteria for Iterative Processes,” *AIAA Journal*, May 2012, doi: 10.2514/3.50313.

[24] “STLED524 - Intelligent matrix LED display driver - STMicroelectronics.” https://www.st.com/en/power-management/stled524.html (accessed May 10, 2023).

[25] “MT0380-UV-A,” *Digi-Key Electronics*. https://www.digikey.no/no/products/detail/marktech-optoelectronics/MT0380-UV-A/4214613 (accessed May 10, 2023).

[26] “OP265A Components | Optoelectronics | Products | TT Electronics.” https://www.ttelectronics.com/products/sensors/optoelectronics/components/op265a (accessed May 10, 2023).

[27] “2X, 0.08 NA, Ultra Compact Objective | Edmund Optics.” https://www.edmundoptics.com/p/2x-0-08na-ultra-compact-objective/38500/ (accessed May 10, 2023).

[28] “4X Extension Tube for #36-280 and #36-281 | Edmund Optics.” https://www.edmundoptics.com/p/4x-extension-tube-for-10mm-lenses/38506/ (accessed May 10, 2023).

[29] “Industrial USB 3.0 Cameras, 1" Sensors - PL-D799,” *Pixelink*. https://pixelink.com/products/industrial-cameras/usb-30/1-sensors/pl-d799/ (accessed May 10, 2023).

[30] “2015a USAF – Ready Optics.” https://readyoptics.com/product/2015a/ (accessed May 10, 2023).

[31] “3&quote; x 3&quote; Positive, USAF 1951 Hi-Resolution Target | Edmund Optics.” https://www.edmundoptics.com/p/3-x-3-positive-1951-usaf-hi-resolution-target/19543/ (accessed May 10, 2023).

List of figures and charts

List of tables

List of abbreviations

Abbreviations	Meaning
FPM	Fourier Ptychography Microscopy
UV	Ultraviolet
IR	Infrared
FPM-UV	Fourier Ptychography Microscopy combined with UV light
FPM-IR	Fourier Ptychography Microscopy combined with IR light
NA	Numerical of Aperture
EPRY	Embedded Pupil Function Recovery
Si	Silicon
GaAs	Gallium arsenide
CTF	Coherent transfer function
CCD	Charge-coupled device
CMOS	Complementary Metal-Oxide Semiconductor
LED	Light-emitting diode
GPIO	General-purpose input/output
UART	Universal asynchronous receiver-transmitter
SPI	Serial Peripheral Interface
GUI	Graphical user interface
InGaAs	Indium gallium arsenide

Appendix A: Conference paper

During the thesis work, we have written and submitted a conference paper to the Optica Imaging Congress, hosted by Optica (OSA). The conference title is "Fourier ptychography microscopy with home-built UV and IR LED matrix and drive electronics." The conference requires 35 words of abstract and a maximum of two pages.

The conference is attached on the following page.

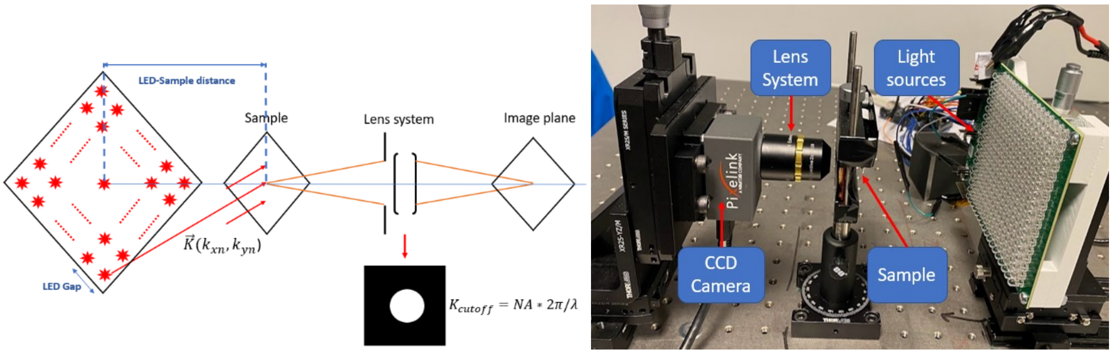
Fourier ptychography microscopy with home-built UV and IR LED matrix and drive electronics

Huy Duong Gia[1], Mahdieh Gholamimayani[1], Dag Werner Breiby[1,2], Mohammad Nadeem Akram[1]
[1]Department Micro-Nano System technology, University of South-Eastern Norway, Raveien 215 NO-3184 Borre, Norway
[2]Department of Physics, Norwegian University of Science and Technology (NTNU), Høgskoleringen 5, 7491 Trondheim, Norway
**Email: mna@usn.no*

Abstract: A home-built 17x17 LED matrix using discrete LEDs (λ=400nm and 890nm) and drive electronics is developed. Fourier Ptychographic microscopy is demonstrated with experimental resolution matching the theoretical predictions.

1. Introduction

Fourier Ptychography microscopy (FPM) is a coherent computational imaging method that surpasses the diffraction limitation of the optics and Nyquist limit of the digital camera chip. In ordinary imaging, the optics limit the high-frequency signal of the light field, and the phase of the light field is lost as well. The FPM captures high-frequency signal content by illuminating the sample at different angles and recovers extended complex Fourier spectrum by iteratively combining all the sub-spectral. This method effectively increases the numerical aperture (NA) of the optics [1]. A schematic draw of FPM model is shown in Fig. 1(a).



a) b)

Fig. 1. a) Schematic draw of FPM model; b) FPM setup with 17x17 UV (or IR) LED matrix and drive electronics

Under monochromatic light, imaging system is linear in complex amplitude [2]. The relation between the light field at the object plane and the image plane in frequency domain is given by (1):

$$G_{OUT}(k_x, k_y) = G_{in}(k_x - k_{xn}, k_y - k_{yn}) \times H(k_x, k_y) \tag{1}$$

where $H(k_x, k_y)$ is the coherent transfer function (CTF) of the lens system, $G_{OUT}(k_x, k_y)$ and $G_{in}(k_x, k_y)$ are the Fourier transform of light fields at the image plane and sample plane respectively, $G_{in}(k_x - k_{xn}, k_y - k_{yn})$ is the shifted Fourier transform of light field at the sample plane under an off-axis LED illumination. The idea of FPM algorithm is taking images under different angles of illumination, thus shifting the higher frequencies allow them to pass through the low-pass CTF. Then iteratively stitching the obtain images in frequency domain enforcing the spectral and CTF constraints. Eventually, we recover a wider range complex spectrum of the object, which means the resolution of recovered complex sample will be higher. With UV light, the resolution will naturally be higher due to shorter wavelength, while with infrared light, this method can be applied to see through Silicon wafers which are opaque in the visible waveband.

2. Methods

To implement FPM with UV and IR light, custom drive electronics and PCB are designed to control the discrete LEDs with matrix size 17x17, as shown in Fig. 1(b). Three STLED524 driver chips are used in the custom drive electronics to control 289 constant light sources (equivalent to 17x17 LED matrix). These 3 chips received command from Arduino Board to turn LEDs on. Then, the LEDs are sequentially lit to illuminate the sample and images are captured. A CMOS camera is used to capture the raw image corresponding to each LED position. A photograph of the experimental setup is shown in Fig. 1(b), and experimental parameter is shown in Table 1.

Table 1. Experiment parameters

Objective lens numerical aperture NA	0.1	LED gap	6 mm
Magnification	4x	LED to sample distance	101 mm
CMOS pixel size	3.45x3.45 µm	Illumination NA	0.42

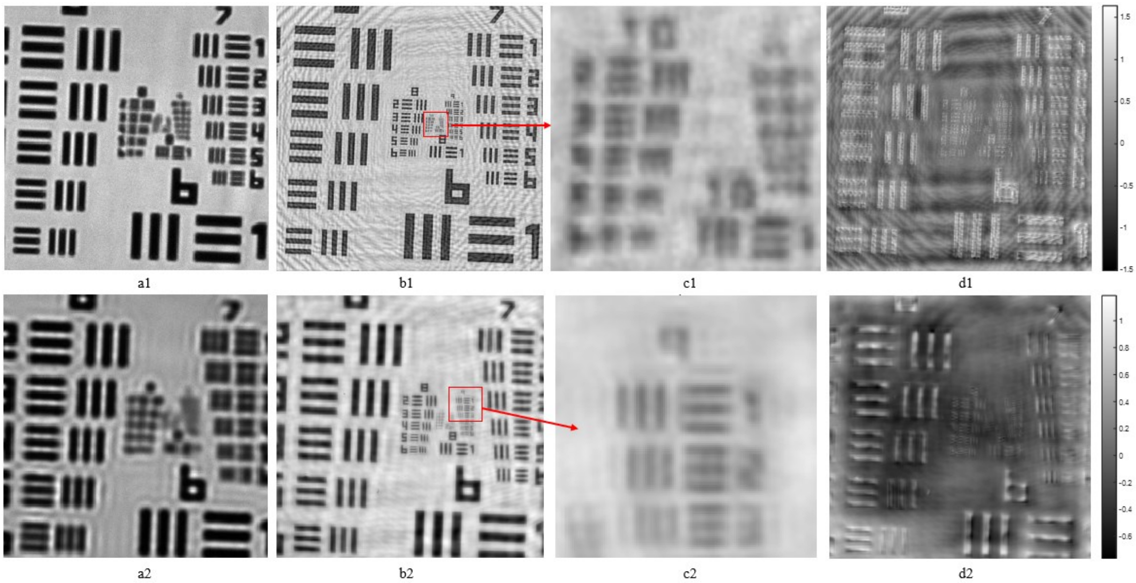


Fig. 2. a1)-a2) Raw images under illumination of center UV and IR LEDs, respectively, image size 256x256; b1) Recovered image with UV light, image size 2048x2048; b2) Recovered image with IR light, image size 1024x1024; c1)-c2) Zoom-in images to highest resolution of UV and IR light, respectively; d1)-d2) Recovered phase with UV and IR, respectively.

3. Results and discussion:

Table 2: FPM Experimental Results

Wavelength λ	**Highest USAF element resolved with FPM**	**Theoretical half pitch resolution with FPM**	**Theoretical half pitch resolution without FPM**
UV - 400 nm	10-2 (half pitch: 435 nm)	379 nm	2000 nm
IR - 890 nm	9-3 (half pitch: 775 nm)	844 nm	4450 nm

The experiment with UV light (Fig. 2(a1) - (d1)) demonstrates a more than 4.6-fold increase in resolution compared to traditional imaging methods. The resolution can be improved further with higher angles of illumination. The experiment resolution is slightly worse than the expected resolution due to non-uniform intensity of LEDs, LED position uncertainty in the matrix, which can be minimized by adaptive system correction for robust FPM [3]. Additionally, the results are also affected by stray light because the used lens system is not designed for UV or IR wavelength. As shown in Fig. 2(a2) - (d2) with FPM and IR light, USAF element 9-3 is resolved. This is significantly better than element 6-6, which is the highest resolvable element without FPM. This work may lead to improved wide field-of-view, high resolution IR microscopes in Si wafer packaging applications, and high-resolution phase-contrast UV microscopy. We gratefully acknowledge funding from the Norwegian Research Council, NANO2021 (project 272148) and FRINATEK (project 275182).

4. References

[1] Guoan Zheng, Fourier ptychographic imaging: a MATLAB® tutorial. San Rafael, California: Morgan Claypool Publishers, 2016.

[2] J. W. Goodman, Introduction to fourier optics, Fourth edition. New York: W.H. Freeman, Macmillan learning, 2017.

[3] Z. Bian, S. Dong, and G. Zheng, "Adaptive system correction for robust Fourier ptychographic imaging," Opt. Express, vol. 21, no. 26, p. 32400, Dec. 2013, doi: 10.1364/OE.21.032400.

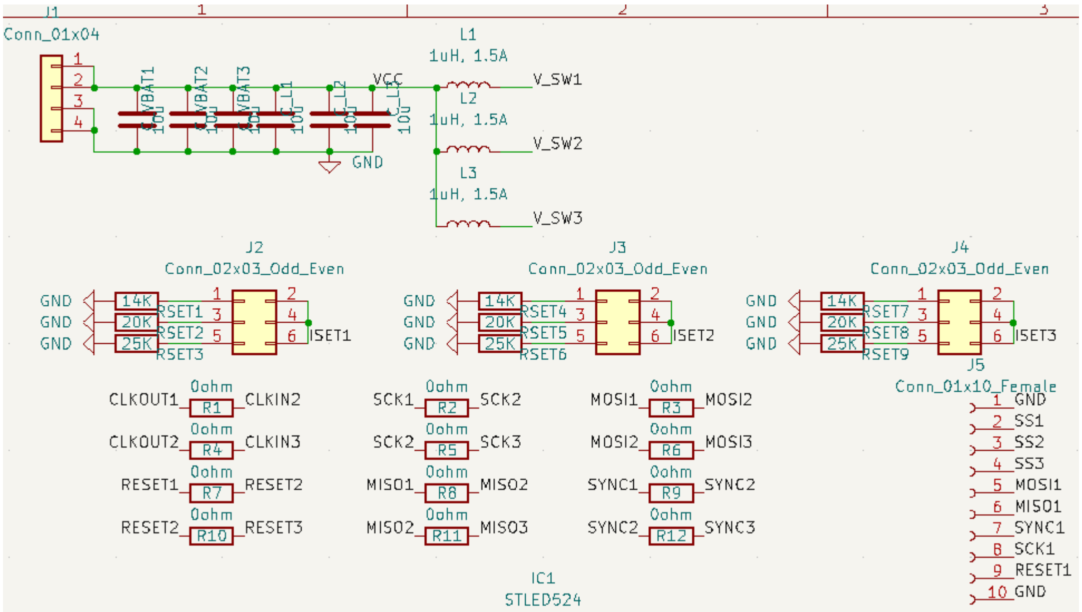
Appendix B: Arduino Firmware

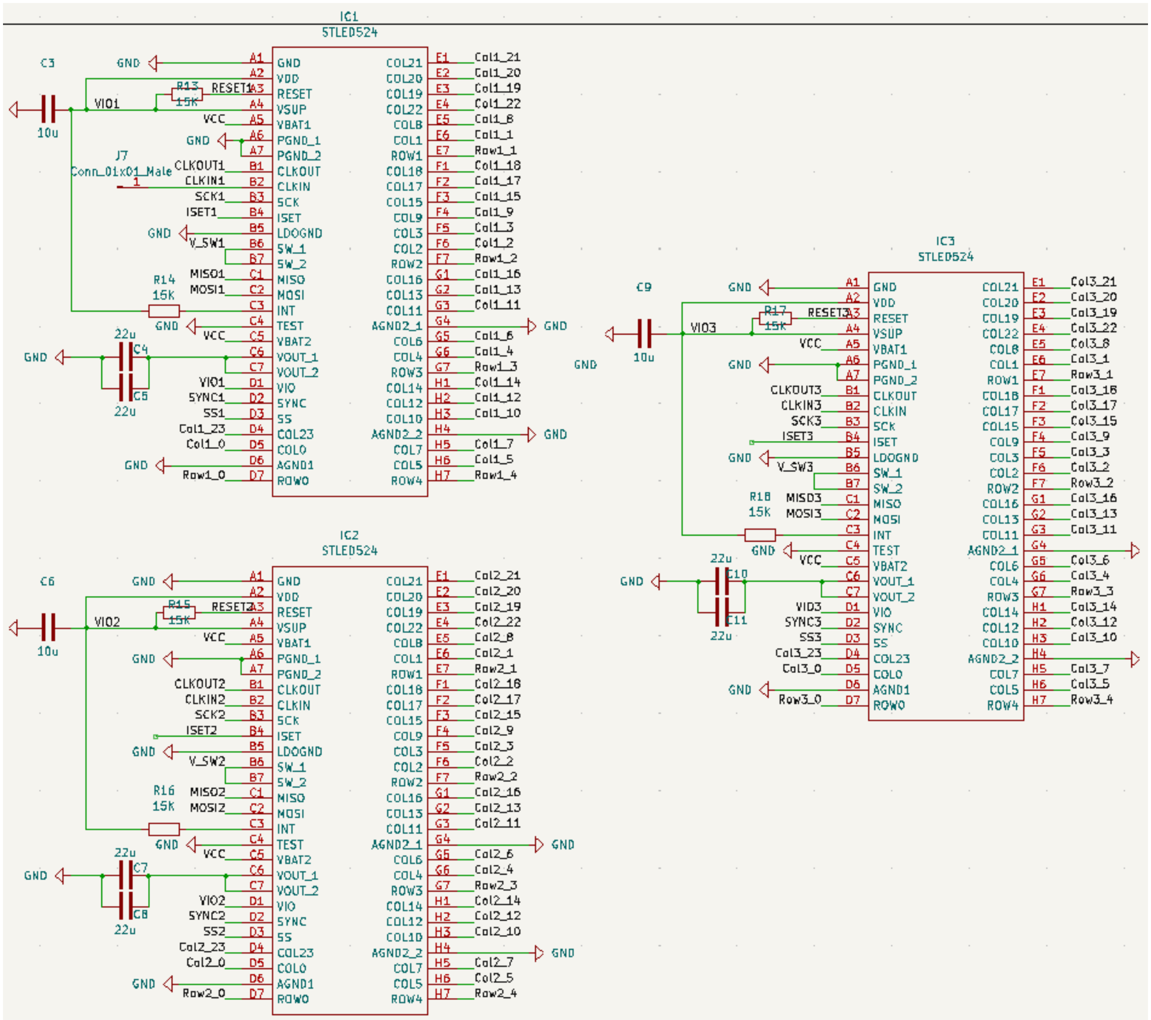
For a better view when reading code, please visit my GitHub page to read the Arduino Firmware programming, which controls the 17x17 LEDs matrix. The link is given below: https://github.com/HuyDuongGia/STLED524-Driver-for-17x17-LED-matrix---Constant-light/blob/main/STLED524_17x17_LED_matrix.ino

Appendix C: Custom drive electronic schematic design

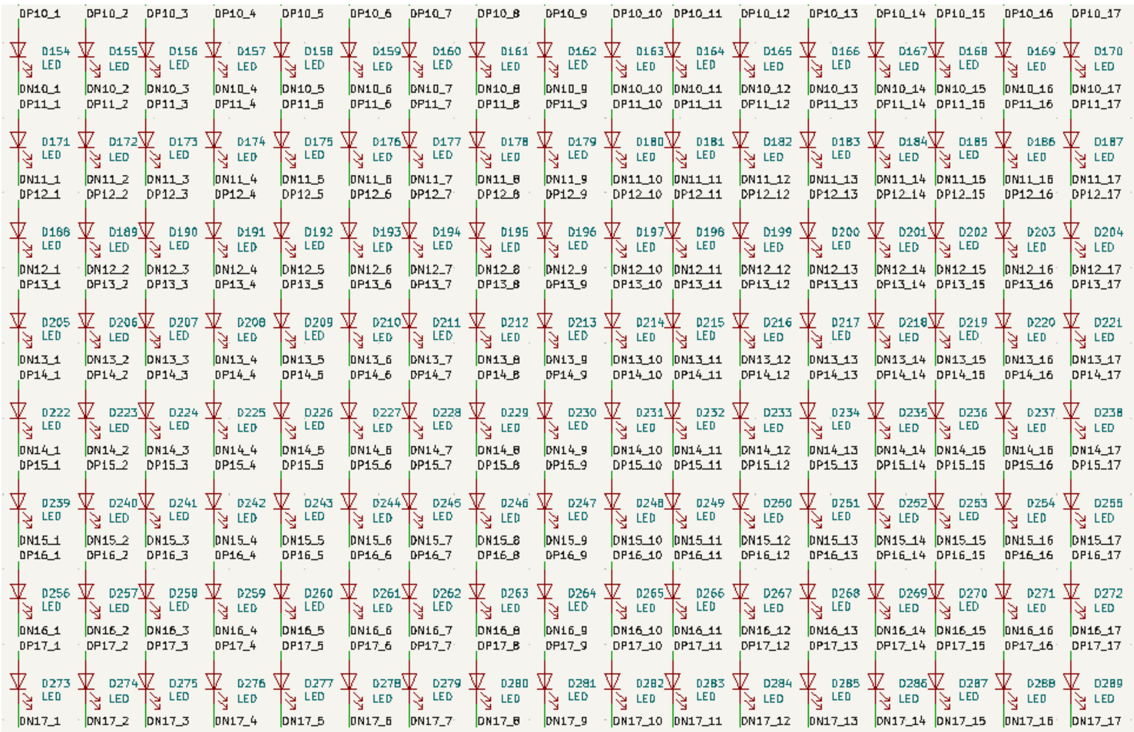
The custom drive electronic schematic and PCB are designed by Kicad version 6.

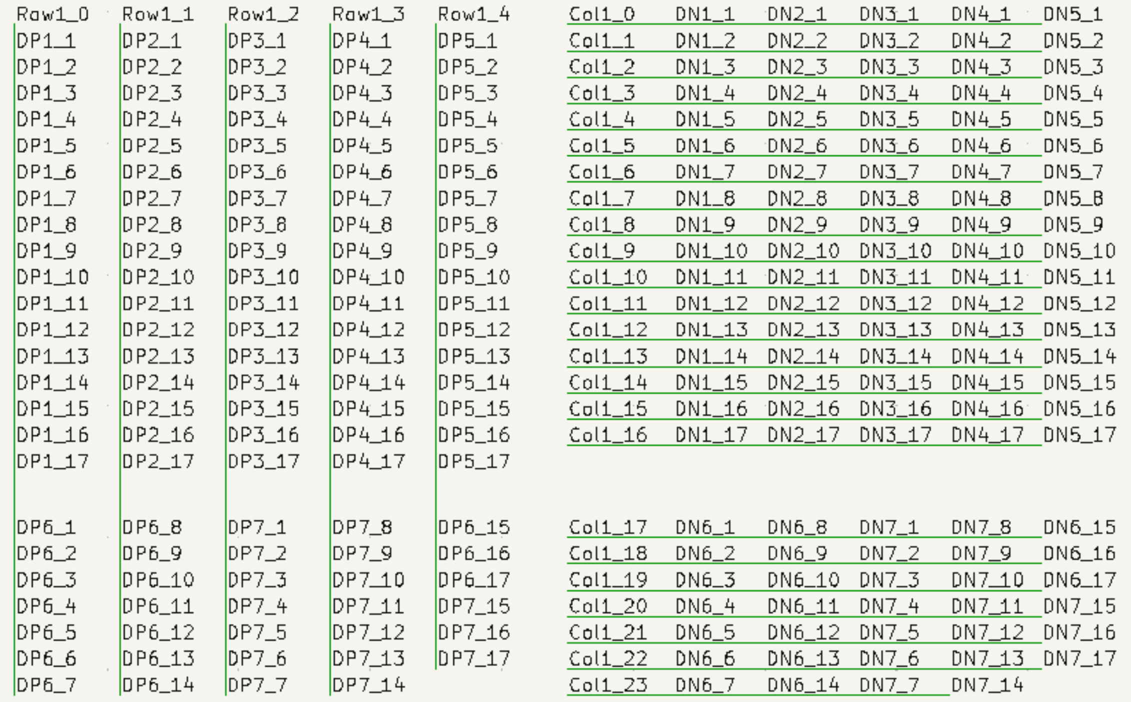
- Schematic design

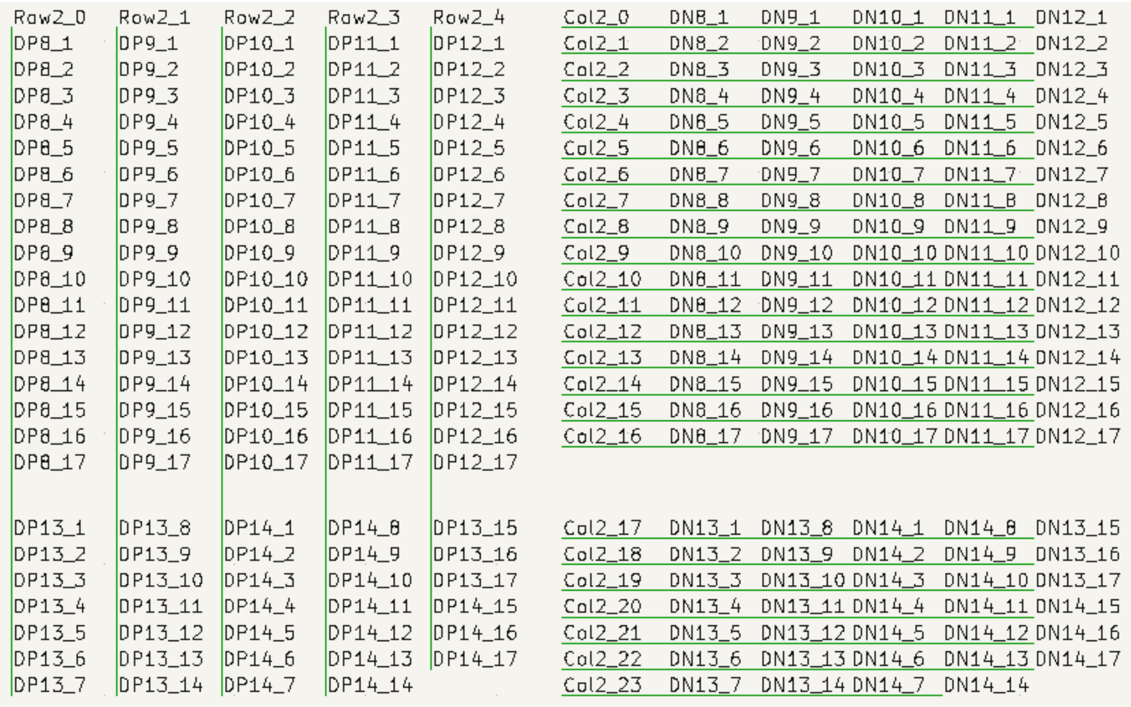


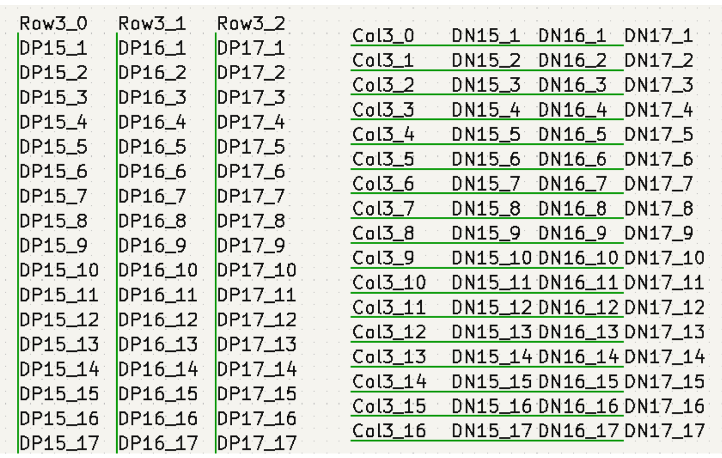



DP1_1 DP1_2 DP1_3 DP1_4 DP1_5 DP1_6 DP1_7 DP1_8 DP1_9 DP1_10 DP1_11 DP1_12 DP1_13 DP1_14 DP1_15 DP1_16 DP1_17
D1 D2 D3 D4 D5 D6 D7 D8 D9 D10 D11 D12 D13 D14 D15 D16 D17
LED
DN1_1 DN1_2 DN1_3 DN1_4 DN1_5 DN1_6 DN1_7 DN1_8 DN1_9 DN1_10 DN1_11 DN1_12 DN1_13 DN1_14 DN1_15 DN1_16 DN1_17
DP2_1 DP2_2 DP2_3 DP2_4 DP2_5 DP2_6 DP2_7 DP2_8 DP2_9 DP2_10 DP2_11 DP2_12 DP2_13 DP2_14 DP2_15 DP2_16 DP2_17
D18 D19 D20 D21 D22 D23 D24 D25 D26 D27 D28 D29 D30 D31 D32 D33 D34
DN2_1 DN2_2 DN2_3 DN2_4 DN2_5 DN2_6 DN2_7 DN2_8 DN2_9 DN2_10 DN2_11 DN2_12 DN2_13 DN2_14 DN2_15 DN2_16 DN2_17
DP3_1 DP3_2 DP3_3 DP3_4 DP3_5 DP3_6 DP3_7 DP3_8 DP3_9 DP3_10 DP3_11 DP3_12 DP3_13 DP3_14 DP3_15 DP3_16 DP3_17
D35 D36 D37 D38 D39 D40 D41 D42 D43 D44 D45 D46 D47 D48 D49 D50 D51
DN3_1 DN3_2 DN3_3 DN3_4 DN3_5 DN3_6 DN3_7 DN3_8 DN3_9 DN3_10 DN3_11 DN3_12 DN3_13 DN3_14 DN3_15 DN3_16 DN3_17
DP4_1 DP4_2 DP4_3 DP4_4 DP4_5 DP4_6 DP4_7 DP4_8 DP4_9 DP4_10 DP4_11 DP4_12 DP4_13 DP4_14 DP4_15 DP4_16 DP4_17
D52 D53 D54 D55 D56 D57 D58 D59 D60 D61 D62 D63 D64 D65 D66 D67 D68
DN4_1 DN4_2 DN4_3 DN4_4 DN4_5 DN4_6 DN4_7 DN4_8 DN4_9 DN4_10 DN4_11 DN4_12 DN4_13 DN4_14 DN4_15 DN4_16 DN4_17
DP5_1 DP5_2 DP5_3 DP5_4 DP5_5 DP5_6 DP5_7 DP5_8 DP5_9 DP5_10 DP5_11 DP5_12 DP5_13 DP5_14 DP5_15 DP5_16 DP5_17
D69 D70 D71 D72 D73 D74 D75 D76 D77 D78 D79 D80 D81 D82 D83 D84 D85
DN5_1 DN5_2 DN5_3 DN5_4 DN5_5 DN5_6 DN5_7 DN5_8 DN5_9 DN5_10 DN5_11 DN5_12 DN5_13 DN5_14 DN5_15 DN5_16 DN5_17
DP6_1 DP6_2 DP6_3 DP6_4 DP6_5 DP6_6 DP6_7 DP6_8 DP6_9 DP6_10 DP6_11 DP6_12 DP6_13 DP6_14 DP6_15 DP6_16 DP6_17
D86 D87 D88 D89 D90 D91 D92 D93 D94 D95 D96 D97 D98 D99 D100 D101 D102
DN6_1 DN6_2 DN6_3 DN6_4 DN6_5 DN6_6 DN6_7 DN6_8 DN6_9 DN6_10 DN6_11 DN6_12 DN6_13 DN6_14 DN6_15 DN6_16 DN6_17
DP7_1 DP7_2 DP7_3 DP7_4 DP7_5 DP7_6 DP7_7 DP7_8 DP7_9 DP7_10 DP7_11 DP7_12 DP7_13 DP7_14 DP7_15 DP7_16 DP7_17
D103 D104 D105 D106 D107 D108 D109 D110 D111 D112 D113 D114 D115 D116 D117 D118 D119
DN7_1 DN7_2 DN7_3 DN7_4 DN7_5 DN7_6 DN7_7 DN7_8 DN7_9 DN7_10 DN7_11 DN7_12 DN7_13 DN7_14 DN7_15 DN7_16 DN7_17
DP8_1 DP8_2 DP8_3 DP8_4 DP8_5 DP8_6 DP8_7 DP8_8 DP8_9 DP8_10 DP8_11 DP8_12 DP8_13 DP8_14 DP8_15 DP8_16 DP8_17
D120 D121 D122 D123 D124 D125 D126 D127 D128 D129 D130 D131 D132 D133 D134 D135 D136
DN8_1 DN8_2 DN8_3 DN8_4 DN8_5 DN8_6 DN8_7 DN8_8 DN8_9 DN8_10 DN8_11 DN8_12 DN8_13 DN8_14 DN8_15 DN8_16 DN8_17
DP9_1 DP9_2 DP9_3 DP9_4 DP9_5 DP9_6 DP9_7 DP9_8 DP9_9 DP9_10 DP9_11 DP9_12 DP9_13 DP9_14 DP9_15 DP9_16 DP9_17
D137 D138 D139 D140 D141 D142 D143 D144 D145 D146 D147 D148 D149 D150 D151 D152 D153
DN9_1 DN9_2 DN9_3 DN9_4 DN9_5 DN9_6 DN9_7 DN9_8 DN9_9 DN9_10 DN9_11 DN9_12 DN9_13 DN9_14 DN9_15 DN9_16 DN9_17

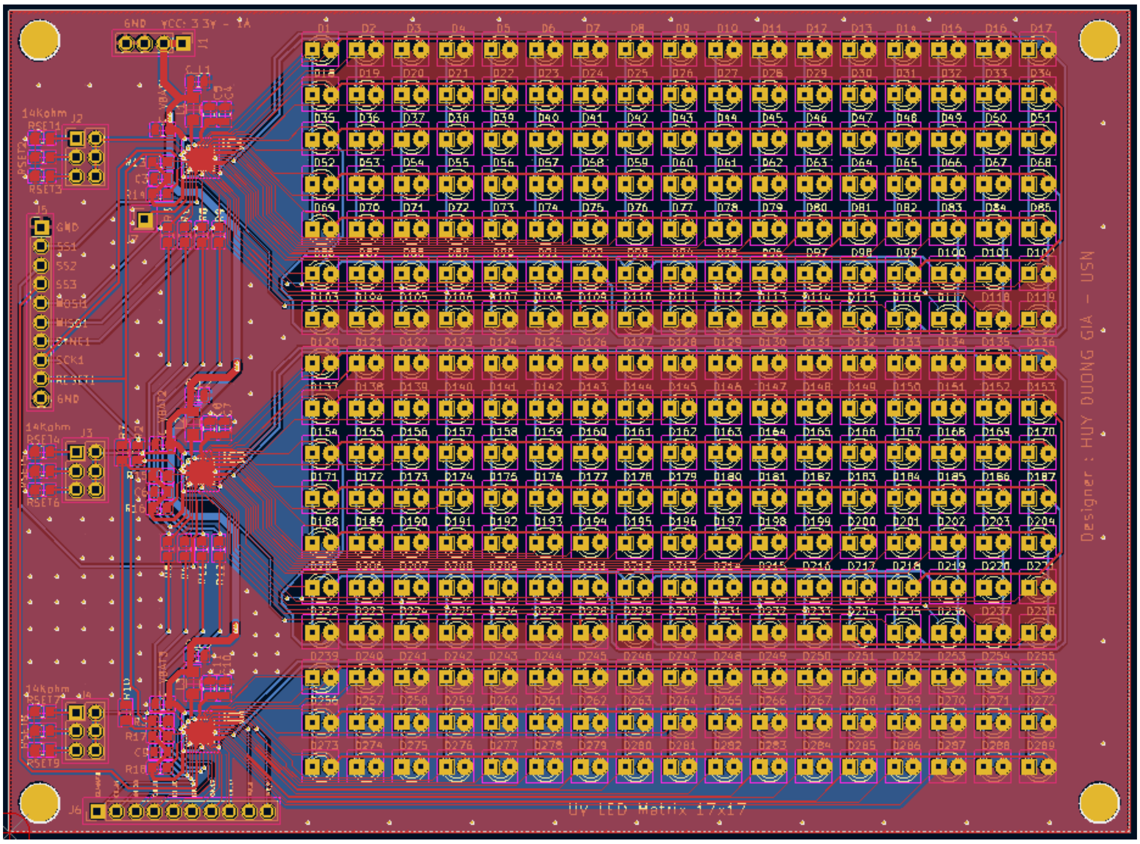

DP10_1 DP10_2 DP10_3 DP10_4 DP10_5 DP10_6 DP10_7 DP10_8 DP10_9 DP10_10 DP10_11 DP10_12 DP10_13 DP10_14 DP10_15 DP10_16 DP10_17
D154 D155 D156 D157 D158 D159 D160 D161 D162 D163 D164 D165 D166 D167 D168 D169 D170
LED
DN10_1 DN10_2 DN10_3 DN10_4 DN10_5 DN10_6 DN10_7 DN10_8 DN10_9 DN10_10 DN10_11 DN10_12 DN10_13 DN10_14 DN10_15 DN10_16 DN10_17
DP11_1 DP11_2 DP11_3 DP11_4 DP11_5 DP11_6 DP11_7 DP11_8 DP11_9 DP11_10 DP11_11 DP11_12 DP11_13 DP11_14 DP11_15 DP11_16 DP11_17
D171 D172 D173 D174 D175 D176 D177 D178 D179 D180 D181 D182 D183 D184 D185 D186 D187
DN11_1 DN11_2 DN11_3 DN11_4 DN11_5 DN11_6 DN11_7 DN11_8 DN11_9 DN11_10 DN11_11 DN11_12 DN11_13 DN11_14 DN11_15 DN11_16 DN11_17
DP12_1 DP12_2 DP12_3 DP12_4 DP12_5 DP12_6 DP12_7 DP12_8 DP12_9 DP12_10 DP12_11 DP12_12 DP12_13 DP12_14 DP12_15 DP12_16 DP12_17
D188 D189 D190 D191 D192 D193 D194 D195 D196 D197 D198 D199 D200 D201 D202 D203 D204
DN12_1 DN12_2 DN12_3 DN12_4 DN12_5 DN12_6 DN12_7 DN12_8 DN12_9 DN12_10 DN12_11 DN12_12 DN12_13 DN12_14 DN12_15 DN12_16 DN12_17
DP13_1 DP13_2 DP13_3 DP13_4 DP13_5 DP13_6 DP13_7 DP13_8 DP13_9 DP13_10 DP13_11 DP13_12 DP13_13 DP13_14 DP13_15 DP13_16 DP13_17
D205 D206 D207 D208 D209 D210 D211 D212 D213 D214 D215 D216 D217 D218 D219 D220 D221
DN13_1 DN13_2 DN13_3 DN13_4 DN13_5 DN13_6 DN13_7 DN13_8 DN13_9 DN13_10 DN13_11 DN13_12 DN13_13 DN13_14 DN13_15 DN13_16 DN13_17
DP14_1 DP14_2 DP14_3 DP14_4 DP14_5 DP14_6 DP14_7 DP14_8 DP14_9 DP14_10 DP14_11 DP14_12 DP14_13 DP14_14 DP14_15 DP14_16 DP14_17
D222 D223 D224 D225 D226 D227 D228 D229 D230 D231 D232 D233 D234 D235 D236 D237 D238
DN14_1 DN14_2 DN14_3 DN14_4 DN14_5 DN14_6 DN14_7 DN14_8 DN14_9 DN14_10 DN14_11 DN14_12 DN14_13 DN14_14 DN14_15 DN14_16 DN14_17
DP15_1 DP15_2 DP15_3 DP15_4 DP15_5 DP15_6 DP15_7 DP15_8 DP15_9 DP15_10 DP15_11 DP15_12 DP15_13 DP15_14 DP15_15 DP15_16 DP15_17
D239 D240 D241 D242 D243 D244 D245 D246 D247 D248 D249 D250 D251 D252 D253 D254 D255
DN15_1 DN15_2 DN15_3 DN15_4 DN15_5 DN15_6 DN15_7 DN15_8 DN15_9 DN15_10 DN15_11 DN15_12 DN15_13 DN15_14 DN15_15 DN15_16 DN15_17
DP16_1 DP16_2 DP16_3 DP16_4 DP16_5 DP16_6 DP16_7 DP16_8 DP16_9 DP16_10 DP16_11 DP16_12 DP16_13 DP16_14 DP16_15 DP16_16 DP16_17
D256 D257 D258 D259 D260 D261 D262 D263 D264 D265 D266 D267 D268 D269 D270 D271 D272
DN16_1 DN16_2 DN16_3 DN16_4 DN16_5 DN16_6 DN16_7 DN16_8 DN16_9 DN16_10 DN16_11 DN16_12 DN16_13 DN16_14 DN16_15 DN16_16 DN16_17
DP17_1 DP17_2 DP17_3 DP17_4 DP17_5 DP17_6 DP17_7 DP17_8 DP17_9 DP17_10 DP17_11 DP17_12 DP17_13 DP17_14 DP17_15 DP17_16 DP17_17
D273 D274 D275 D276 D277 D278 D279 D280 D281 D282 D283 D284 D285 D286 D287 D288 D289
DN17_1 DN17_2 DN17_3 DN17_4 DN17_5 DN17_6 DN17_7 DN17_8 DN17_9 DN17_10 DN17_11 DN17_12 DN17_13 DN17_14 DN17_15 DN17_16 DN17_17


Row1_0 Row1_1 Row1_2 Row1_3 Row1_4 Col1_0 DN1_1 DN2_1 DN3_1 DN4_1 DN5_1
DP1_1 DP2_1 DP3_1 DP4_1 DP5_1 Col1_1 DN1_2 DN2_2 DN3_2 DN4_2 DN5_2
DP1_2 DP2_2 DP3_2 DP4_2 DP5_2 Col1_2 DN1_3 DN2_3 DN3_3 DN4_3 DN5_3
DP1_3 DP2_3 DP3_3 DP4_3 DP5_3 Col1_3 DN1_4 DN2_4 DN3_4 DN4_4 DN5_4
DP1_4 DP2_4 DP3_4 DP4_4 DP5_4 Col1_4 DN1_5 DN2_5 DN3_5 DN4_5 DN5_5
DP1_5 DP2_5 DP3_5 DP4_5 DP5_5 Col1_5 DN1_6 DN2_6 DN3_6 DN4_6 DN5_6
DP1_6 DP2_6 DP3_6 DP4_6 DP5_6 Col1_6 DN1_7 DN2_7 DN3_7 DN4_7 DN5_7
DP1_7 DP2_7 DP3_7 DP4_7 DP5_7 Col1_7 DN1_8 DN2_8 DN3_8 DN4_8 DN5_8
DP1_8 DP2_8 DP3_8 DP4_8 DP5_8 Col1_8 DN1_9 DN2_9 DN3_9 DN4_9 DN5_9
DP1_9 DP2_9 DP3_9 DP4_9 DP5_9 Col1_9 DN1_10 DN2_10 DN3_10 DN4_10 DN5_10
DP1_10 DP2_10 DP3_10 DP4_10 DP5_10 Col1_10 DN1_11 DN2_11 DN3_11 DN4_11 DN5_11
DP1_11 DP2_11 DP3_11 DP4_11 DP5_11 Col1_11 DN1_12 DN2_12 DN3_12 DN4_12 DN5_12
DP1_12 DP2_12 DP3_12 DP4_12 DP5_12 Col1_12 DN1_13 DN2_13 DN3_13 DN4_13 DN5_13
DP1_13 DP2_13 DP3_13 DP4_13 DP5_13 Col1_13 DN1_14 DN2_14 DN3_14 DN4_14 DN5_14
DP1_14 DP2_14 DP3_14 DP4_14 DP5_14 Col1_14 DN1_15 DN2_15 DN3_15 DN4_15 DN5_15
DP1_15 DP2_15 DP3_15 DP4_15 DP5_15 Col1_15 DN1_16 DN2_16 DN3_16 DN4_16 DN5_16
DP1_16 DP2_16 DP3_16 DP4_16 DP5_16 Col1_16 DN1_17 DN2_17 DN3_17 DN4_17 DN5_17
DP1_17 DP2_17 DP3_17 DP4_17 DP5_17
DP6_1 DP6_8 DP7_1 DP7_8 DP6_15 Col1_17 DN6_1 DN6_8 DN7_1 DN7_8 DN6_15
DP6_2 DP6_9 DP7_2 DP7_9 DP6_16 Col1_18 DN6_2 DN6_9 DN7_2 DN7_9 DN6_16
DP6_3 DP6_10 DP7_3 DP7_10 DP6_17 Col1_19 DN6_3 DN6_10 DN7_3 DN7_10 DN6_17
DP6_4 DP6_11 DP7_4 DP7_11 DP7_15 Col1_20 DN6_4 DN6_11 DN7_4 DN7_11 DN7_15
DP6_5 DP6_12 DP7_5 DP7_12 DP7_16 Col1_21 DN6_5 DN6_12 DN7_5 DN7_12 DN7_16
DP6_6 DP6_13 DP7_6 DP7_13 DP7_17 Col1_22 DN6_6 DN6_13 DN7_6 DN7_13 DN7_17
DP6_7 DP6_14 DP7_7 DP7_14 Col1_23 DN6_7 DN6_14 DN7_7 DN7_14

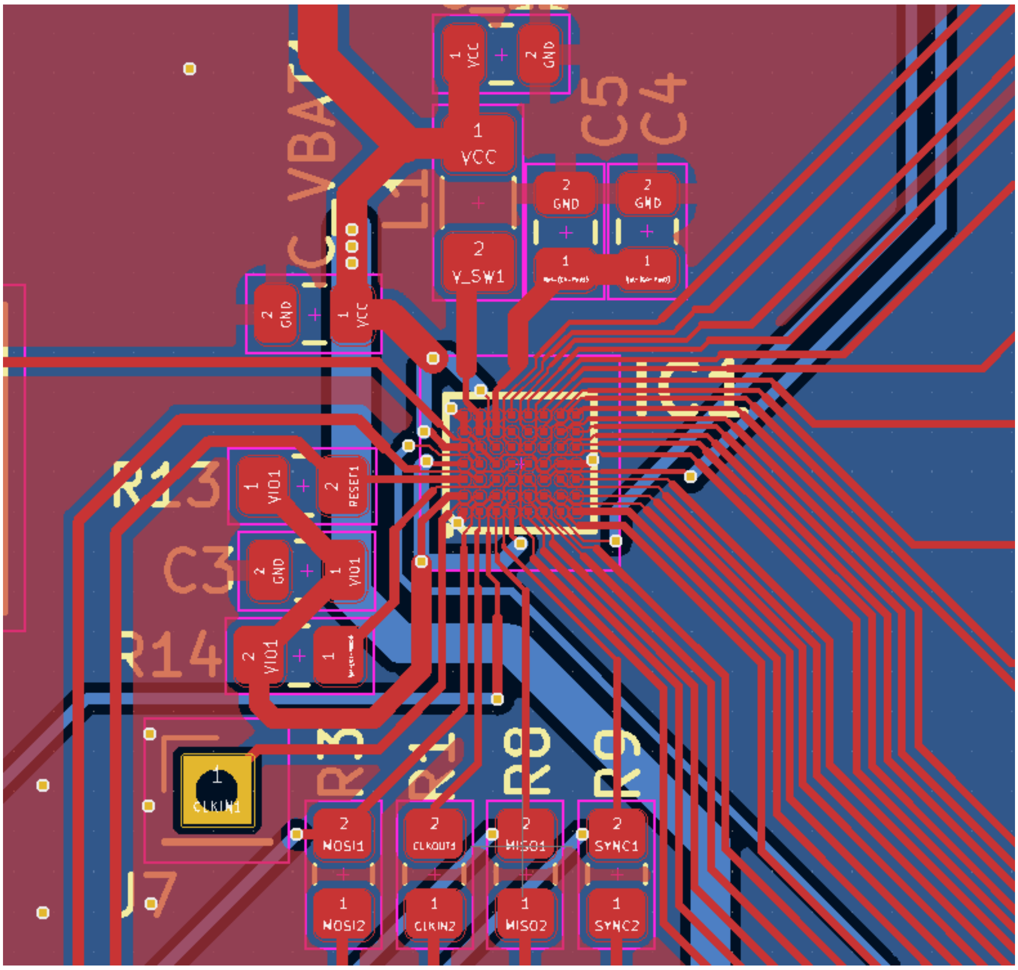

Row2_0 Row2_1 Row2_2 Row2_3 Row2_4 Col2_0 DN8_1 DN9_1 DN10_1 DN11_1 DN12_1
DP8_1 DP9_1 DP10_1 DP11_1 DP12_1 Col2_1 DN8_2 DN9_2 DN10_2 DN11_2 DN12_2
DP8_2 DP9_2 DP10_2 DP11_2 DP12_2 Col2_2 DN8_3 DN9_3 DN10_3 DN11_3 DN12_3
DP8_3 DP9_3 DP10_3 DP11_3 DP12_3 Col2_3 DN8_4 DN9_4 DN10_4 DN11_4 DN12_4
DP8_4 DP9_4 DP10_4 DP11_4 DP12_4 Col2_4 DN8_5 DN9_5 DN10_5 DN11_5 DN12_5
DP8_5 DP9_5 DP10_5 DP11_5 DP12_5 Col2_5 DN8_6 DN9_6 DN10_6 DN11_6 DN12_6
DP8_6 DP9_6 DP10_6 DP11_6 DP12_6 Col2_6 DN8_7 DN9_7 DN10_7 DN11_7 DN12_7
DP8_7 DP9_7 DP10_7 DP11_7 DP12_7 Col2_7 DN8_8 DN9_8 DN10_8 DN11_8 DN12_8
DP8_8 DP9_8 DP10_8 DP11_8 DP12_8 Col2_8 DN8_9 DN9_9 DN10_9 DN11_9 DN12_9
DP8_9 DP9_9 DP10_9 DP11_9 DP12_9 Col2_9 DN8_10 DN9_10 DN10_10 DN11_10 DN12_10
DP8_10 DP9_10 DP10_10 DP11_10 DP12_10 Col2_10 DN8_11 DN9_11 DN10_11 DN11_11 DN12_11
DP8_11 DP9_11 DP10_11 DP11_11 DP12_11 Col2_11 DN8_12 DN9_12 DN10_12 DN11_12 DN12_12
DP8_12 DP9_12 DP10_12 DP11_12 DP12_12 Col2_12 DN8_13 DN9_13 DN10_13 DN11_13 DN12_13
DP8_13 DP9_13 DP10_13 DP11_13 DP12_13 Col2_13 DN8_14 DN9_14 DN10_14 DN11_14 DN12_14
DP8_14 DP9_14 DP10_14 DP11_14 DP12_14 Col2_14 DN8_15 DN9_15 DN10_15 DN11_15 DN12_15
DP8_15 DP9_15 DP10_15 DP11_15 DP12_15 Col2_15 DN8_16 DN9_16 DN10_16 DN11_16 DN12_16
DP8_16 DP9_16 DP10_16 DP11_16 DP12_16 Col2_16 DN8_17 DN9_17 DN10_17 DN11_17 DN12_17
DP8_17 DP9_17 DP10_17 DP11_17 DP12_17
DP13_1 DP13_8 DP14_1 DP14_8 DP13_15 Col2_17 DN13_1 DN13_8 DN14_1 DN14_8 DN13_15
DP13_2 DP13_9 DP14_2 DP14_9 DP13_16 Col2_18 DN13_2 DN13_9 DN14_2 DN14_9 DN13_16
DP13_3 DP13_10 DP14_3 DP14_10 DP13_17 Col2_19 DN13_3 DN13_10 DN14_3 DN14_10 DN13_17
DP13_4 DP13_11 DP14_4 DP14_11 DP14_15 Col2_20 DN13_4 DN13_11 DN14_4 DN14_11 DN14_15
DP13_5 DP13_12 DP14_5 DP14_12 DP14_16 Col2_21 DN13_5 DN13_12 DN14_5 DN14_12 DN14_16
DP13_6 DP13_13 DP14_6 DP14_13 DP14_17 Col2_22 DN13_6 DN13_13 DN14_6 DN14_13 DN14_17
DP13_7 DP13_14 DP14_7 DP14_14 Col2_23 DN13_7 DN13_14 DN14_7 DN14_14


Row3_0 Row3_1 Row3_2 Col3_0 DN15_1 DN16_1 DN17_1
DP15_1 DP16_1 DP17_1 Col3_1 DN15_2 DN16_2 DN17_2
DP15_2 DP16_2 DP17_2 Col3_2 DN15_3 DN16_3 DN17_3
DP15_3 DP16_3 DP17_3 Col3_3 DN15_4 DN16_4 DN17_4
DP15_4 DP16_4 DP17_4 Col3_4 DN15_5 DN16_5 DN17_5
DP15_5 DP16_5 DP17_5 Col3_5 DN15_6 DN16_6 DN17_6
DP15_6 DP16_6 DP17_6 Col3_6 DN15_7 DN16_7 DN17_7
DP15_7 DP16_7 DP17_7 Col3_7 DN15_8 DN16_8 DN17_8
DP15_8 DP16_8 DP17_8 Col3_8 DN15_9 DN16_9 DN17_9
DP15_9 DP16_9 DP17_9 Col3_9 DN15_10 DN16_10 DN17_10
DP15_10 DP16_10 DP17_10 Col3_10 DN15_11 DN16_11 DN17_11
DP15_11 DP16_11 DP17_11 Col3_11 DN15_12 DN16_12 DN17_12
DP15_12 DP16_12 DP17_12 Col3_12 DN15_13 DN16_13 DN17_13
DP15_13 DP16_13 DP17_13 Col3_13 DN15_14 DN16_14 DN17_14
DP15_14 DP16_14 DP17_14 Col3_14 DN15_15 DN16_15 DN17_15
DP15_15 DP16_15 DP17_15 Col3_15 DN15_16 DN16_16 DN17_16
DP15_16 DP16_16 DP17_16 Col3_16 DN15_17 DN16_17 DN17_17
DP15_17 DP16_17 DP17_17

PCB Design



GND VCC 3.3V - 1A
Uv LED Matrix 17x17
Designer : HUY DUONG GIA - USN



VBAT
C5
C4
L1
IC1
R13
C3
R14
J7
CLKIN1
R3
R8
R9
V_SW1
RESET1
MOSI1
MOSI2
CLKIN2
MISO1
MISO2
SYNC1
SYNC2

Appendix D: Software program

The software is programmed by Python version 3.10.2

```
import tkinter as tk
import time as t
from PIL import ImageTk, Image, features
import cv2
import serial
import time
from pixelink import PixeLINK, PxLapi
import numpy as np
import os
import math


global folderLocation
folderLocation = "C://Users//huydu//OneDrive - USN//master thesis//Measurment//IR blood
cell 1//"
if not os.path.exists(folderLocation):
    os.makedirs(folderLocation)


global camera_gain
camera_gain = 0
global camera_pixel_format
camera_pixel_format = 1
global camera_roi
#camera_roi = (1648, 680, 800, 800)
camera_roi = (1920, 952, 256, 256)
# camera_roi = (0, 0, 4096, 2160)

global exposure_time
exposure_time = \
[[5000,5000,5000,5000,5000,5000,5000,5000,5000,5000,5000,5000,5000,5000,5000,5000,5000],
[5000,4000,4000,4000,4000,4000,4000,4000,4000,4000,4000,4000,4000,4000,4000,4000,5000],
[5000,4000,3000,3000,3000,3000,3000,3000,3000,3000,3000,3000,3000,3000,3000,4000,5000],
[5000,4000,3000,2000,2000,2000,2000,2000,2000,2000,2000,2000,2000,2000,3000,4000,5000],
[5000,4000,3000,2000,900,900,900,900,900,900,900,900,900,2000,3000,4000,5000],
[5000,4000,3000,2000,900,600,600,600,600,600,600,600,900,2000,3000,4000,5000],
[5000,4000,3000,2000,900,600,100,100,100,100,100,600,900,2000,3000,4000,5000],
[5000,4000,3000,2000,900,600,100,22,22,22,100,600,900,2000,3000,4000,5000],
[5000,4000,3000,2000,900,600,100,17,22,22,100,600,600,2000,3000,4000,5000],
[5000,4000,3000,2000,900,600,100,22,20,22,100,600,900,2000,3000,4000,5000],
[5000,4000,3000,2000,900,600,100,100,100,100,100,600,900,2000,3000,4000,5000],
[5000,4000,3000,2000,900,600,600,600,600,600,600,600,900,2000,3000,4000,5000],
[5000,4000,3000,2000,900,900,900,900,900,900,900,900,900,2000,3000,4000,5000],
[5000,4000,3000,2000,2000,2000,2000,2000,2000,2000,2000,2000,2000,2000,3000,4000,5000],
[5000,4000,3000,3000,3000,3000,3000,3000,3000,3000,3000,3000,3000,3000,3000,4000,5000],
[5000,4000,4000,4000,4000,4000,4000,4000,4000,4000,4000,4000,4000,4000,4000,4000,5000],
[5000,5000,5000,5000,5000,5000,5000,5000,5000,5000,5000,5000,5000,5000,5000,5000,5000]]

LED_number_in_order = \
[145,162,161,144,127,128,129,146,163,180,179,178,177,160,143,126,109,110,111,112,113,130
,147,164,181,198,197,196,195,194,193,176,159,142,125,108,91,92,93,94,95,96,97,114,131,14
8,165,182,199,216,215,214,213,212,211,210,209,192,175,158,141,124,107,90,73,74,75,76,77,
78,79,80,81,98,115,132,149,166,183,200,217,234,233,232,231,230,229,228,227,226,225,208,1
91,174,157,140,123,106,89,72,55,56,57,58,59,60,61,62,63,64,65,82,99,116,133,150,167,184,
201,218,235,252,251,250,249,248,247,246,245,244,243,242,241,224,207,190,173,156,139,122,
105,88,71,54,37,38,39,40,41,42,43,44,45,46,47,48,49,66,83,100,117,134,151,168,185,202,21
9,236,253,270,269,268,267,266,265,264,263,262,261,260,259,258,257,240,223,206,189,172,15
5,138,121,104,87,70,53,36,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,50,67,84,101,118,
135,152,169,186,203,220,237,254,271,288,287,286,285,284,283,282,281,280,279,278,277,276,
275,274,273,256,239,222,205,188,171,154,137,120,103,86,69,52,35,18,1,2,3,4,5,6,7,8,9,10,
11,12,13,14,15,16,17,34,51,68,85,102,119,136,153,170,187,204,221,238,255,272,289]

global circle_radius
circle_radius = 215

def take_image_func():
    global exposure_time
    global camera_gain
    global camera_pixel_format
    global camera_roi
    cam = PixeLINK()
    cam.roi = camera_roi
```

```
    cam.streaming = False
    cam.pixel_format = camera_pixel_format
    cam.gain = 0
    cam.shutter = 3
    cam.streaming = True
    waiting_for_heating_camera()
    image_name = image_name_entry.get()
    take_image(cam, image_name)

def take_image(cam, image_name):
    frame = cam.grab()
    frame = frame*16
    im = Image.fromarray(frame)
    global folderLocation
    im.save(folderLocation + image_name + ".tiff")

def turn_on_func():
    print(ser.name)          # check which port was really used
    check_serial_connection(ser)
    led_position_X = led_position_X_entry.get();
    led_position_Y = led_position_Y_entry.get();
    turn_on_Led(int(led_position_X), int(led_position_Y))

# this function heat camera first and dont turn off camera while taking measuremnet
def take_measurement_func():
    # check the serial is opened?
    check_serial_connection(ser)
    # time.sleep(60)
    global exposure_time
    global camera_gain
    global camera_pixel_format
    global camera_roi
    cam = PixeLINK()
    cam.roi = camera_roi
    cam.streaming = False
    cam.pixel_format = camera_pixel_format
    cam.gain = camera_gain
    cam.streaming = True
    waiting_for_heating_camera()

    for order_number in range(289):
        image_name = str(LED_number_in_order[order_number])
        row = math.ceil((LED_number_in_order[order_number]/17))
        col = LED_number_in_order[order_number] - (row-1)*17

        turn_on_Led(row, col)
        waiting_for_settle_light()

        cam.streaming = False
        cam.shutter = exposure_time[row-1][col-1]/1000
        cam.streaming = True

        take_image(cam, image_name)

    cam.close()

def waiting_for_settle_light():
    time.sleep(0.5)

def wait_for_exposure_time(exposure_time):
    time.sleep(exposure_time)

def waiting_for_heating_camera():
    time.sleep(15*60)

def waiting_for_cooling_cemara():
    time.sleep(30)

def turn_on_Led(row, col):
    if row < 10:
        row_str = '0' + str(row)
    else: row_str = str(row)
    if col < 10:
        col_str = '0' + str(col)
    else: col_str = str(col)
    send_data = 's' + 'a' + row_str + col_str + 'e'
    bytes_send = bytes(send_data,'UTF-8')
    ser.write(bytes_send)
```

```
def turn_off_led_func():
    send_data = 's' + 'c' + 'e'
    bytes_send = bytes(send_data,'UTF-8')
    ser.write(bytes_send)

def set_PWM_func():
    check_serial_connection(ser)
    PWM_value = PWM_value_entry.get()
    set_PWM(int(PWM_value))

def set_PWM(PWM_value):
    if PWM_value < 10: PWM_value_str = "00" + str(PWM_value)
    elif PWM_value < 100 and PWM_value > 9: PWM_value_str = "0" + str(PWM_value)
    else: PWM_value_str = str(PWM_value)
    send_data = 's' + 'b' + PWM_value_str + 'e'
    bytes_send = bytes(send_data,'UTF-8')
    ser.write(bytes_send)

def calibration_func():
    check_serial_connection(ser)
    turn_on_Led(9,9) # turn on center LED
    # set_PWM(3)
    global cam
    cam = PixeLINK()
    camera_roi = (0, 0, 4096, 2160)
    cam.shutter = 0.03
    video_stream()

# function for video streaming
def video_stream():
    global cam
    im = cam.grab()

    image_size_X = 4096
    image_size_Y = 2160
    a = int(image_size_X*0.2)
    b = int(image_size_Y*0.2)
    resized_im = cv2.resize(im, (a,b))

    ret, threshold_im = cv2.threshold(resized_im, 9, 255, cv2.THRESH_BINARY)
    # print(type(threshold_im[100][100]))

    threshold_im = cv2.cvtColor(threshold_im, cv2.COLOR_GRAY2RGB)

    circle_center = ((int(a/2), int(b/2)))
    # circle_radius = int(circle_radius_entry.get())
    global circle_radius
    circle_thickness = 2
    circle_color = (255, 0, 0)
    im = cv2.circle(threshold_im, circle_center, circle_radius, circle_color,
circle_thickness)

    im2 = Image.fromarray(im)
    imgTk = ImageTk.PhotoImage(image=im2)
    image_label.imgtk = imgTk
    image_label.configure(image=imgTk)
    image_label.after(1, video_stream)

def check_serial_connection(ser):
    try:
        ser.isOpen()
        print("serial is opening")
        return True
    except:
        print("serial port is disconnected")
        return False

# conect serial port
ser = None
try:
    ser = serial.Serial("COM3", 9600)
except:
    print("can not connect serial")

check_serial_connection(ser)

window = tk.Tk()
```

```
window.wm_title("Fourier ptychography")
window.geometry("1100x800")

take_image_btn = tk.Button(text="take image", width=10, height= 3, command =
take_image_func)
take_image_btn.place(x= 100, y = 10)

image_name_label = tk.Label(text = "image name")
image_name_label.place(x=200, y= 10)

image_name_entry = tk.Entry()
image_name_entry.place(x=200, y= 40)

send_uart_btn = tk.Button(text="Turn on", width=10, height= 3, command = turn_on_func)
send_uart_btn.place(x= 100, y = 100)

led_position_label = tk.Label(text = "LED position")
led_position_label.place(x= 200, y = 80)

led_position_X_entry = tk.Entry()
led_position_X_entry.place(x= 200, y = 100)

led_position_X_label = tk.Label(text = "Row number")
led_position_X_label.place(x= 250, y = 100)

led_position_Y_entry = tk.Entry()
led_position_Y_entry.place(x= 200, y = 130)

led_position_Y_label = tk.Label(text = "Column number")
led_position_Y_label.place(x= 250, y = 130)

take_measurement_btn = tk.Button(text="take measurement", width=20, height= 3, command =
take_measurement_func)
take_measurement_btn.place(x= 400, y = 10)

set_PWM_btn = tk.Button(text= "set PWM", width=10, height= 3, command = set_PWM_func)
set_PWM_btn.place(x = 400, y=100)

PWM_value_label = tk.Label(text = "PWM value: from 0 to 255")
PWM_value_label.place(x = 500, y = 100)

PWM_value_entry = tk.Entry()
PWM_value_entry.place(x = 500, y = 130)

turn_off_led_btn = tk.Button(text= "Turn off LED", width=10, height= 3, command =
turn_off_led_func)
turn_off_led_btn.place(x = 700, y=100)

image_frame = tk.Frame(window)
image_frame.place(x=10,y=180)
# Create a label in the frame
image_label = tk.Label(image_frame)
image_label.grid()

calibration_btn = tk.Button(text= "calibrate Center LED", width=20, height= 3, command =
calibration_func)
calibration_btn.place(x=850, y=10)

circle_radius_label = tk.Label(text="circle radius")
circle_radius_label.place(x = 850, y = 70)

circle_radius_entry = tk.Entry()
circle_radius_entry.place(x = 850, y = 90)

exposure_time_for_calibration_label = tk.Label(text="exposure time for calibration in
s")
exposure_time_for_calibration_label.place(x = 850, y = 110)

exposure_time_for_calibration_entry = tk.Entry()
exposure_time_for_calibration_entry.place(x = 850, y = 130)

PWM_for_calibration_label = tk.Label(text="PWM for calibration")
PWM_for_calibration_label.place(x = 850, y = 150)

PWM_for_calibration_entry = tk.Entry()
PWM_for_calibration_entry.place(x = 850, y = 170)
```

```
def set_circle_parameters():
    global circle_radius
    circle_radius = int(circle_radius_entry.get())
circle_radius_btn = tk.Button(text= "set", width=7, height= 1, command =
set_circle_parameters)
circle_radius_btn.place(x = 1000, y = 80)

def set_exposure_time_for_calibration_func():
    global cam
    cam.shutter = float(exposure_time_for_calibration_entry.get())
exposure_time_for_calibration_btn = tk.Button(text= "set", width=7, height= 1, command =
set_exposure_time_for_calibration_func)
exposure_time_for_calibration_btn.place(x = 1000, y = 120)

window.mainloop()
```

Appendix E: Recovery process programming

The recovery process is programmed by Matlab version R2020a:

```
clear all; clc; close all;

%% init parameter
waveLength = 0.89e-6; % IR light
k0 = 2*pi/waveLength;
NA = 0.1;
cutOffFrequency = NA*k0;
spSize = 3.45e-6/4; % 3.45um
pixelSize = spSize/8;
LEDgap = 6e-3; % 6mm
% guest value
LEDheight = 70e-3;

max_arraySize = 17;
maxLEDnumber = max_arraySize^2;
seg = gseq(max_arraySize);
arraySize = 17;
LEDnumber = arraySize^2;

xInputImageSize = 256;
yInputImageSize = 256;

xRecoveryImageSize = 1024;
yRecoveryImageSize = 1024;

%% create the wave vectors for the LED illumination
xlocation = zeros(1,max_arraySize^2);
ylocation = zeros(1,max_arraySize^2);

for i=1:max_arraySize
    xlocation(1,1+max_arraySize*(i-1):max_arraySize+max_arraySize*(i-1)) = ((-
(max_arraySize-1)/2):1:(max_arraySize-1)/2)*LEDgap;
    ylocation(1,1+max_arraySize*(i-1):max_arraySize+max_arraySize*(i-1)) = -
((max_arraySize-1)/2-i+1)*LEDgap;
end

kx_relative = (-xlocation./sqrt(LEDheight^2+xlocation.^2+ylocation.^2));
ky_relative = (-ylocation./sqrt(LEDheight^2+xlocation.^2+ylocation.^2));

dkx = 2*pi/(pixelSize*xRecoveryImageSize);
dky = 2*pi/(pixelSize*yRecoveryImageSize);

kx = k0*kx_relative;
ky = k0*ky_relative;

%% generate coherent transfer function
kmax = pi/spSize;
[kxm kym] = meshgrid(-kmax:dkx:kmax-dkx, -kmax:dkx:kmax-dkx);
CTF = ((kxm.^2+kym.^2)<cutOffFrequency^2);
%CTF = CTF_(1:xInputImageSize, 1:yInputImageSize);
figure; imshow(CTF,[], 'XData', [-kmax, kmax], 'YData', [-kmax, kmax]);
title('CTF'); xlabel('rad/s');ylabel('rad/s')
axis('on', 'image');

%% calculate overlap
kx1 = k0*sin(atan(LEDgap/LEDheight));
r = cutOffFrequency;
AB = kx1;
overlap_angle = 2*acos(AB/2/r);
circle_area = pi*r^2;
overlap_area = circle_area*overlap_angle/(2*pi) - 0.5*r*r*sin(overlap_angle);
overlap_area = 2*overlap_area;
overlap_percent = overlap_area/circle_area

%% read real images
measurementName = 'GaAs 5';
imagesFolderLocation = 'C:/Users/huydu/OneDrive - USN/master thesis/Measurment/';
imagesFolderLocation = strcat(imagesFolderLocation, measurementName, '/');

inputImage = zeros(xInputImageSize, yInputImageSize, maxLEDnumber);
```

```matlab
for i=1:LEDnumber
    i2 = seg(i);
    imageNumberChar = int2str(i2);
    imageDirection = append(imagesFolderLocation, imageNumberChar);
    imageDirection = append(imageDirection, '.tiff');
    realImage = double(imread(imageDirection));
    [xRealImageSize, yRealImageSize] = size(realImage);
    if xRealImageSize~=xInputImageSize
        inputImage(:,:,i2) = realImage(xRealImageSize/2-
xInputImageSize/2:xRealImageSize/2+xInputImageSize/2-1, yRealImageSize/2-
yInputImageSize/2:yRealImageSize/2+yInputImageSize/2-1);
        %inputImage(:,:,i) = sqrt(inputImageIntensity);
    else
        inputImage(:,:,i2) = realImage;
    end
end

%% scale image because each image corresponds to different
exposure time
BrightImageNumbers = [127 128 129 144 145 146 161 162 163];
badImage = [4 14 (14*17+9)];
% each measurement has different exposure time table
exposure_time = ...
[[5000 5000 5000 5000 5000 5000 5000 5000 5000 5000 5000 5000 5000 5000 5000 5000 5000];
[5000 4000 4000 4000 4000 4000 4000 4000 4000 4000 4000 4000 4000 4000 4000 4000 5000] ;
[5000 4000 3000 3000 3000 3000 3000 3000 3000 3000 3000 3000 3000 3000 3000 4000 5000] ;
[5000 4000 3000 2000 2000 2000 2000 2000 2000 2000 2000 2000 2000 2000 3000 4000 5000] ;
[5000 4000 3000 2000 900 900 900 900 900 900 900 900 900 2000 3000 4000 5000] ;
[5000 4000 3000 2000 900 600 600 600 600 600 600 600 900 2000 3000 4000 5000] ;
[5000 4000 3000 2000 900 600 100 100 100 100 100 600 900 2000 3000 4000 5000] ;
[5000 4000 3000 2000 900 600 100 22 22 22 100 600 900 2000 3000 4000 5000] ;
[5000 4000 3000 2000 900 600 100 17 22 22 100 600 600 2000 3000 4000 5000] ;
[5000 4000 3000 2000 900 600 100 22 20 22 100 600 900 2000 3000 4000 5000] ;
[5000 4000 3000 2000 900 600 100 100 100 100 100 600 900 2000 3000 4000 5000] ;
[5000 4000 3000 2000 900 600 600 600 600 600 600 600 900 2000 3000 4000 5000];
[5000 4000 3000 2000 900 900 900 900 900 900 900 900 900 2000 3000 4000 5000] ;
[5000 4000 3000 2000 2000 2000 2000 2000 2000 2000 2000 2000 2000 2000 3000 4000 5000];
[5000 4000 3000 3000 3000 3000 3000 3000 3000 3000 3000 3000 3000 3000 3000 4000 5000] ;
[5000 4000 4000 4000 4000 4000 4000 4000 4000 4000 4000 4000 4000 4000 4000 4000 5000] ;
[5000 5000 5000 5000 5000 5000 5000 5000 5000 5000 5000 5000 5000 5000 5000 5000 5000]];

inputImageScale = zeros(xInputImageSize, yInputImageSize, maxLEDnumber);
for i = 1:maxLEDnumber
    imageNumber = seg(i);
    row = ceil(imageNumber/17);
    col = imageNumber - (row-1)*17;
    inputImageScale(:,:,imageNumber) = inputImage(:,:,imageNumber)/(exposure_time(row,
col)/exposure_time(ceil(arraySize/2), ceil(arraySize/2)));
end

%% Noise reduction using thresholding method
FiltedInputAmplitude = zeros(xInputImageSize, yInputImageSize, maxLEDnumber);
% meanvalue calculation
area_1_XL = 100; area_1_XH = 135; area_1_YL = 65; area_1_YH = 80;
area_2_XL = 85; area_2_XH = 130; area_2_YL = 235; area_2_YH = 255;

for i = 1:LEDnumber
    imageNumber = seg(i);
    if all(BrightImageNumbers~=imageNumber)
        % calculate everageNumber
        Area1IntensitySum = sum(sum(inputImageScale(area_1_YL:area_1_YH,
area_1_XL:area_1_XH, imageNumber)));
        Area2IntensitySum = sum(sum(inputImageScale(area_2_YL:area_2_YH,
area_2_XL:area_2_XH, imageNumber)));
        Area1 = (area_1_XH-area_1_XL+1)*(area_1_YH-area_1_YL+1);
        Area2 = (area_2_XH-area_2_XL+1)*(area_2_YH-area_2_YL+1);
        averageValue = (Area1IntensitySum/Area1+Area2IntensitySum/Area2)/2;


        % reduce noise
        FiltedImage = inputImageScale(:,:,imageNumber) - averageValue;
        maskImage = FiltedImage > 0;
        FiltedImage = FiltedImage.*maskImage;
    else
        FiltedImage = inputImageScale(:,:,imageNumber);
    end
    FiltedInputAmplitude(:,:,imageNumber) = sqrt(FiltedImage);
```

```
end
```

%% Monitoring the convergence index versus LED-Sample distance.

```
% input low resolution image
inputAmplitude = FiltedInputAmplitude;
LED_sample_distance = 65:0.5:80;
LED_sample_distance = LED_sample_distance*10^-3;
length_ = length(LED_sample_distance);
error_distance_L_S = zeros(1, length_);
converIndex = zeros(1, length_);

for i = 1:length_
    LED_sample_distance(i)
    kx_relative_tunning = (-
xlocation./sqrt(LED_sample_distance(i)^2+xlocation.^2+ylocation.^2));
    ky_relative_tunning = (-
ylocation./sqrt(LED_sample_distance(i)^2+xlocation.^2+ylocation.^2));

    kx_tunning = k0*kx_relative_tunning;
    ky_tunning = k0*ky_relative_tunning;

    objectRecover = imresize(inputAmplitude(:,:,145), [yRecoveryImageSize
xRecoveryImageSize]);
    objectRecoverFT_after_tunning = fftshift(fft2(objectRecover));
    pupil_after_tunning = double(CTF);
    loop = 6;
    for tt=1:loop
        tt
        for i3=1:9^2
            i2 = seg(i3);
            kxc = round((xRecoveryImageSize)/2+1-kx_tunning(1,i2)/dkx);
            kyc = round((yRecoveryImageSize)/2+1-ky_tunning(1,i2)/dky);
            kyl = round(kyc-(yInputImageSize)/2);kyh=round(kyc+(yInputImageSize)/2-1);
            kxl = round(kxc-(xInputImageSize)/2);kxh=round(kxc+(xInputImageSize)/2-1);
            unUpdate_lowResFT =
objectRecoverFT_after_tunning(kyl:kyh,kxl:kxh).*CTF.*pupil_after_tunning;
            im_lowRes = ifft2(ifftshift(unUpdate_lowResFT));

            amplitude_lowRes = abs(im_lowRes); % amplitude
            amplitude_lowRes = amplitude_lowRes/max(max(amplitude_lowRes)); % normalize

            measureAmplitude =
(yRecoveryImageSize/yInputImageSize)^2*inputAmplitude(:,:,i2);
            measureAmplitudeNorm = measureAmplitude/max(max(measureAmplitude)); %
normalize

            converIndex(i) = converIndex(i) +
sum(sum(abs(measureAmplitudeNorm)))/sum(sum(abs(amplitude_lowRes-
measureAmplitudeNorm)));
            % update
            updated_im_lowRes = measureAmplitude.*exp(1i.*angle(im_lowRes));
            updated_lowResFT = fftshift(fft2(updated_im_lowRes));
            objectRecoverFT_after_tunning(kyl:kyh,kxl:kxh) =
objectRecoverFT_after_tunning(kyl:kyh,kxl:kxh) +
conj(pupil_after_tunning)./(max(max(abs(pupil_after_tunning).^2))).*(updated_lowResFT-
unUpdate_lowResFT);
            pupil_after_tunning = pupil_after_tunning +
conj(objectRecoverFT_after_tunning(kyl:kyh,kxl:kxh))./(max(max(abs(objectRecoverFT_after
_tunning(kyl:kyh,kxl:kxh)).^2))).*(updated_lowResFT-unUpdate_lowResFT);

        end
    end
    figure; imshow(ifft2(ifftshift(objectRecoverFT_after_tunning)),[]);
end
maxConv = max(converIndex(:));
LEDheight_all = LED_sample_distance(find(converIndex == maxConv));
figure; plot(LED_sample_distance, converIndex);
xlabel('LED-Sample distance (m)');ylabel('Convergence index')
if length(LEDheight_all)~=1
    LEDheight = LEDheight_all(1);
else
    LEDheight = LEDheight_all;
end
% recalculate the wave vector with new LED_height
kx_relative = (-xlocation./sqrt(LEDheight^2+xlocation.^2+ylocation.^2));
ky_relative = (-ylocation./sqrt(LEDheight^2+xlocation.^2+ylocation.^2));
kx = k0*kx_relative;
```

```
ky = k0*ky_relative;


%% recovery
inputAmplitude = FiltedInputAmplitude; % avoid losing the intial data

% init high-resolution image
objectRecover = imresize(inputAmplitude(:,:,145), [yRecoveryImageSize
xRecoveryImageSize]);
objectRecoverFT = fftshift(fft2(objectRecover));

% init pupil function
pupil = double(CTF);

%%% this array size for control the matrix size while recover
rec_arraySize = 17;

%% intensity correction, bright-field images only. It helps!
intensity_factor = ones(1, maxLEDnumber);
loop = 5;
for tt=1:loop
    for i3=1:length(BrightImageNumbers)
        i2 = BrightImageNumbers(i3);
        kxc = round((xRecoveryImageSize)/2+1-kx(1,i2)/dkx);
        kyc = round((yRecoveryImageSize)/2+1-ky(1,i2)/dky);
        kyl = round(kyc-(yInputImageSize)/2);kyh=round(kyc+(yInputImageSize)/2-1);
        kxl = round(kxc-(xInputImageSize)/2);kxh=round(kxc+(xInputImageSize)/2-1);
        unUpdate_lowResFT = objectRecoverFT(kyl:kyh,kxl:kxh).*CTF.*pupil;
        im_lowRes = ifft2(ifftshift(unUpdate_lowResFT));

        % calculate intensity factor
        measured_lowRes = (yRecoveryImageSize/yInputImageSize)^2*inputAmplitude(:,:,i2);
        intensity_factor(i2) =
sum(sum(abs(im_lowRes).^2))/sum(sum(abs(measured_lowRes).^2));
        % overwrite the intesity measurement
        inputAmplitude(:,:,i2) =
sqrt(intensity_factor(i2)*(abs(inputAmplitude(:,:,i2)).^2));

        % update
        updated_im_lowRes =
(yRecoveryImageSize/yInputImageSize)^2*inputAmplitude(:,:,i2).*exp(1i.*angle(im_lowRes))
;
        updated_lowResFT = fftshift(fft2(updated_im_lowRes));
        objectRecoverFT(kyl:kyh,kxl:kxh) = objectRecoverFT(kyl:kyh,kxl:kxh) +
conj(pupil)./(max(max(abs(pupil).^2))).*(updated_lowResFT-unUpdate_lowResFT);
        pupil = pupil +
conj(objectRecoverFT(kyl:kyh,kxl:kxh))./(max(max(abs(objectRecoverFT(kyl:kyh,kxl:kxh)).^
2))).*(updated_lowResFT-unUpdate_lowResFT);
    end
end

%% improved threshold noise reduction. for dark-field images
only. It helps!
loop = 1;
for tt=1:loop
    tt
    for i3=1:rec_arraySize^2
        i2 = seq(i3);
        if all(BrightImageNumbers~=i2) && all(badImage ~= i2)
            kxc = round((xRecoveryImageSize)/2+1-kx(1,i2)/dkx);
            kyc = round((yRecoveryImageSize)/2+1-ky(1,i2)/dky);
            kyl = round(kyc-(yInputImageSize)/2);kyh=round(kyc+(yInputImageSize)/2-1);
            kxl = round(kxc-(xInputImageSize)/2);kxh=round(kxc+(xInputImageSize)/2-1);
            unUpdate_lowResFT = objectRecoverFT(kyl:kyh,kxl:kxh).*CTF.*pupil;
            im_lowRes = ifft2(ifftshift(unUpdate_lowResFT));
            intensity_lowRes = (yInputImageSize/yRecoveryImageSize)^4*abs(im_lowRes).^2;

            measureIntensity = inputAmplitude(:,:,i2).^2;

            threshold = abs(mean2(measureIntensity) - mean2(intensity_lowRes));
            updatedItensity = measureIntensity - threshold;
            maskImage = updatedItensity > 0;

            inputAmplitude(:,:,i2) = sqrt(updatedItensity.*maskImage);
```

```
        end
    end
end

%% final recovery using EPRY algorithm
loop = 50;
for tt=1:loop
    for i3=1:rec_arraySize^2
        i2 = seg(i3);
        if all(badImage ~= i2)
            i2;
            kxc = round((xRecoveryImageSize)/2+1-kx(1,i2)/dkx);
            kyc = round((yRecoveryImageSize)/2+1-ky(1,i2)/dky);
            kyl = round(kyc-(yInputImageSize)/2);kyh=round(kyc+(yInputImageSize)/2-1);
            kxl = round(kxc-(xInputImageSize)/2);kxh=round(kxc+(xInputImageSize)/2-1);
            unUpdate_lowResFT = objectRecoverFT(kyl:kyh,kxl:kxh).*CTF.*pupil;
            im_lowRes = ifft2(ifftshift(unUpdate_lowResFT));

            % update
            updated_im_lowRes =
(yRecoveryImageSize/yInputImageSize)^2*inputAmplitude(:,:,i2).*exp(1i.*angle(im_lowRes))
;
            updated_lowResFT = fftshift(fft2(updated_im_lowRes));
            objectRecoverFT(kyl:kyh,kxl:kxh) = objectRecoverFT(kyl:kyh,kxl:kxh) +
conj(pupil)./(max(max(abs(pupil).^2))).*(updated_lowResFT-unUpdate_lowResFT);
            pupil = pupil +
conj(objectRecoverFT(kyl:kyh,kxl:kxh))./(max(max(abs(objectRecoverFT(kyl:kyh,kxl:kxh)).^
2))).*(updated_lowResFT-unUpdate_lowResFT);
        end
    end
end

%% show images
objectRecover = ifft2(ifftshift(objectRecoverFT));
figure; imshow(inputImage(:,:,145),[]); title('Raw image 256x256')
figure; imshow(abs(objectRecover),[]); title('final amplitude');
figure; imshow(angle(objectRecover),[]); title('final recovery phase');
figure; imshow(log(objectRecover),[]); title('final log scale');
figure; imagesc(abs(pupil).*CTF, 'XData', [-kmax, kmax], 'YData', [-kmax, kmax]);
title('final pupil amplitude'); axis image; colorbar; xlabel('rad/m');ylabel('rad/m');
figure; imagesc(angle(pupil).*CTF, 'XData', [-kmax, kmax], 'YData', [-kmax, kmax]);
title('final pupil phase'); axis image; colorbar; xlabel('rad/m');ylabel('rad/m');
figure; imshow(log(abs(objectRecoverFT)),[], 'XData', [-dkx*xRecoveryImageSize,
dkx*xRecoveryImageSize], 'YData', [-dky*yRecoveryImageSize, dky*yRecoveryImageSize]);
title('final fourier'); xlabel('rad/m');ylabel('rad/m');
axis('on', 'image');
```