FRIEDRICH-SCHILLER-UNIVERSITÄT JENA

Leibniz ipht LEIBNIZ INSTITUTE of PHOTONIC TECHNOLOGY

Creating A low-cost Ptychography System Using The UC2 Platform

By

Sobhi Saeed

Supervisor

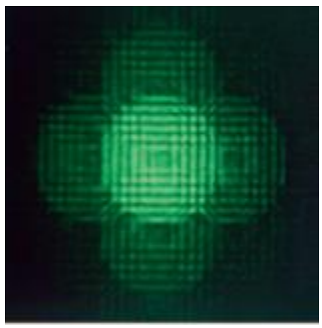
Dr. Dirk Boonzajer

Examiner

Prof. Dr. Rainer Heintzmann

Based on previous project by

Dr. Benedict Diederich



Abstract

Light detectors can only measure the intensity. However, the phase information is lost. Ptychography is one method to retrieve the phase information, ptychographic imaging is highly robust and quantitative in both phase and modulus.

In this project, we are implementing a Ptychography system on a UC2 (You See Too) setup [1], we were able to achieve a resolution of 18µm using a laser diode as an illumination source, a stepper motor to move the sample while detecting the relative positions by measuring the shift between two images; reference image at (0, 0) and another image recorded at the recording position.

We achieved our goal to build a functioning academic platform with acceptable resolution, the setup could be easily modified to help new researchers in the field understanding the impact of each system parameter, how to choose the parameters optimally, what are the limitations of the system and how can we improve the system.

Key words

Phase Retrieval

Ptychography

Correlation

Scanning Grid

Overlap

Diffraction Patterns

FOV: Field Of View

Propagation distance

Table of Contents

Table of Figures

1. Introduction

A scalar optical field is described by the valued function $\psi(x, y)$ in an (x, y) plane however, we can only directly measure the intensity $|\ \psi(x)\ |^2$, while the phase information is lost. Many techniques are used to retrieve the missing phase information; by retrieve phase we mean find the phase from intensity measurement. Ptychography is one of these techniques, in which a target specimen is translated to a series of positions relative to a localized probe wavefront and a diffraction pattern is recorded in each position [5]. These diffraction patterns can be used in conjunction with iterative algorithms [6] to reconstruct the object, providing an overlap ratio above 70% among adjacent illuminated areas of the sample (the recorded diffraction pattern in different positions).

Ptychography comes from the word ptycho which is a Greek word means to fold, was first introduced by Hoppe and Hegerl [4], they introduced it to describe a method to calculate the phase of Bragg reflections. However, over the last decade thanks to high computing processors, the field of Ptychography has had increasing interest because of its capability of retrieving quantitative phase information, and it considers the illumination source, besides it is easy to apply.

This project is a part of "Open UC2", a project was initiated by Benedict Diederich et. all [1].

We are using the setup shown in figure 1 to apply a Ptychography system.

The cubes shown in the figure 1 are printed with a 3d printer in IPHT Leibniz Institute, we use a Laser diode as an illumination source, a mirror to align the beam; center the beam onto the sample holder, vimba camera as an intensity detector. To move the sample holder, we use a stepper motor controlled by an Arduino. On the left side of figure 1 is our positioning system consisting of a LEDs grid, objective lens and an ESP32 camera.

In this project we focus on how to set up a Ptychography system, how to choose its system parameters and how can we reconstruct the object using PtyLab [3].

We refer the reader to [2],[5] for a deeper understanding of theoretical principles of Ptychography.

Figure 1, UC2 Ptychography Platform

Figure 2 illustrates the system design for the platform shown in figure 1

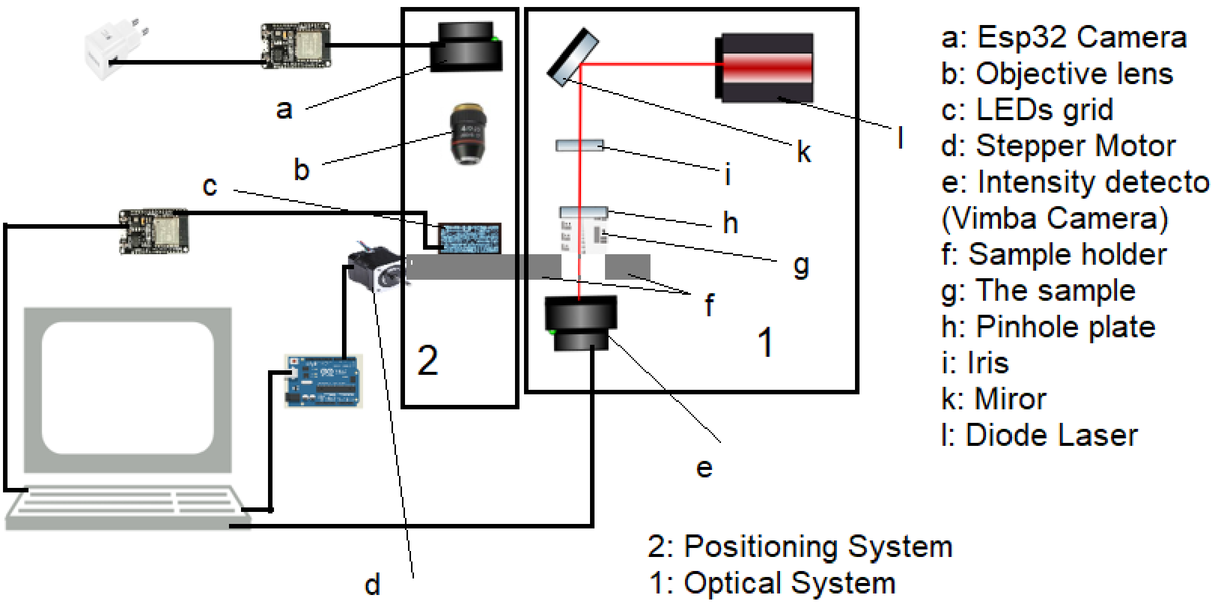


Figure 2, UC2 optical and position systems

2. Object Reconstruction

We use the word reconstruction because we first record diffraction patterns at different positions with their relative position and then we run the reconstruction code from PtyLab to reconstruct the object after retrieving the missing phase information.

The reconstruction codes in PtyLab apply iterative ptychographic engines to retrieve quantitative information of the phase for both computational and Fourier Ptychography.

To reconstruct the object using PtyLab codes we need to generate a Hdf5 file containing those subfiles:

- **Ptychogram:** recorded diffraction patterns.
- **Encoder**: Positions of those recorded diffraction patterns relative to (0, 0); the starting position.
- **Background:** recorded images with illumination off.
- **dxd**: size of the detector's pixels.
- **zo:** propagation distance from the sample to the detector.
- **Wavelength:** wavelength of the illumination source.
- **Entrance pupil diameter:** the beam size on the sample.
- **Nd:** number of pixels of the recorded diffraction pattern.
- **Direction:** optional parameter, although it should be known when we run the reconstruction code.

3. Scanning Grid and Moving Motor

3.1. Design of the Scanning Grid

Ptychography is based on illuminating the sample and recording diffraction patterns in different positions then reconstruct the object depending on the overlap of these diffraction patterns. Many criteria should be considered when designing the scanning grid. First of all, how far from the starting position can the sample be moved on each axis, it depends directly on the sample size and beam size on the sample; how much should we move the sample, and also the platform limitation how much can it move in each direction.

We should think as well about the overlap ratio between each two adjacent recorded intensities, the overlap ratio depends on the distance between adjacent points on the grid (scanning step) and the beam size on the sample (0.5mm in our case), figure 3 illustrates the overlap among the recorded diffraction patterns.

In order to generate a scanning grid for our platform, we use scanGrids code from PtyLab [3]. We choose spiral pattern.

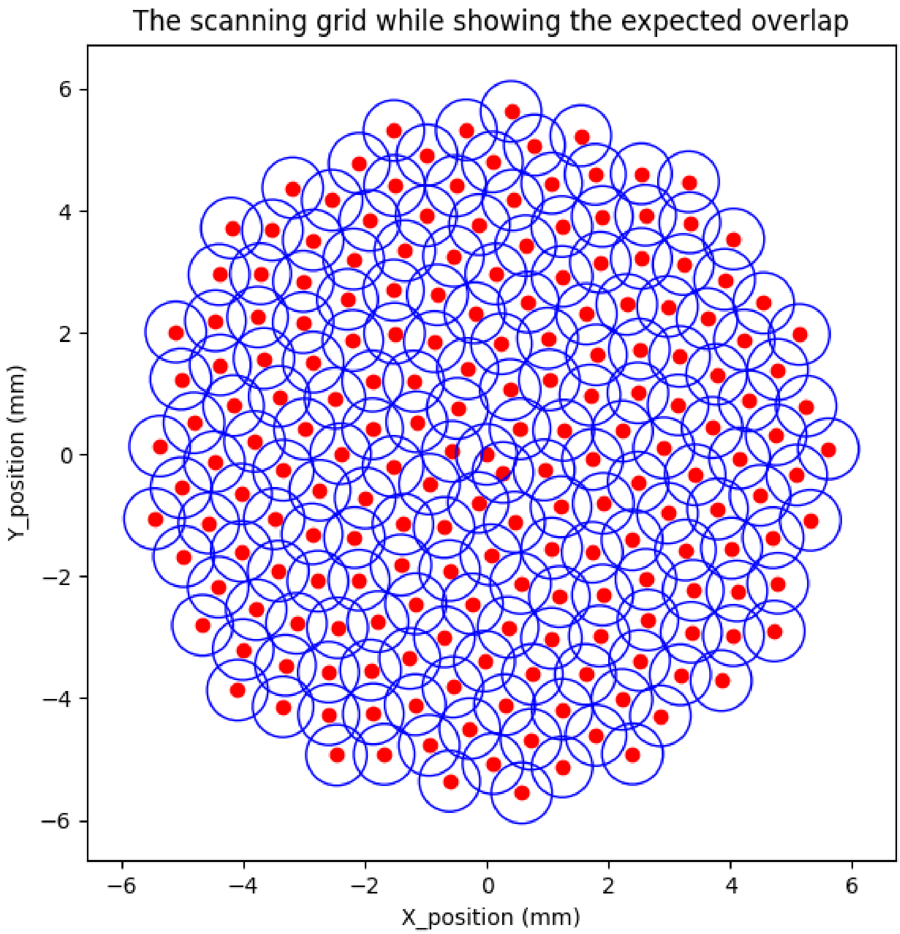


Figure 3, The scanning grid, the red dots are the recording positions of the scan grid, blue circles around represent the illuminated area of the sample, they diameter of 0.5mm which is the diameter of the beam on the sample.

In order to have a good reconstruction we need an overlap ratio between 75% to 80%, therefore we should change the current scanning grid, we do this by changing the scaling factor in scanGrids code, a new scanning grid to be seen in figure 4.

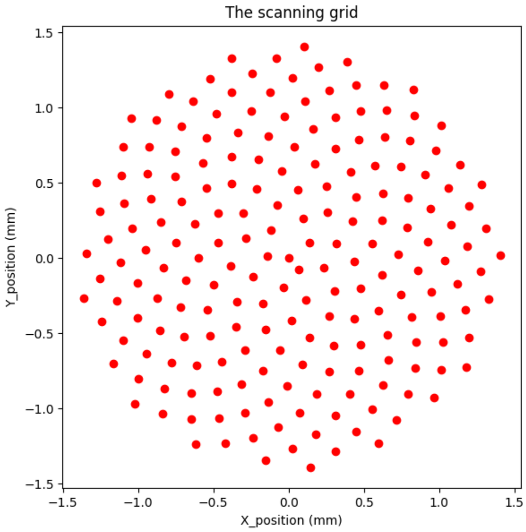


Figure 4, Spiral scanning grid for a ptychography system., 3mmx3mm wide.

We test the overlap ratio for the new grid as we did before by drawing circles with diameter equal to the beam diameter on the sample around each scanning position, figure 5 illustrates a high expected overlap ratio among diffraction patterns.

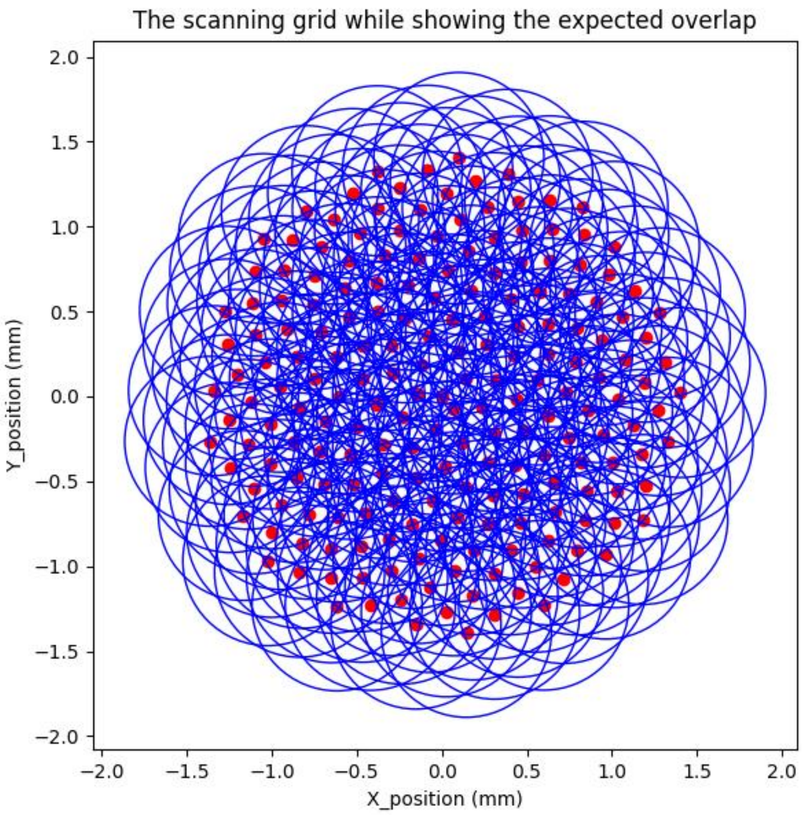


Figure 5, The red dots are the recording positions of the scan grid, blue circle around them has the diameter of 0.5mm which is the diameter of the beam on the sample.

Although, the overlap ratio is clear. In order to show the expected overlap between two adjacent diffraction patterns, we draw only two circles around two adjacent scanning points as shown in figure 6, we notice good overlap ratio, to make sure we can use this scanning grid, we calculate the average of the overlap ratio for each two adjacent circles, we get an average of 77%, an accepted ratio so we can use this grid; as long as we have entrance pupil diameter of 0.5mm and procced to the next step.

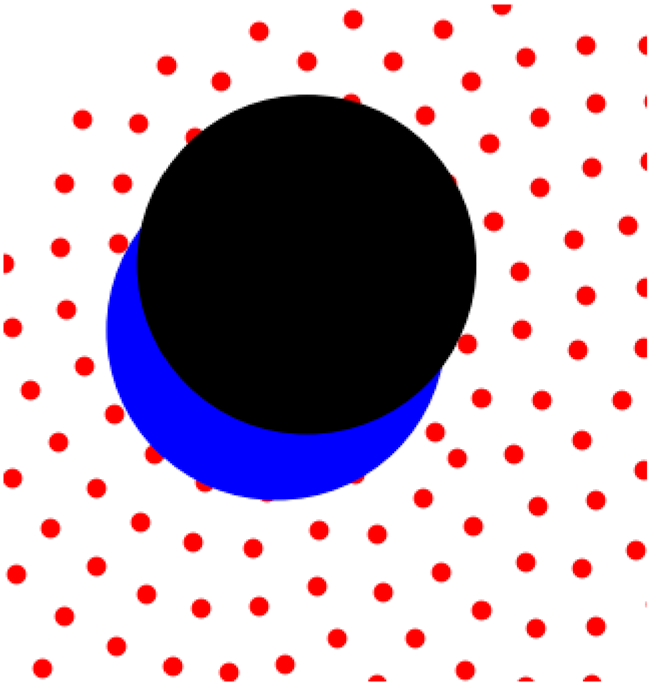


Figure 6, The intersection between the two circles illustrates the overlap between two adjacent diffraction patterns

3.2. Motor control

In order to record intensities in different positions, we need to move the sample holder latterly. To move the sample holder, we use a stepper motor, led by an Arduino Esp32 board and an A4988 driver shown in figure 7.

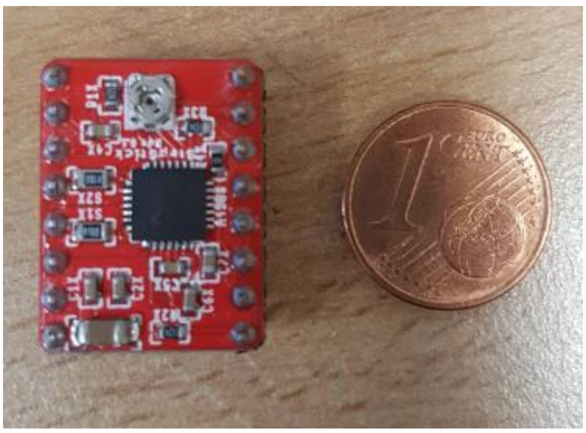


Figure 7, A4988 driver for the stepper motor

3.3. Motor Control algorithm

We send moving commands to the Arduino board through serial connection; the command contains steps number on both x and y axes and the directions of the movement. More details about the code and how to lead the stepper motor in general are to be found in the appendix.

We cannot depend on the motor's estimated movement to detect the position; hence we need another positioning system, first to detect the position accurately and also to apply a feedback loop to force the sample holder to record intensity in specific planed scanning position which were given to ensure a specific overlapping ratio among detected diffraction patterns.

4. Positioning System

Ptychographic imaging is highly robust, especially when compared to single-shot phase retrieval, but this robustness is balanced by a limitation: the resolution and accuracy of ptychographic images are restricted by the precision with which the specimen positions can be measured [5]. Unless a position correction algorithm were applied which also has its limitation.

To detect the position, we use an optical system consisting of a camera, an objective lens and an illuminated board fixed to the sample holder.

There are no specific requirements on the camera, we use an ESP32 camera (cost 3 Euro). We need also an adjustable magnifying lens (cost around 15 Euro).

As an illuminated reference, fixed on the sample holder, we use a led grid (OLED display ssd1306, cost ~10 Euro) instead of illuminating a dark reference.

Figure 8 shows all 3 parts of the positioning system.

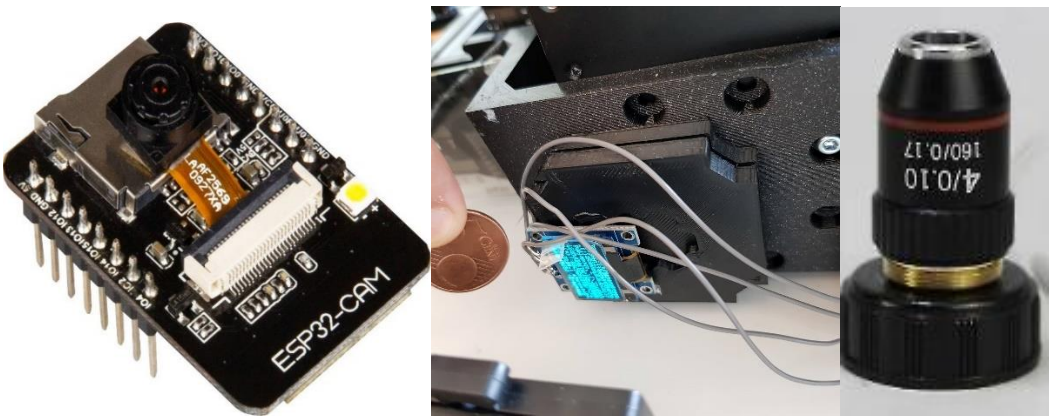


Figure 8, Positioning System Components, from left to right; ESP32 Camera, LEDs grid OLED display ssd1306, Objective Lens 4/0.10

Known for us is the pixel pitch of the LEDs grid which is 170µm, we going to magnify this LEDs-grid's illumination pattern, and measure the shift using these magnified images, figure 9 shows how these magnified images look like after some filtering.

After each movement we measure the shift between the original image which has the relative position (0, 0) and the captured image by EPS32_Cam in current position. We measure the shift using phase_cross_correlation from skimage.registration [4].

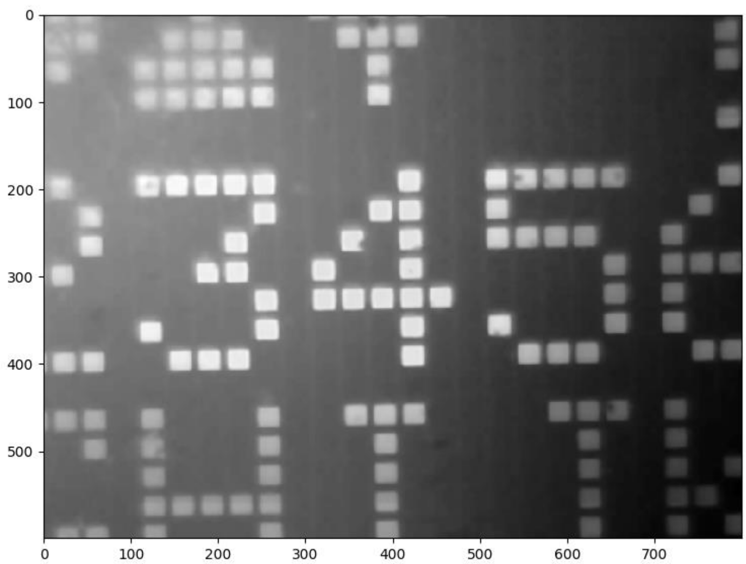


Figure 9, Filtered magnified image for a part of the LEDs grid

4.1. Shift to Distance Conversion

Before we start moving the sample and measuring the shift, we should calculate the distance to pixel conversion ratio because we are going to calculate the shift in pixels first then convert it to micro meters using this conversion ratio. In order to calculate this ratio, we first use a periodic squares illumination pattern which is shown if figure 10.

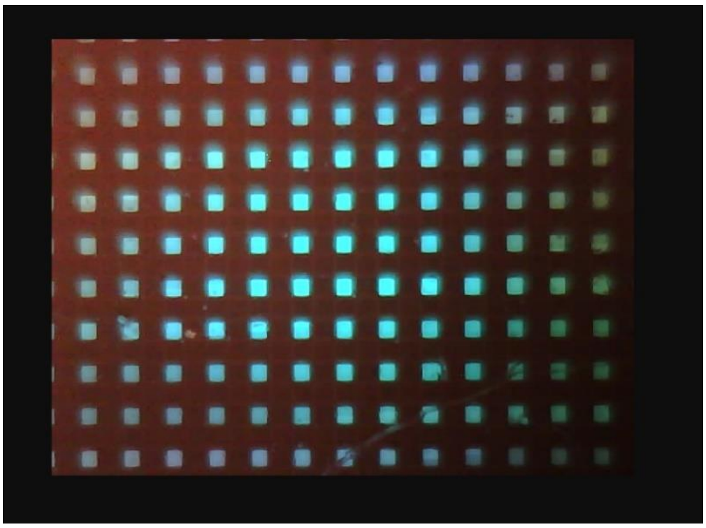


Figure 10, periodic illumination pattern with one pixel pitch difference between adjacent squares

The periodic pattern is better for calibration because we can minimize the error when calculating how many pixels from the magnified detected image by ESP32 cam there are per one LEDs grid pixel pitch.

To do this calibration we first process the captured image and apply Sobel filter to see the edges of the squares better as shown in figure 11.

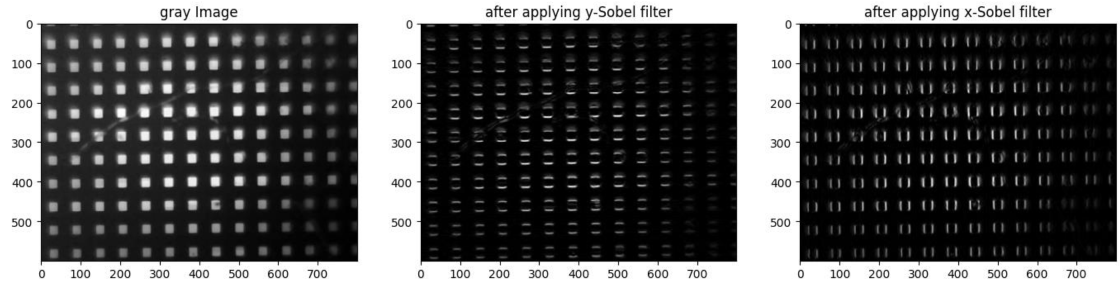


Figure 11, from left to right: processed magnified image of magnified part of the LEDs grid, same image after applying y-Sobel filter, same image after applying x-Sobel filter

To explain the calibration process, we capture a small part of the Sobel-filtered images, the program will ask the user to detect key points as shown in figure 12.

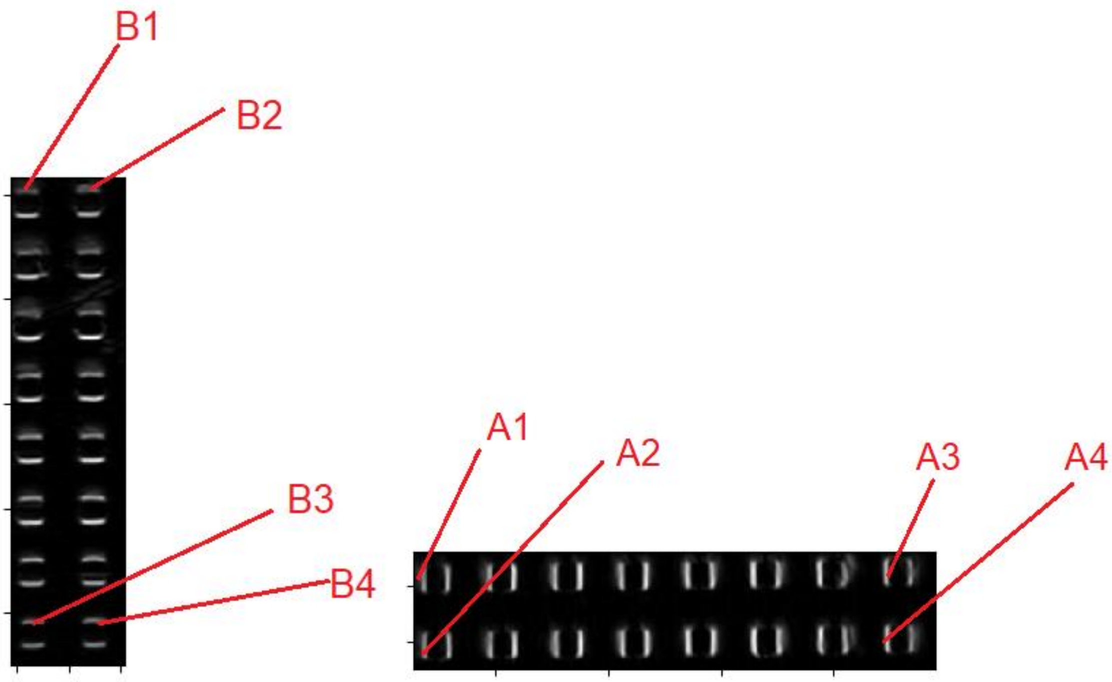


Figure 12, The detected key points, and other detectable key points with known pixel pitches between them

We write down the x coordinates of each point between A1 and A3 as well as for the points between A2 and A4 and the y coordinates of each point between B1_B3 and B2_B4, we subtract the reference points (A1, A2, B1, and B2) coordinates from the detected coordinates (the coordinates of the points between A1 and A3) and then divide the subtraction result over the pixel pitches between the detected point and the reference point (7 pixel pitches between A3 and A1 for example), then we calculate the average of how many detector's pixels are needed to represent the magnified pixel of the LEDs grid. This number represents the magnification ratio and also relates the shift with LEDs grid known pixel pitch (170um).

By dividing this convert ratio over the known pixel pitch of the LEDs grid we get the conversion ratio we need which is distance per pixel, **we get a conversion ratio of 5um per pixel**. An example table of these calculation is provided in appendix (table 1)

4.2. Magnifying Ratio

After calculating the distance per pixel conversion ratio, we change the displayed pattern on the LEDs grid (refer to appendix 12.1 for details on how to change it). When calculating shift we avoid periodic patterns because they lead to fake results sometimes, we chose the pattern shown in figure 13 because there is no repeated pattern (key objects).

Figure 13,Characters illumination pattern, the whole LEDs grid

When choosing the magnification ratio, two parameters should be considered, the magnification ratio affects directly the distance per pixel conversion ratio, higher magnification ratio means lower distance per pixel conversion ratio, hence higher resolution. The magnification ratio controls the scanning field too. To measure the shift between two images, they should have common features, figure 14 illustrates how the original and shifted images have common objects (features) in different positions, we measure the shift at each position between all encoder images (the images of the LEDs grid at each scanning position) and the reference image at (0, 0) directly to avoid accumulated error, so all the encoder images should contain objects present at the reference encoder image at (0, 0), otherwise we would be measuring shift between two different frames have nothing in common and the results will be wrong.

To get more robust results, we apply gradient filter on both images before measuring the shift.

A possible way to widen the scanning grid is to consider furthest points in each direction as new reference points, calculate a new shift to them and add it to the distance between the new reference and original reference to have the actual relative position.

4.3. Proof of concept (Reverse Shift)

To test the shift, we save the shifted images and their measured shift before converting this shift to distance and apply a reverse shift; the resulting image of this reverse shift should be the original image recorded at (0, 0). Figure 14 illustrates the original shift, a shifted image and the shifted image after applying reverse shift on it.

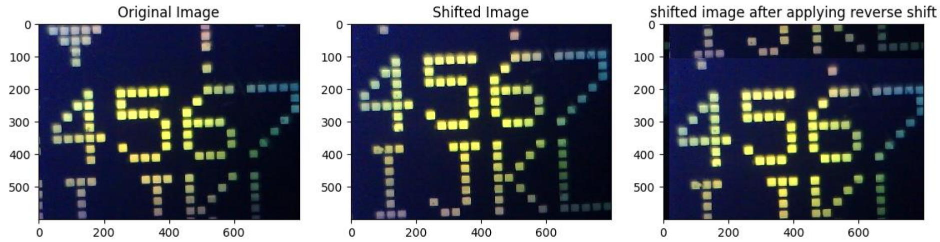


Figure 14, Testing the measured shift by applying a reverse shift on it and compare it with the original image

4.4. Forcing the Planned Recording Positions

We mentioned in chapter 3.1 the importance of the overlap ratio among recorded diffraction patterns, in order to ensure the expected overlap ratio, we apply a feedback loop forces the sample holder to go to the given (planned) recording positions before adding a new diffraction pattern to the Ptychogram. Figure 15 shows how does this feedback loop work.

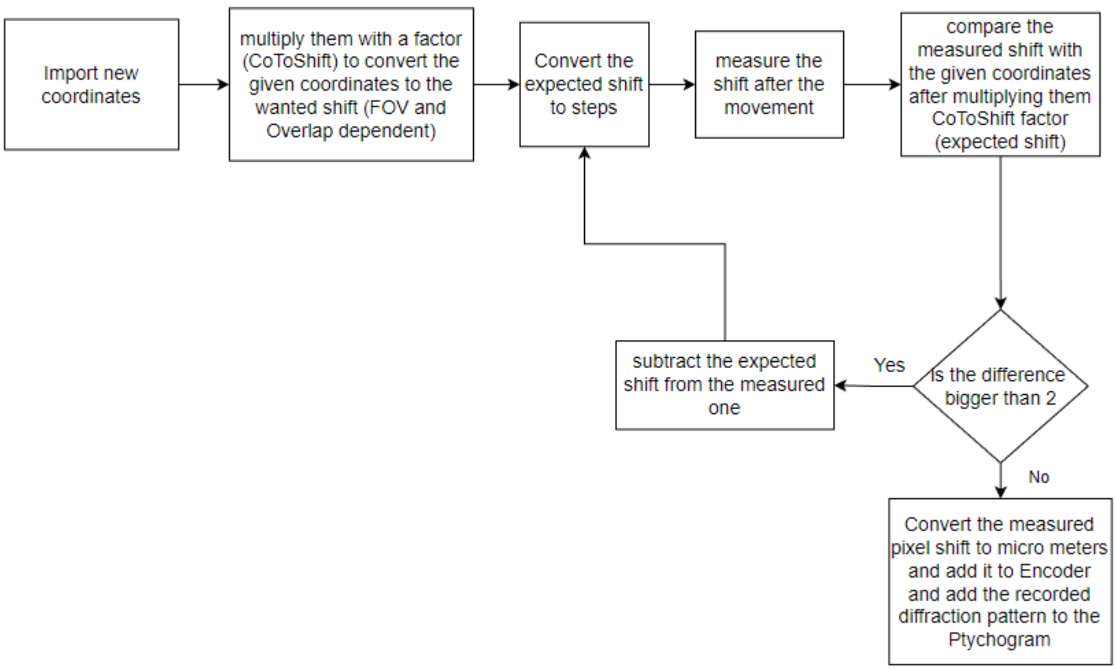


Figure 15, Forcing planned recording positions feedback loop

We test the control system with the afro mentioned feedback loop by comparing the given coordinates with the measured coordinates, the coordinates are given in mm, to compare them with the measured shift, we should first convert this shift from pixel to distance in mm. Figure 16, 17 shows the measured coordinates and the given coordinates in separate figures, and on the same figure respectively.

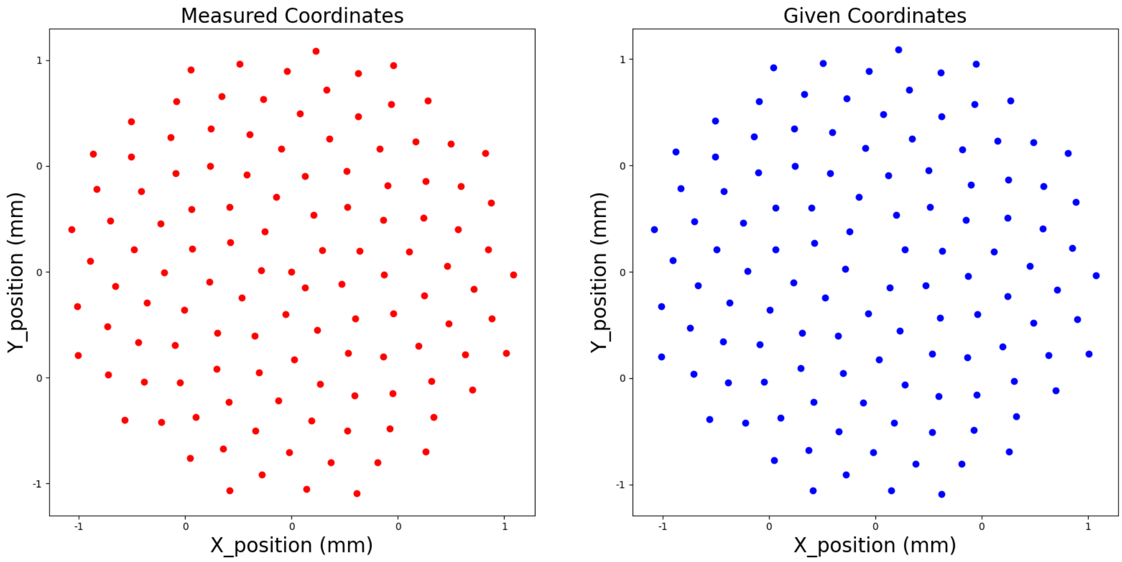


Figure 16, given recording positions Vs. measured recording positions in different figures

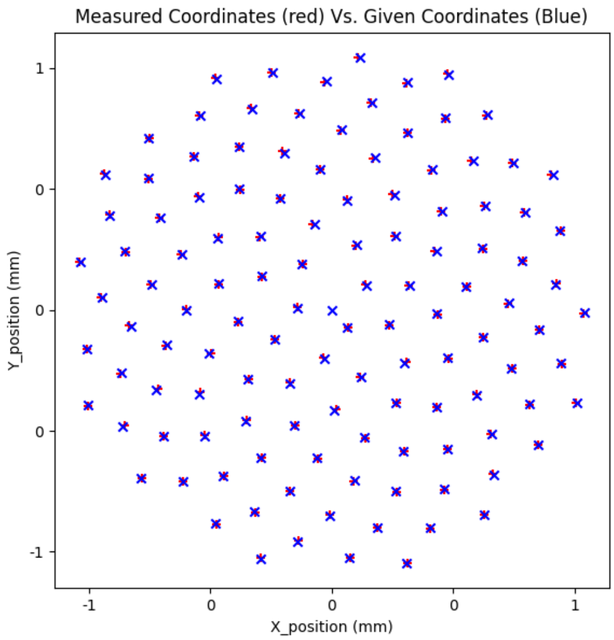


Figure 17, given recording positions Vs. measured recording positions on the same figure

As shown in figure 17, the difference between the given and measured coordinates is as expected less than 2pixels = 10µm (refer to figure 15, the feedback loop).

As we mentioned before, recording at the correct position is important to keep the overlap ratio, to evaluate this result, we divide the maximum difference 10µm, over the beam size on the sample (0.5mm), this division shows how much could the overlap ratio be changed.

$$\Delta = \frac{10\mu\mathrm{m}}{0.5mm} = 0.02$$

Δ is negligible comparing to 77 (current overlap ratio), ergo we can use the loop and expect the planned overlap ratio.

4.5. Positioning Error

Before we wrap up positioning chapter, we calculate the expected positioning error, first we list error sources.

Positioning Error Sources:

1. **Shift Error**
   Subpixel accuracy, up to 0.01 pixel. However, there is no possible way to test such high accuracy. Due to lack of testing method, we assume a shifting error of **0.2 pixel** (0.1 on the x axis + 0.1 on the y axis)
2. **Calibration error**
   When calculating Distance Per Pixel conversion ratio (refer to 4.1) there is an error detecting the key points, comparing the highest number of pixel (for the magnified image) per pixel pitch of the LEDs and lowest number we end up with one pixel error. However, we include many points in this calculation to minimize the error, after minimizing this error we end up with an estimated error of **0.1 pixel.**
3. **Datasheet error**
   We built the calibration on knowing the pixel pitch of the LEDs grid which is 170µm. No information from manufacturer on about how accurate is this so we assume 1 µm error per pixel pitch, this gives an error of 0.03µm

Considering all the mentioned factors we have a total error of 0.3pixel multiplied with shift to distance conversion ratio (4.96µm per pixel) plus the Datasheet error

$$e = 0.3 * 4.96 + 0.03 = 1.58\mu\mathrm{m}$$

There is always forgotten errors in such systems, we round up the error to 2µm.

The positioning error with good estimation is 2 micro meters!

5. System Parameters

5.1. Background

In order to remove the background noise, we record some images with the same detector used to record diffraction patterns after turning off the illumination keeping in mind those background images should have the same pixels number and same exposure time as the recorded the diffraction patterns.

Before we run the reconstruction code, we run a preprocessing code using the background images, it calculates the average intensity of the acquired background images and subtracting it from the Ptychogram images.

5.2. Wavelength (λ), Propagation distance (z) and Detector Pixel-Size (dxd)

Wavelength, Propagation distance and detector pixel size impact the resolution, the Field of View (FOV), the Numerical Aperture (NA) and Fresnel number.

Fresnel Number $N_f \gg 1$ in this case regardless of small changes in λ *and* , hence we have to use ASP propagator in the reconstruction code, and we do not consider it when choosing λ *and* z.

The resolution in Microscopy is defined as the smallest distance between two points on a specimen that can still be distinguished as two separate entities.

$$\Delta x = \frac{\lambda . z}{D} \tag{1}$$

Δx: Resolution [m]

D: the side length of the detector [m]

Z: the propagation distance[m]

λ: Wavelength [m]

Smaller Δx means higher resolution. However, we want a wide FOV which depends on λ *and* z as well, as shown in equation 2.

$$FOV = \frac{\lambda z}{\Delta q} \tag{2}$$

Δq: side length of the detector's pixel (dxd in Ptychogram)

The reciprocal relation of resolution is shown in figure 18

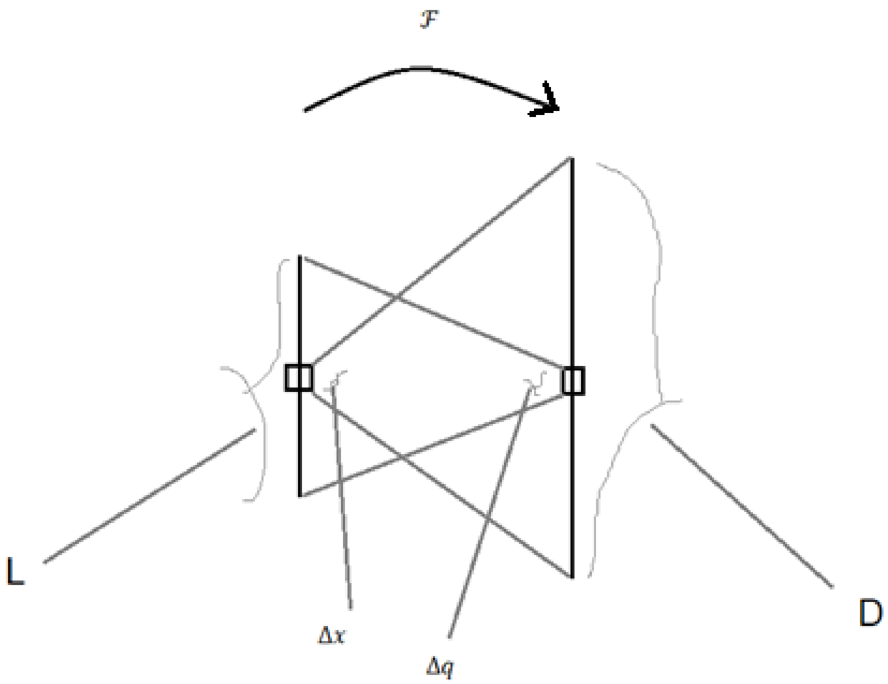


Figure 18, Reciprocal relation of resolution, L is the side length of the illuminated area of the specimen, D: side length of the detector, Δx: the resolution, F: Fourier Propagator, Δq: Pixel pitch of the detector

Having a wide FOV is not essential in Ptychography because the sample is moving and a new diffraction pattern is recorded at each position, the total detected area will be the sum of the scanning-grid width and the full FOV of one detection (half from each side).

The beam size on the diameter is 0.5mm, and the resolution can only be improved by ptychography if dark-field data are processed [2], hence FOV should include the illuminated part and surrounding dark-field area. FOV should be at least 3 times the width of the beam size on the sample. We Choose 1.5mm as a minimum value for FOV. Now we choose λ *and* z values to get the best possible resolution while keeping FOV > 1.5mm.

We are operating in the visible spectrum $\mathbf{400nm < \lambda < 700nm}$.

Our platform already has an available Laser diode with a wavelength of 450nm, we can use this one.

The setup has also a mounted Vimba camera with pixel size of $\Delta q(dxd) = 3.45\mu m$ and Side-length (D) = 3.75mm

(D) = minimum pixel numbers between Height and Width * pixel pitch

$D = 1088 * 3.45\mu m = 3.75mm$

We use the minimum between Height and Width here because we need a square image.

By choosing $z = 14mm$, referring to equations 1 and 2, we get a resolution, FOV and NA of:

$$\Delta x = 1.68\mu m$$

$$FOV = 1.83mm$$

$$NA = n * \sin(\Theta) = 0.134$$

$$n = 1, propagation\ in\ air$$

Lower NA means lower resolution but higher brightness, normally one would aim for a higher NA in Ptychography. However, we have good resolution, ergo we do not need to rebuilt the setup.

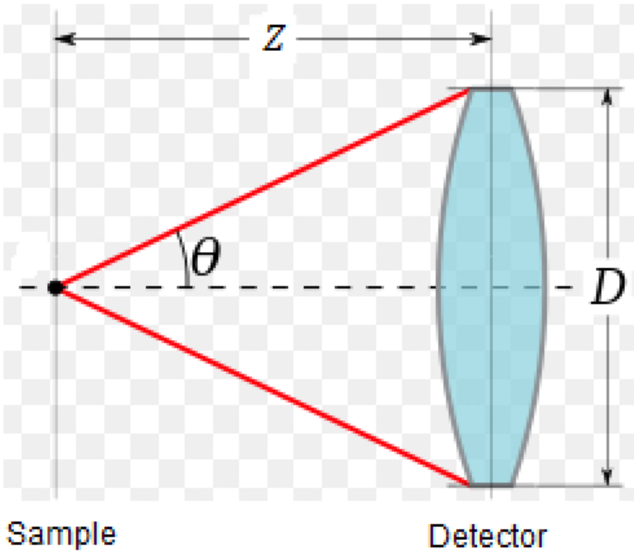


Figure 19, NA calculation, left is the sample plane, right is the detector plane, light propagate in air between them.

Before we wrap up this chapter, we refer to figures 28, 29 in results chapter which conclude the importance of choosing the optimal propagation distance, $\Delta x = \frac{1}{7}mm = 142\mu m$ were the highest achievable resolution with propagation distance of 41.5mm, while the resolution goes high to $\Delta x = \frac{1}{54}mm = 18.5\mu m$ when changing the propagation distance to 14.5mm. Same positioning system were used in both cases.

5.3. Entrance pupil diameter

One important parameter of Ptychography is the beam size on the sample (Entrance Pupil Diameter), it affects the overlap ratio, ergo the scanning grid design (refer to 3.1) it also impacts the quality of the reconstructed object.

When using a relatively wide beam on the sample, this could result in losing phase information at high frequency. However, choosing a very small beam size means low overlap ratio which compromises the reconstruction algorithm and we will be obligated to choose very small translating steps, which increases the scanning time.

The beam size on the sample should be chose depending on the sample resolution and the system divergence, in our system we were not able to add for example an objective lens before the detector, hence we have low divergence.

We choose 0.5mm as a beam size on the sample.

The illumination system had a focusing lens at first. Although the focal point was not where it supposed to be and we have in our setup a limitation on the distance from the focusing lens to the sample, due to this limitation we had to remove the focusing lens form the system.

Without the focusing lens, to control the beam size on the sample and measure it precisely, we add a plate with a pinhole right before the sample.

Figure 20 illustrates the designed pinholes plate.

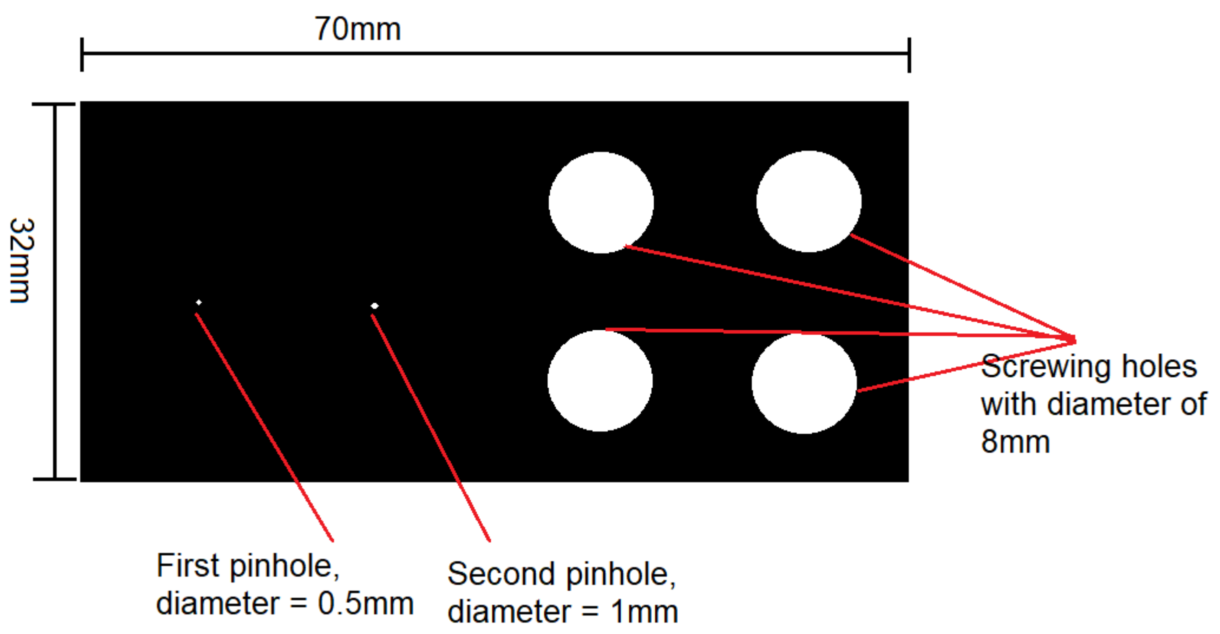


Figure 20, designed plate with pinholes to control the beam size on the sample and apply spatial filtering

The plate was manufactured at IPHT Leibniz institute workshop, shown in figure 21, we choose the black color to minimize reflection Artifacts.

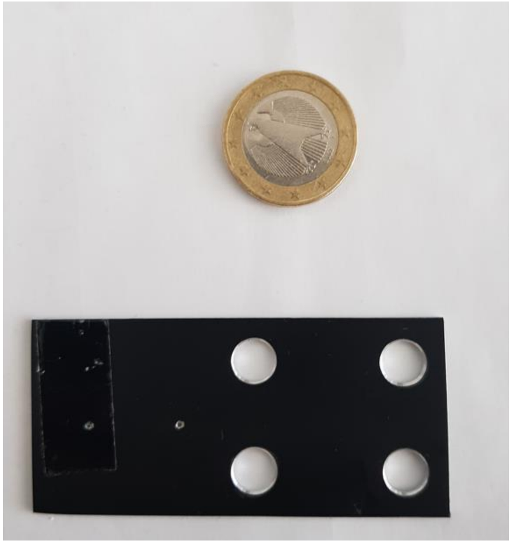



Figure 21, manufactured spatial filtering plate (pinholes plate)

The tape on the pinhole makes the beam diverge, although the divergence angle is low resulting a beam curvature of $2 * 10^{-3}$ (estimated value, we cannot measure it accurately).

Why should the beam diverge?

- To use as many pixels as possible on the detector
- Avoid missing information from the sample, if for example these details are positioned on the edge of the illuminated area of the sample.
- High spatial frequency on the sample will be easier to detect.
- Background scattering better detected

How to measure the Entrance Pupil Diameter if we are not applying spatial filtering?

A possible way to measure the beam size on the sample is using a resolution target (shown in figure 22), and center the beam on a known clear pattern, we used 9, the distance between the lines is $\frac{1}{9}mm$.

Figure 22, Thorlabs resolution target and the pattern we used to measure the beam size on the sample

Figure 23 shows the detected image when illuminating the "9" lines. Pixel size is known for us, we detect how many pixels separating the highest and lowest points in a single line and. Also known for us is the distance between the lines. From these two known parameters, we can measure the beam size on the sample; by scaling the number of pixels of one line to the number of pixels of the whole light-field diameter.

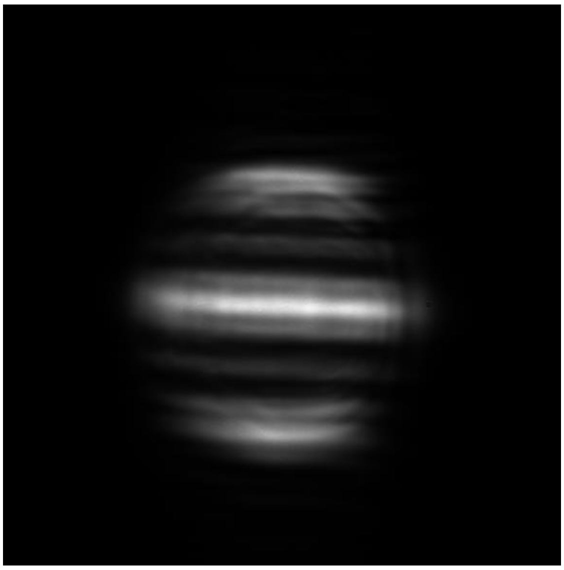


Figure 23, detected image to measure the entrance pupil diameter

6. Recording The Diffraction Patterns

Retrieving the phase information in Ptychography is depending on the overlapped diffraction patterns, optimal recording of these diffraction patterns is essential.

An optimal detected intensity would have a ratio of 33.3% light field and 66.3% dark field so we can process the dark field and be able to detect high frequency details in the sample as shown in figure 24.

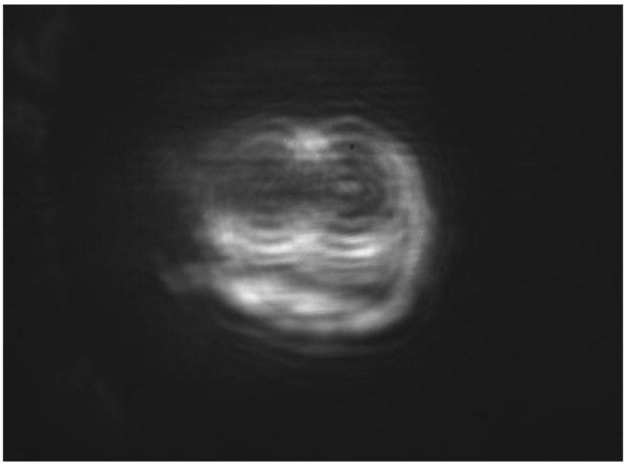


Figure 24, recorded diffraction pattern on the Vimba viewer

Although the dark-field _ light-field ratio in figure 24 is very good, the contrast is low.

We should increase the contrast by using all the detector's bits range while avoiding over exposing (this is achievable through changing the exposure time and the power supply for the illumination system), and improve the alignment of the beam and the sample.

After removing the focusing lens and adding the plate with the pinholes right before the sample, the new recorded diffraction patterns are smaller but the contrast is much better as shown in figure 25.

Figure 25, recorded diffraction pattern on the Vimba Viewer

Before saving the detected diffraction pattern, we center the image and bin the pixels for faster processing, as shown in figure 26.

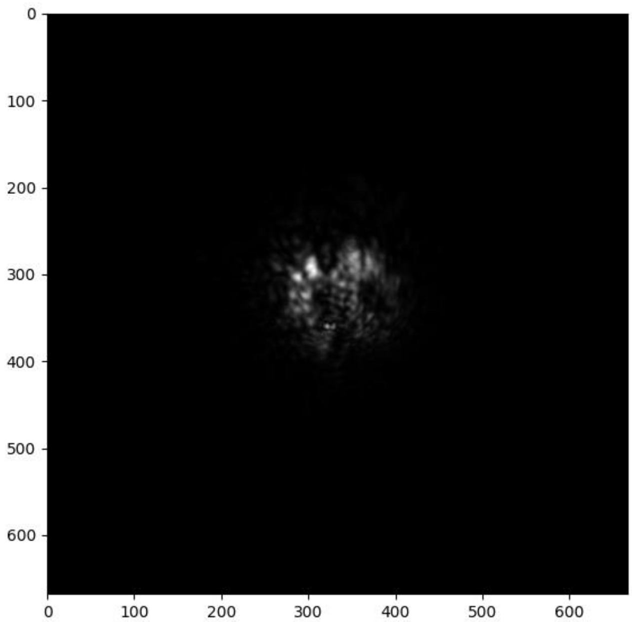


Figure 26, Recorded diffraction pattern after centering the light-field and binning the pixels

7. Programming Codes Sequence

StepperMotor
Uploaded on the stepper motor controller

OLED (periodic squares)
Uploaded on the LEDs grid

ESP32CamStream.
Uploaded on ESP32_Camera contr oller

Prepare the Setup, upload the Arduino Code

DistancePerPixelCalibration.
Python Code shows the LEDs grid and ask the user to enter coordinates of some key points, it yields the distance in um per pixel and saves it in the IniConfig

OLED (Characters Pattern)
Arduino Code, uploaded on the LEDs grid

Run the first Calibration and change the LEDs grid pattern

StepsToShiftCalibration
It yields the conversion ratio between Motor Steps and measured shift and saves it in the IniConfig

COM (Center Of Mass)
It shows one diffraction pattern, and it has the same centering algorithm as the main code.

ScanGrid
It generates the scanning grid and saves it as a text containing the coordinates,

Measure Shift
Python

- Run all the needed Calibrations
- Create a Scanning grid
- Run the shift test

CreatePtychogram
it creates the hdf5 file we need to reconstruct the object using PtyLab [3]

Create the hdf5 file which contains: Ptychogram, encoder, and the systems parameters

Codes for testing the Ptychogra m run them before starting the reconstruct ion

ReadHdf5
simple code to check the Hdf5 files

ReverseShift
Test the shift

CompareGivenAndMeasuredP ositions
Compare the given and the measured positions

PreProcessing
Process the Ptychogram before running the Reconstruction

CP_Reconstruction
It applies Ptychography engines to reconstruct the object

PtyLab codes generate a hdf5 contains information about the object

---

PhaseUnwrapping
Extract the object information from the hdf5 file generated by CP_Reconstruction

Plot the reconstructed object

8. Results

Although Ptychography is used to retrieve phase information, we use Thorlabs resolution target as a sample to test the resolution of our setup.

For the first result we had a propagation distance z = 41.5 mm. After generating a Ptychogram (saving it with the other parameters in an Hdf5 file) and preprocessing it by subtracting the background images intensity from the recorded diffraction patterns and then run Computational Ptychography reconstruction code from PtyLab, we were able to reconstruct an object with resolution of 140μm only, as shown in figure 27.

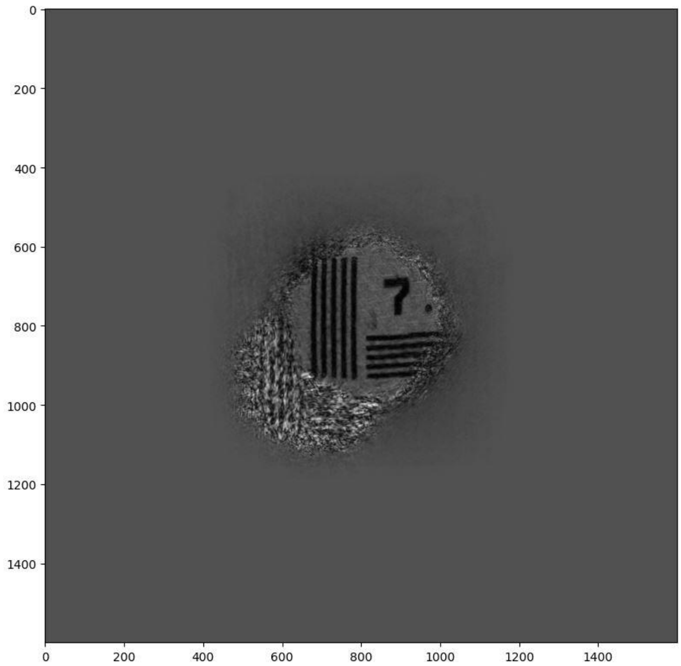


Figure 27, Reconstructed object with UC2 Ptychography system

We tried another reconstruction for another pattern on the target, this pattern has higher resolution (between 30 – 40 μm), the results were not satisfying, as shown in figure 28,

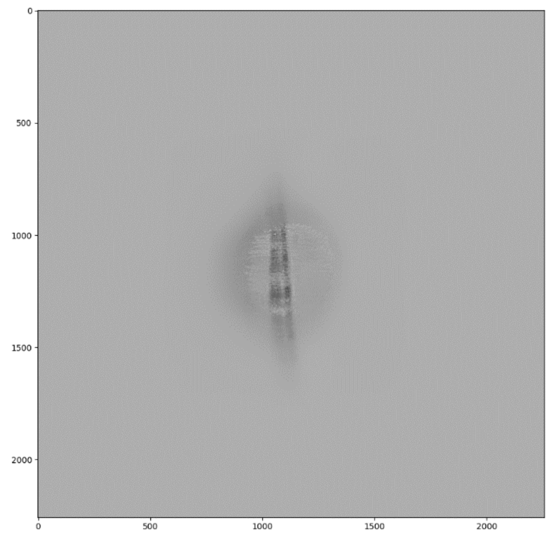


Figure 28, first Reconstruction try with UC2

To improve the results, we modified the setup to change the propagation distance to z = 14.5mm, while using a black spatial filtering plate to avoid reflection artifacts, we were able to achieve ~20µm resolution, as shown in figure 29.

The bad quality of the reconstructed image is due to low divergence because it reflects on how many detector's pixels were used to record each diffraction pattern.

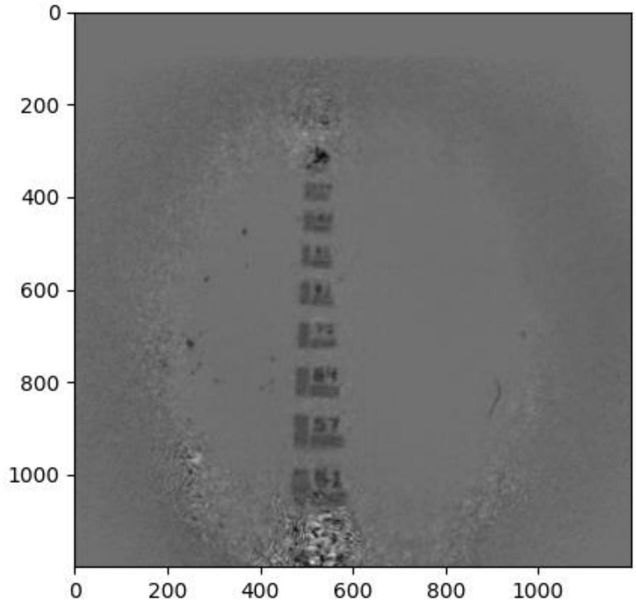


Figure 29, Resolution Target reconstructed with the UC2

9. Future Possibilities

After proving the functionality of this system, we could redesign the setup to solve the alignment and low divergence problems to increase the resolution to enable us to retrieve phase information from micro specimen.

Improve and optimize the positioning system to widen the scanning field without decreasing its accuracy.

Build a friendly interface for the system to enable new researchers in the field of Ptychography to work on an adjustable platform.

10. Conclusion

We achieved our goal of building an open-source low cost Ptychography system with ~20μm resolution, which could be used as an academic platform to teach Ptychography and image processing in microscopy, and also to help students understand control algorithm.

11. References

1. A versatile and customizable low-cost 3D-printed open standard for microscopic imaging by Benedict Diederich et all. (2020) 11:5979 | https://doi.org/10.1038/s41467-020-19447-9
2. Ptychography 2019 by John Rodenburg and Andrew Malden
3. PtyLab: a cross-platform inverse modeling toolbox for conventional and Fourier ptychography by Lars Lötgering et all.
4. https://scikitimage.org/docs/stable/api/skimage.registration.html#skimage.registration.phase_cross_correlation
5. An annealing algorithm to correct positioning errors in ptychography A.M. Maiden et. all. Ultramicroscopy 120 (2012) 64-72
6. A phase retrieval algorithm for shifting illumination by J.M. Rodenburg, H.M.L. Faulkner,, Applied Physics Letters 85 (20) (2004) 4795–4797.
7. An improved ptychographical phase retrieval algorithm for diffractive imaging by A.M. Maiden and J.M. Rodenburg Ultramicroscopy 109 (2009) 1256–1262

8. Computational illumination for high-speed in vitro Fourier ptychographic microscopy September 22, 2015 by Lei Tian et. all.

12. Appendix

12.1. Programming Codes

StepperMotor

An Arduino code controls the movement of the Motor, the control algorithm depends on get a string order through serial, id decodes this order to get movement directions on x_y, then change the value (High _ Low) of both direction pins accordingly. The serial message contains also the steps number on both axes.

Movement is achieved through a loop, where we switch the pins between high and low for each step.

ESP32CamStream.

Arduino Code, should be uploaded to the ESP32_Camera's Arduino controller, for images acquisition, it connects the Arduino to the PC via WIFI through a mobile hotspot created from the PC.

This Code acquires ESP32_Cam images and creates two links, one to watch the camera stream, and the other link shows one image.

OLED (Periodic Squares or Characters)

Arduino Code, should be uploaded to LEDs grid Arduino Controller, it creates the illumination pattern on the LEDs grid.

We need the periodic squares pattern for calibration; pixels to distance conversion ratio figure 10. However, a non-periodic pattern is needed when measuring the shift, hence we use the Characters Pattern, figure 13

The LEDs grid is connected to an EPS32 Arduino, we just use an example code (Characters) from AdafruitSSD1306.h, the library could be downloaded from https://unsinnsbasis.de/oled-display-ssd1306

Tips for Arduino users:

When connecting to Arduino eps32 on windows 10 we face an "unknown driver" to solve this problem, we search for FT232R driver and install it.

When using serial connection between Python and Arduino, and we want to test the serial port from Arduino Ide we should use (No New Lines) in testing.

StepsToShiftCalibration

Python code, it yields the conversion ratio between Motor Steps and measured shift and saves it in the IniConfig file.

In order to get the conversion ratio, we move the motor back and forth on both axes many times, we save the shift of each movement and divide it over the number of steps, calculate the average, which would be the StepsToShift conversion ratio.

DistancePerPixelCalibration.

Python Code shows the LEDs grid and ask the user to enter coordinates of some key points, the key points are located 1, 2, 3, 4, 5, ... pixel pitches away from the reference point the distance in um per pixel and saves it in the IniConfig file

ScanGrid
Python code, it generates the scanning grid and saves it as a text containing the coordinates,

COM (Center Of Mass)

Python Code, it shows one diffraction pattern and it has the same centering algorithm as the main code. the user should run it before recording the Ptychogram to check if the diffraction patterns are centered and there is no overexposing

Measure Shift

This code is to test the possibility of using the designed scanning grid, first we need to convert the remotest distance in the scanning grid to steps using this calculation and then run the code after giving the resulting steps on x, y, the shift should be close the expected shift otherwise we have to check the positioning system or redesign the scanning grid and choose a smaller one.

Coordinates (mm)	1.1
CoToShift	4.96E-03
Pixels	2.22E+02
xfactor	0.027
Steps	8.21E+03

CreatePtychogram
It generates the scanning grid and saves it as a text containing the coordinates,

ReadHdf5

simple code to check the Hdf5 files

ReverseShift

It applies a reverse shift on the encoder images to test the measured shift, we should get the original image after applying reverse shift.

CompareGivenAndMeasuredPositions

Compare the given and the measured positions

PreProcessing

The resolution of a perfect lens can only be improved by Ptychography if dark-field data are processed. [2]

Process the Ptychogram before running the Reconstruction

CP_Reconstruction

It applies Ptychographic engines to reconstruct the object

PhaseUnwrapping

Extract the object information from the hdf5 file generated by CP_Reconstruction

12.2. Pixel to distance calculation

	A1	A11	A12	A13	A14	A15			
X	463,53	505,48	548,49	590,5	632,49	675,54			
y	379,48	378,52	377,49	376,52	375,5	374,54			
distance in pixels to A1		41,96098307	84,9833	127,0045	169,0068697	212,0675			
pixel pitches to A1		1	2	3	4	5	Average	pixel pitch of the LEDs grid	distance/pixel
Camera pixels per LED grid pixel pitch		41,96098307	42,49165	42,33483	42,25171742	42,41351	42,29053868	1,70E-04m	4,02E-06m

All codes are available on my GitHub account, please visit https://github.com/SaeedSobhi

Thank you for reading our paper,
best wishes,
Sobhi Saeed