UC2 Fourier Ptychographic Microscopy

Samuel Perales and Alec Vercruysse

May - July 2020

1 Introduction

Imaging systems often require a trade off between resolution and field-of-view (FOV). Either we can take a very wide picture that we can't zoom very far into or an up-close, high-resolution image. Taking an image that can maximize on both of these metrics would be extremely helpful in many fields. Specifically, biological imaging with a high resolution and wide FOV would allow for capturing the information in millions of cells simultaneously as opposed to only small subregions of some sample. The issue is that this trade off is ingrained into the typical optical setup; capturing more of a sample simply means we will have a lower quality image.

However, by utilizing some clever imaging strategies along with the help from a computer, we can bypass this constraint to maximize both of these metrics. This idea of computational imaging has been an important part of modern day optics as it can solve many issues beyond this one constraint. For example, when taking an image, our sensor only captures the intensity of the light it receives and seems to lose other information like the phase of the light. This phase is valuable because it can give us a topographic mapping of a sample of interest. Other computational methods might allows us to fully recover 3D images of a sample or even enable imaging through scattering media.

Incorporating a computer into the imaging process clearly has many benefits in terms of different imaging modalities but it also allows us to sometimes build our systems for quite cheap. With a strong enough algorithm, many of these types of images can be recovered from very low quality input images, meaning we can have a more simple optical setup with fewer lenses than many modern setups require. Sometimes even using a lens is entirely unnecessary!

In this project, we explore the computational imaging method known as Fourier Ptychographic Microscopy (FPM) (1) which is an algorithm that allows for recover a high-resolution and wide-field-of-view image from many low-resolution input images. In attempt to both standardize and minimize the cost of the project, we use the UC2 open-source, open-standard (2) (3) to 3D print optical modules which constitute the system. We also want to explore various iterations of FPM which can allow for recovery of additional information such as Embedded Pupil Recovery (EPRY) (4) which can help to calculate and account for aberrations in the lens the system uses.

2 Methods

2.1 Theory

The idea of the Fourier Ptychographic Microscopy algorithm is similar to Phase Retrieval (5) in that it consists of projecting between two domains, Real space and Fourier space, and then updating an estimate based on some constraints in the respective domain. A key assumption we make is that illuminating a thin sample with the planewave with wavevector (k_x, k_y) is equivalent to the shift of the samples spectrum by (k_x, k_y). We can make this assumption because of the Fourier Shift Theorem:

$$F(\omega) \cdot exp(-j\omega a) = F[f(t-a)] \tag{1}$$

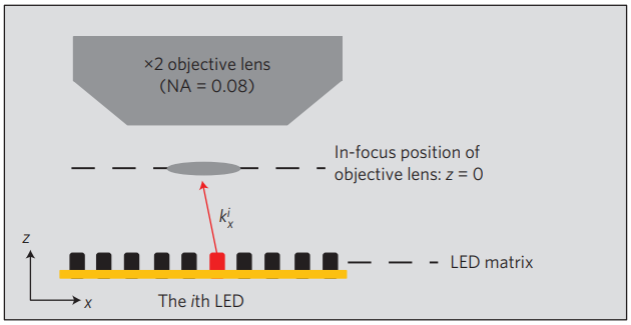

Figure 1: Images are taken of a sample from angle-varied LEDs. Those taken from larger angles have longer exposure times to account for less light hitting the sensor. Figure from (1).

Understanding this assumption, we can image a sample from various illumination angles which will each correspond to a unique shift in the samples spectrum. These shifted spectra then pass through a microscope objective which is equivalent to multiplying by the circular filter function with radius $NA \cdot 2\pi/\lambda$. Capturing these images, we are left with low-resolution views of the sample with circular spectra at different locations in the frequency domain. From here, FPM consists of the following steps:

1. Create a high-resolution estimate of the samples spectrum by up-sampling an acquired image and taking a Fourier Transform.
2. Choose a shifted circular subregion of the estimate and take an inverse Fourier Transform to get a low-resolution image estimate. This will correspond to an acquired image.
3. Use the true acquired image as a constraint by replacing the estimate's intensity values.
4. Fourier transform the updated estimate and replace the values in the high-resolution spectrum estimate.
5. Repeat steps 1-4 for all acquired images.
6. Inverse Fourier Transform the full final high-resolution spectrum estimate to get a high-resolution image estimate of the sample.

This is the core idea of how FPM is able to reconstruct a high-resolution image of some thin sample via ptychography ('stitching together') in the frequency domain. Adding to this core algorithm, we can recover additional information beyond the intensity and phase of the sample by adding steps to update additional functions. For example, the microscope objective doesn't correspond to a perfectly binary circular filter. In reality, this pupil function has some aberration (4). We can make an initial estimate of the pupil as a binary circular filter and then update it to both recover the true objective aberration and improve our estimate of the sample. This is relatively easy to implement, so we have added it to the project.

2.2 Hardware

UC2 is a relatively recent project which aims to standardize modular optical systems. The UC2 GitHub repository contains STL files for printing out many different 50mm modules which can house various optical components. For this project, we use modules for the z-stage to fit the sample and the microscope objective along with a module to hold the LED array in place. A custom unit needed to be designed for the High Quality Raspberry Pi Camera as previous version were meant from the v1 and v2 cameras. Modules are placed on a base plate and connected with 5mm ball magnets so they can easily be rearranged if adjustments to the system are needed.

We use a 4x/0.13 objective whose back was placed 134mm from the image sensor. The front of the objective was place 78mm from the z-stage followed by an additional 132mm to an AdaFruit NeoPixel 8x8 LED array. A custom pinhole covering for the LED array was printed to ensure each of our sources emits coherent light. The components were printed in black in attempt to avoid unnecessary reflection of stray

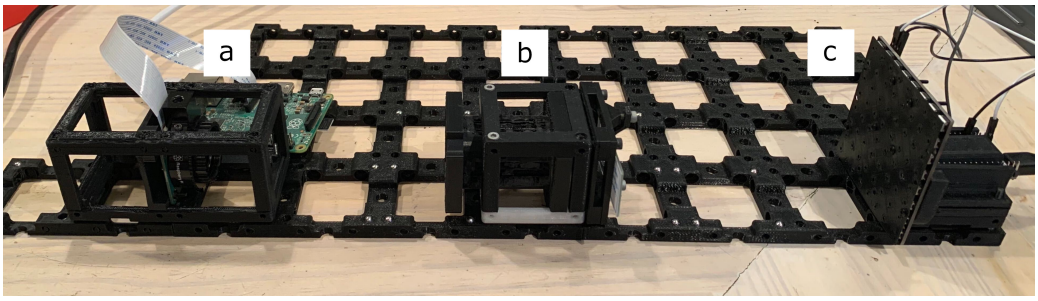

Figure 2: Image of the setup. a) HQ RPI Camera, b) Microscope objective with z-stage, c) LED array with custom pinhole covering.

light which might cause artefacts in the reconstruction but all images are taken in a dark room to compensate for a lack of shielding on the components.

Initially, the distances of each of the key elements was much shorter and we used an additional lens to infinity-correct the system. This led to a very low magnification of 0.28 however. This low magnification gave the system a sample factor of 0.38 when a paper analyzing this metric calculates a minimum of 2 and a suggestion of 3. By removing the additional lens and spreading the components out, we increase the magnification to 1.5 which improved the sample factor to 2.29. In total, the cost of these components is around $300. Without the access to a 3D printer, we ordered many of our parts from a third party, so this price will decrease if components are printed personally.

2.3 Software

We implement a version of FPM which includes Embedded Pupil Recovery (EPRY) (4) so that we can recover information about the aberration within our system and improve the overall reconstruction. Before having access to our own data, we tested the algorithm on blood smear data provided by Dr. Josh Brake from a previous FPM project. The images were relatively small at only 128x128px which was very useful for quickly running and debugging the Python and MATLAB versions of the code. However, the provided data did not include some of the limitations that our physical system would impose on our acquired images.

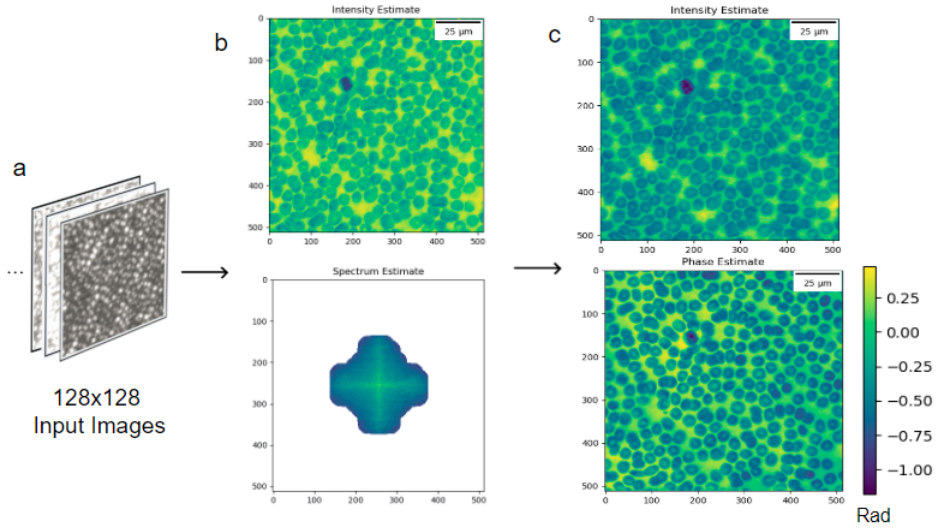

Figure 3: a) 64 Low-resolution acquired images. b) Iteration of FPM showing the current high-resolution estimate along with its estimated spectrum. c) Recovered high-resolution image along with its phase.

There are three primary limitations which require image processing prior to running the algorithm: demosaicing, denoising, and image segmentation.

Demosaicing: The HQ Raspberry Pi Camera comes included with a bayer filter (BGGR specifically). This essentially designates each pixel a specific color channel (red, green, or blue) as opposed to having each pixel be three channels. Because we take our images under red light only (using multiple wavelengths renders image / phase recovery extremely difficult) we are left with a very sparse red-channel image.

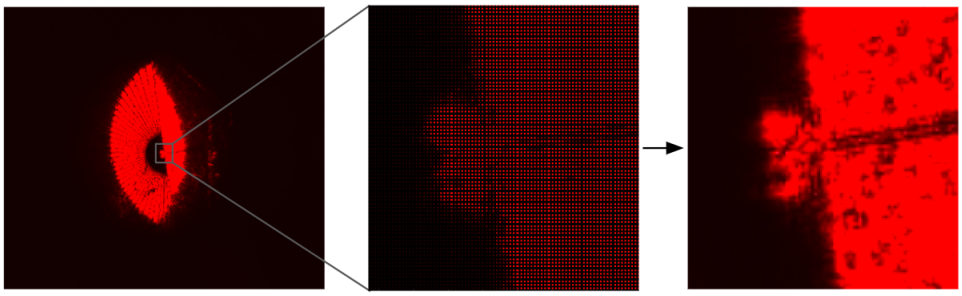

Figure 4: An acquired image (left) of a transparency sample can be seen in MATLAB as sparse (middle). Using the built in *demosaic* function, we can interpolate the missing data in the image (right).

Denoising: We try to minimize external light passing through the sample by placing a mask on the objective and taking images at night but inevitably, some noise will end up on the sensor. Denoising had already been implemented in the open-source MATLAB code provided by Tian Labs so all this required was adjusting some of the parameters. Noise is estimated for each acquired images by looking at multiple subregions of the image which correspond to the background and averaging the values in these regions together. This average value is then compared to some threshold to see whether this is truly noise or an important part of the image. In the case that it is determined to be noise, we subtract this value from each pixel to essentially cancel out the background of the image.

Image Segmentation: Running simulations on EPRY-FPM is not computationally intensive as the input images are simulated as relatively small for efficiency (128x128). In practice, the HQ Raspberry Pi Camera captures images of size 4056x3040. 64 images of this size from each of the LEDs on the array makes running the algorithm highly inefficient and requires that we optimize. To do this, each image is segmented into n by n subregions which the algorithm runs on separately. Each image is then stitched back together to recover the full image. This improves the speed of the algorithm but suffers from imperfect recovery as the stitches between each subregion can be highly visible.

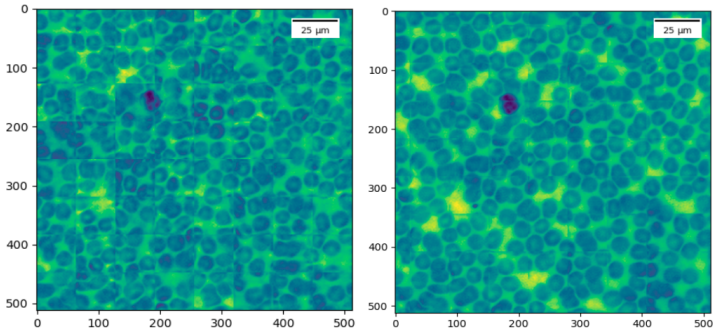

Figure 5: Here the seam can be seen in the unpadded reconstruction (left) and is less visible in the padded reconstruction (right).

To solve this, we pad each of the subregions by some amount such that they overlap with the adjacent subregions. This decreases the contrast between the reconstructions and preserves the improved recovery time. The amount of padding is restricted by how small the subregions are but we've found that around 10-20 pixels of padding is the most useful.

Image segmentation also allows for the recovery of the spatially varying pupil functions which is very useful since the aberration may be different from various angles. This means we get both a faster and higher quality reconstruction after a sufficient amount of padding is included.

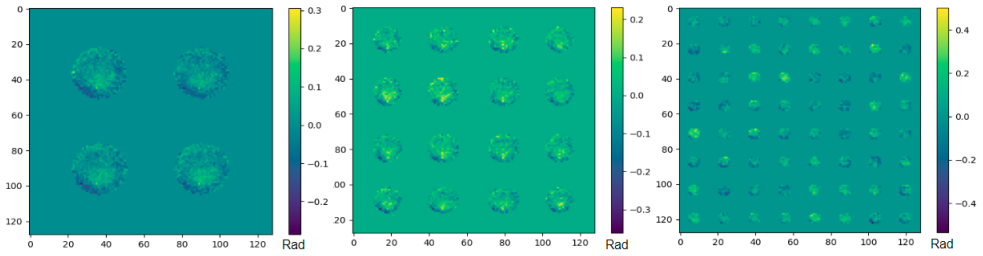

Figure 6: Recovered pupils for different amounts of segmentation. More specific pupils are recovered as the image is segmented more.

3 Results

Many of the acquired images from LEDs at a high angle with the sample were very dark. With low contrast, this caused us to lose the information encoded in some of these "darkfield" images. To increase this contrast and preserve the information in these images we do two things. First, we increase the exposure time on all darkfield images. Second, we use a High Dynamic Range (HDR) to capture the images. Without doing this, only a few LEDs from the center of the matrix would offer enough illumination to be useful for image reconstruction.

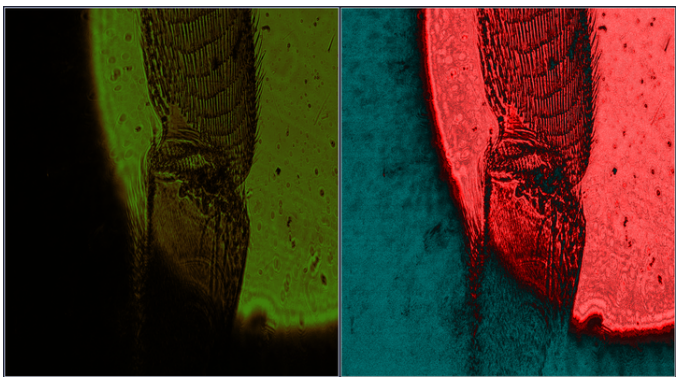

Figure 7: Acquired brightfield images of a honeybee leg. Left: SDR and Right: HDR.

From these HDR intensity images, we were able to recover both intensity and phase images of a honeybee leg. Comparing these reconstructions to the low-resolution acquired image, we can see how much detail was recovered as the resolution was increased by a factor of four. The phase image is also useful for noticing depth differences in the sample. For example, we can see that the center of the leg is relatively flat and there is an area which bulges out towards the top. This type of topographic analysis would not be available without the phase retrieval built into the FPM algorithm.

Future work on this project might include the implementation of other versions of Fourier Ptychography such as Fourier Ptychographic Tomography (7) which enables the imaging of thick samples, Multiplex FPM (8) which enables faster acquisition speeds, and Self-Learning FPM (9) which helps to eliminate unnecessary

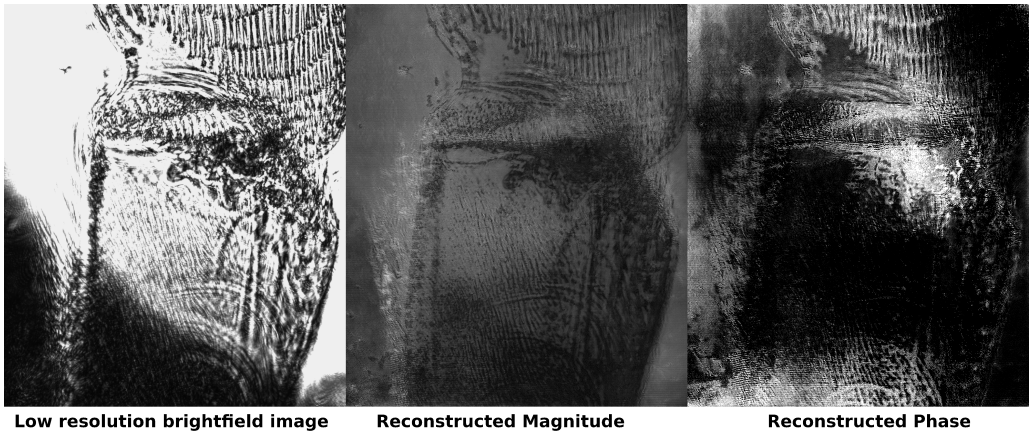

Figure 8: Left) Intensity of acquired image (2000x2000 px), Middle) Reconstructed image from FPM (8000x8000 px), and Right) Recovered phase information of image (8000x8000 px)

redundancy in input data. Adjusting the physical system to increase the sample factor beyond 2.28 would also be a great place to continue.

4 Reflection

Personally, this project has given me new experience in several areas. The most important of these for me was learning how to extract useful information from scientific articles. These can often overwhelm a reader with information, but breaking them down in detailed notes and practicing implementing some of the work they discussed went a long way in terms of helping finish my part of the project.

Specifically, I worked on software; implementing EPRY-FPM in Python, understanding and running data on Tian Labs MATLAB code, image processing, and analyzing efficiency of the algorithms. Prior to this summer I had very limited experience with Python and MATLAB and had never used GitHub before. These were all key components of our project so I got a great deal of practice with various libraries such as numpy, scipy, matplotlib, pillow, and others.

An important aspect of my learning was communication with my research advisor Dr. Josh Brake and my partner Alec Vercruysse. Multiple meetings each week were very useful for asking questions and organizing our ideas for the project. These discussions in addition to continuous reading of new articles throughout the summer were very helpful for keeping a good workflow. One issue to overcome was the online aspect of the research this summer. Motivation was sometimes difficult to maintain which everyone had to deal with in their own way. For me, maintaining a detailed checklist of sub-goals helped me break down our large project into specific daily work.

Something I wish I had focused more on was the theory behind a lot of the algorithms the team looked at. We worked mainly towards implementing them without comprehending the underlying mathematics which structured them. Having this sort of understanding would have helped greatly with debugging the code efficiently, however, this is something I plan on reading into in the near future.

Overall, this summer was a very positive learning experience which gave me a direction to look for more applied mathematics research. Computational imaging is extremely interesting, especially with how useful it has become on a macroscopic scale (10). I look forward to the possibility of working in other undergraduate labs such as the Functional Optical Imaging Laboratory (FOIL) at UT Austin as I continue to pursue a PhD.

References

[1] Zheng, G., Horstmeyer, R., & Yang, C. "Wide-field, high-resolution Fourier ptychographic microscopy." *Nature Photonics.* (2013).

[2] Diederich, B., Lachmann, R., Carlstedt, S., Marsikova, B., Wang, H., Uwurukundo, X., Mosig, A., & Heintzmann, R. "UC2 – A Versatile and Customizable low-cost 3D-printed Optical Open-Standard for microscopic imaging." (2020).

[3] Diederich, Benedict Lachmann, Renè Carlstedt, Swen Marsikova, Barbora Wang, Haoran Uwurukundu, Xavier Mosig, Alexander Heintzmann, Rainer. (2020). UC2 - A Versatile and Customizable low-cost 3D-printed Optical Open-Standard for microscopic imaging. 10.1101/2020.03.02.973073.

[4] Xiaoze Ou, Guoan Zheng, and Changhuei Yang, "Embedded pupil function recovery for Fourier ptychographic microscopy," Opt. Express 22, 4960-4972 (2014)

[5] Fienup, J.R. "Phase retrieval algorithms: a comparison." *Appl. Opt.* 21, 2758-2769 (1982).

[6] Aidukas, T., Eckert, R., Harvey, A.R. et al. Low-cost, sub-micron resolution, wide-field computational microscopy using opensource hardware. Sci Rep 9, 7457 (2019). ht

[7] Roarke Horstmeyer, Jaebum Chung, Xiaoze Ou, Guoan Zheng, and Changhuei Yang, "Diffraction tomography with Fourier ptychography," Optica 3, 827-835 (2016)

[8] Tian, L., Li, X., Ramchandran, K., & Waller, L. "Multiplexed coded illumination for Fourier Ptychography with an LED array microscope." *Biomed. Opt. Express.* 5, 2376-2389 (2014).

[9] Zhang, Y., Jiang, W., Tian, L., Waller, L., & Dai, Q. "Self-learning based Fourier ptychographic microscopy." *Opt. Express.* 23, 18471-18486 (2015).

[10] C. Goddi et al. 2019. "First M87 Event Horizon Telescope Results and the Role of ALMA," 177, Pp. 25-35.