Accelerating Reconstruction of Reflective Fourier Ptychographic Microscopy by Employing a Global Optimal Search Algorithm in a Graphics Processing Unit

V. H. Pham ©, B. H. Chon, and H. K. Ahn ©

Abstract—Reflective-type Fourier ptychographic microscopy, an outstanding solution for surface inspection, employs a phase retrieval algorithm to reconstruct a high-resolution image with a wide field-of-view. This technique is time-consuming because it requires processing a large number of low-resolution images coming from the dark-field illuminator and bright-field illuminator. To decrease the computation time, we propose a new Fourier ptychography recovery method running in parallel in a graphics processing unit. We also utilized an adaptive step-size incremental gradient descent searching algorithm. The approach was validated by measuring a USAF reflective resolution target and by inspecting the surface of a smartphone integrated circuit. The processing time was remarkably reduced to 88.29 s, about 11.1 times less than the state-of-the-art method, while obtaining about 250 nm half-pitch resolution and about 1 × 1 mm² field-of-view.

Index Terms—Microscopy, three-dimensional microscopy, other imaging technique.

I. INTRODUCTION

A COMPUTATIONAL method invented in 2013 [1] named Fourier ptychographic microscopy (FPM) has become increasingly popular due to its extendable space-bandwidth product (SBP). Since it was first introduced, many studies have been performed to further enhance FPM resolution, using hardware improvements [2], as well as algorithmic approaches [3] [4] [5]. In addition, the data acquisition speed and the reconstruction speed of FPM have been improved by using multiplexed illumination [6] [7], or a single shot mechanism [8].

Generally, there are two categories of Fourier ptychography (FP), the transmission and reflective types. Transparent organic specimens, such as human blood cells and animal cells that have low light absorption properties have been extensively measured using transmission FPM. Nevertheless, most industrial components in manufacturing processes have reflective properties and cannot be inspected by the first type. In such cases it is more effective to employ reflective FPM for the surface inspection of, for example, components made of ceramics, semiconductors, or metals. Because of difficulties with optical alignment and improving signal to noise ratio (SNR), only a few articles related to reflective FPM [9]–[11]. The state-of-the-art research on the reflective type was reported in 2019 [9], attaining a half-pitch resolution of 250 nm using a parabolic mirror for dark-field illumination.

In 2021, this research group found that segments are independent in the misalignment calculation of mcFPM [12] when using an objective with a small amount of aberration, which allows the calculated misalignment of a small segment to be propagated to the whole object. This method shortens the computational time for mcFPM. Although the mcFPM can reduce LED misalignment estimation time, the total recovery speed is still too slow to be used in industry. For reflective FPM to be directly applied in the industrial field it will be necessary to reduce the task time for image processing. Enhancing the computational hardware (such as using GPU) can be a solution when the GPU has more image processing power than the CPU.

The Central Processing Unit (CPU) contains several powerful cores and is designed to treat a wide variety of tasks and complex serial calculations with low speed and low efficiency, and this requires a lot of time. On the other hand, the Graphics Processing Unit (GPU), structured using a hundred of thousands of small cores, was developed for rapid image rendering and parallel instruction processing. Additionally, with the powerful ability of floating-point arithmetic, a GPU can accelerate the calculation time. For example, if deep learning training [13] or the machine learning training of Neural Networks were five times faster, image filtering and feature extraction from the image could be accelerated by forty times [14], Fast Fourier Transform (FFT) and the cross-correlation of signal processing are computed twenty-eight times more rapidly than a CPU [15].

Recently, Zhou G. [16] compared the use of a GPU and CPU on a single iteration of processing FPM while applying the online strategy process. However, the online framework might neglect the misalignment estimation of the LED array, which leads to
a somewhat poorer recovered image quality. Moreover, in this research, the author did not mention the effect of raw image pixel size on the GPU processing time, which strongly affects the FFT calculation speed. In addition, upgrading hardware by applying GPU acceleration is not a unique solution to reduce the task time of FP recovery. We propose a better method, which is to add the power of the GPU hardware to an optimal search strategy for FP recovery, where we can speed up the recovery of the object and pupil functions.

Gradient searching strategies such as PIE [17] and EPRY [18], have been popular for transmission FPM, offering considerable flexibility and estimating efficiency. However, if there is significant noise contamination in the raw images, their steadiness and reconstruction quality can be remarkably degraded. The adaptive step-size (AS) method [19] was found to overcome the degradation of the reconstruction image caused by noise, and also to increase the stability of FP recovery. Moreover, the AS method has emerged as the smartest update strategy method, since it can quickly determine the optimal solution. The inventor [19] mentioned that AS reduces processing times by three, because it only needs ten iterations to converge to an optimal solution, compared to thirty iterations with other methods.

In this study, using AS FPM and GPU acceleration, we propose a framework for reconstructing FPMs. Firstly, we propose a time saving method by parallelizing the FP recovery kernel functions on the GPU. Secondly, the effective of AS FPM is proven not only by rapid convergence but also by quickly stitching HR sub-images thanks to its global minima. Additionally, this paper analyzes the effect of a number of sub-images and the size of a sub-image on FP recovery times in cooperation of CPU and GPU. Finally, experiments verify the effectiveness of the proposed method, and it is demonstrated that the framework saves a lot of time.

Here we present a two-pronged approach to reduce the computational cost in reflective FPM while achieving a high resolution (HR) reconstructed image with a full FOV. Section II.A describes the configuration of the reflective FPM with bright-field and dark-field illumination. In Section II.B, we specify that the AS method can save a great deal of time by rapidly converging the FP recovery optimization problem, and eliminating the discontinuity between adjacent sub-images when stitching the reconstructed sub-images. And then, Section II.C analyzes the sensitivity of GPU speed to direct 2D FFT and inverse 2D FFT problems in the reflective FPM process. Next, Section III.B and Section III.C show the feasibility of reflective FPM applying and AS GPU by using an experiment measuring a reflective USAF sample and an IC circuit board. Finally, Section IV discusses the achievement of the proposed acceleration methods.

II. METHOD

A. Reflective FPM Mechanism

The construction of our reflective FPM includes two illuminators, an objective lens, and a CMOS camera as shown in Fig. 1. The bright-field illuminator (BI) consists of three LED rings, and a dark-field illuminator (DI) with three bigger LED rings. By using 4f imaging, the LEDs in the BI are imaged on the back focal plane of the objective lens. The DI with a parabolic mirror enhances the SNR of the reflective FPM. The beams coming from the three big LED rings in the DI are reflected from the parabolic mirror and then illuminated on the sample’s surface.

We applied a ring-type arrangement of LEDs to the BI and DI, which guarantees better performance than a rectangular grid-type [2]. To satisfy the convergence of the phase retrieval process in the FPM method, the position of LEDs in the BI and DI has to be designed to satisfy the 40% spectral overlapping condition in the Fourier domain [9]. The design of the reflective FPM with these boundary conditions is shown in Table II. The maximum achievable NA_{synth} is 1.06.

Fig. 2 shows how the reflective FP recovery framework is processed. Overall, there are four processing steps in this procedure, starting with CPU segmenting process, GPU misalignment estimation, GPU sub-images FP recovery, and ending with the CPU stitching process. In the first stage of the FP recovery procedure, all low resolution (LR) images are segmented into a number of sub-images calculated in Section III.B. The
Fig. 2. Sub-image reflective FP recovery framework using a GPU and adaptive step-size incremental gradient descent.

TABLE I

| Given: | Optimization Problem of Fourier Ptychography Microscopy |
|--------|----------------------------------------------------------|
| \( I_m \): \text{Diffraction pattern of } m\text{th LED} | | |
| \( I_{x,m} \): \text{Diffraction pattern of } x\text{th } m\text{th segment} | | |
| \( \Delta x_B, \Delta y_B, \Delta h_B \): \text{Estimated bright-field misalignment} | | |
| \( \Delta x_D, \Delta y_D, h_D \): \text{Estimated dark-field misalignment errors} | | |
| \( f_{x,B}, f_{y,B} \): \text{the } x,y\text{th LED position} | | |
| \( f_p \): \text{parabollic mirror focal length} | | |
| \( O_{x}(k): \text{Random initial object function of the } k\text{th segment} \) | | |
| \( P_{x}(k): \text{Random initial pupil function of the } k\text{th segment} \) | | |
| \( k \): \text{central wavelength of LED} | | |

Find:

| \( O_{x}(k): \text{recovered object function of the } k\text{th segment} \) | | |
| \( P_{x}(k): \text{recovered pupil function of the } k\text{th segment} \) | | |

Satisfy:

\[
\begin{align*}
 k &= \frac{k_{x,m}^2 + k_{y,m}^2}{k_{x,m}^2} \\
 f_{x,m}^2 &\geq \frac{\Delta A_{B}}{\lambda} \\
 f_{y,m}^2 &= \frac{\Delta A_{B}}{\lambda} \\
 k_{x,m} &= \frac{\Delta A_{B}}{\lambda} \cos(\delta_{Bx}) \sin(\delta_{By}) \Delta x_B \Delta y_B \\
 k_{y,m} &= \frac{\Delta A_{B}}{\lambda} \sin(\delta_{Bx}) \cos(\delta_{By}) \Delta x_B \Delta y_B \\
 x_{p,m} &= \frac{\Delta A_{B}}{\lambda} \sin(\delta_{By}) \cos(\delta_{Bx}) \Delta x_B \Delta y_B \\
 y_{p,m} &= \frac{\Delta A_{B}}{\lambda} \cos(\delta_{By}) \sin(\delta_{Bx}) \Delta x_B \Delta y_B \\
 \gamma &= \frac{\Delta A_{B}}{\lambda} \delta_{Bx} \Delta x_B \Delta y_B \\
 \beta &= \frac{\Delta A_{B}}{\lambda} \delta_{By} \Delta x_B \Delta y_B \\
 \end{align*}
\]

Minimize:

\[
\begin{align*}
 e_i = \sum_{i} \sum_{j} |F_{x}(k) - F^{-1}(P_{x}(k)O_{x}(k - k_{ij})) | \\
\end{align*}
\]

The core computations in FP recovery are the computation of 2D-FFT, inverse 2D-FFT, and the phase retrieval process. These operations are very sensitive to the size of the raw image and the number of raw images acquired from the consecutively illuminating LEDs in the BI and DI. To minimize the computational cost of FPM while processing a full FOV image, it is necessary to segment the raw image into many sub-images and recover strongly affect the final recovered HR image. Once the misaligned values are allocated, all sub-images are applied to the smartest FP recovery, which is called adaptive step-size incremental gradient descent searching. This procedure utilizes the GPU to rapidly find the converged object and pupil functions of the FP recovery problem. Finally, the stitching procedure which combines the recovered HR sub-images to a large HR image is accomplished on the CPU, due to the limited capacity of the GPU.

B. Segmenting and Employing Adaptive Step Size in the FP Recovery Process

The design parameter of the reflective Fourier Ptychographic Microscopy

| Parameters | Name | Values |
|------------|------|--------|
| Camera     | Model | Basler acA 4112 - 20 µm |
|            | Pixel size | 3.45 µm |
| Objective Lens | Model | Mitutoyo M Plan Apo |
|            | Numerical Aperture | 0.28 |
|            | Magnification | 10x |
|            | Number of rings | 3 |
| Bright-field Illuminator | Number of rings | 3 |
|            | LED No. in each ring | 1, 5, 10, 0, 5.04, 10.08 mm |
|            | Ring’s radius | 52.5, 72, 96 mm |
| Dark-field Illuminator | Number of rings | 3 |
|            | LED No. in each ring | 18, 24, 30 |
|            | Ring’s radius | 52.5, 72, 96 mm |

| Parameters | Name | Values |
|------------|------|--------|
| Parabolic mirror | Focal length | 100 mm |
them simultaneously. And then, the recovered HR segments are stitched together to obtain the full FOV.

Another reason for segmenting the raw image and reconstructing the sub-images is the prerequisite of estimating a spherical wave illuminated from the LED into a plane wave. The plane wave ensures that every point in the sample is illuminated with a beam that has the same k-vector, which can be approximated when we partition the raw image into many sub-images [14]. The optimization problem of FP recovery of the individual segments is shown in Table I.

The incremental gradient descent can solve the optimization of FP recovery by iteratively updating the object and pupil functions until they converge. The gradient of the minimizing error related to the object and pupil functions is calculated in every iteration. Then, the updated object function and pupil function in the th iteration of each th segment is presented below:

\[
\Psi_{j,s}^{i+1} = \alpha_{j,s} \sqrt{\frac{\Psi_{j,s}^i \big| P_{j,s}^i \big|_{\text{max}}}{\big| P_{j,s}^i \big|^2 + \gamma}} \times \Psi_{j,s}^i - \alpha_{j,s} \bigg| O_{j,s}^i \bigg|_{\text{max}} \bigg| O_{j,s}^i \bigg|^2 + \gamma \bigg] \frac{O_{j,s}^i}{P_{j,s}^i} \bigg|_{\text{max}} \bigg| P_{j,s}^i \bigg|_{\text{max}} \bigg| \bigg| P_{j,s}^i \bigg|^2 + \gamma \bigg]
\]

(1)

(2)

(3)

(4)

In addition, AS selection can rapidly converge to an optimal solution. That is a very useful property for reducing the processing time of FP recovery when the number of input raw images increases. The author [19] observed that AS requires only ten iterations to obtain a good HR image, while other strategies require more than thirty iterations. (this sentence is repeated.)

C. GPU Acceleration and Stitching

The GPU acceleration is performed using a CUDA programming model, in which the CPU acts as a host and the GPU functions as a computing client. The serial arithmetic is initialized by the CPU and then transacted to the GPU where the speedy parallel threads computing occurs. The host space memory, an independent memory address of the CPU, calls CUDA memory management functions such as initializing, opening, releasing the memory space of variables functions, host and client addresses for data transmission functions. The kernel functions deal with large-scale data by computing parallelly in GPU CUDA. The CUDA programming model conveys all kernel functions computing serially in the CPU to parallel processing in the GPU, helping to save a huge amount of time.

For several years, support for running MATLAB on GPUs has been built in, with improved progress with each version. MATLAB contains a GPU coder toolbox which supports compiling MATLAB functions into ‘.mex’ files. Mex files consist of kernel functions, written in C++, which accept a call from MATLAB and process on the GPU in parallel. In the FP recovery problem, the bright-field and dark-field misalignment estimating function, and the FP recovery of sub-images are all converted to ‘.mex’ files and accelerated in the GPU to save time.

The misalignment estimation and recovery process in FPM mainly deals with Fourier Transform (FT), which is time-consuming to compute. For an image with a size of \( L = M \times N \) pixels, computing the spectral 2D Fourier Transform by Discrete FT algorithm is shown in (5) for a CPU. The DFT method requires O(L2) complexity for calculation. In 1965, Cooley-Tukey [21] conducted a divide-and-conquer algorithm to minimize the complexity FFT computation to O(L log L). Since then, many other algorithms have been invented, but the computational cost still remains when running on a CPU. With the help of parallel processing in the GPU, the time required for 2D FFT computing is largely decreased, even though the data size is big.

\[
\mathcal{F} (k_x, k_y) = \frac{1}{M \cdot N} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f (x, y) e^{-j2\pi \left( \frac{x \cdot k_x + y \cdot k_y}{M \cdot N} \right)}
\]

(5)

III. EXPERIMENT

A. Experimental Setup

The experimental setup of the reflective FPM is introduced in Section II.A. All of the equipment’s components and their
parameters are listed in detail in Table II. The parabolic mirror’s diameter must be sufficient to cover the designed NA_illum, which is calculated from the LEDs illumination angle.

B. The First Case Study: The USAF Target

The FP recovery process introduced in Section II.B was validated by examining the USAF target. To perform the FP recovery computation, MATLAB R2021a was utilized on a Windows 10 Enterprise Edition operation system (Intel Core i9–9900KF CPU@3.6 GHz). The system was equipped with an NVIDIA GeForce RTX 3080 Ti GPU and installed NVIDIA CUDA runtime 11.6 version.

Fig. 3 shows the real reflective FPM setup with a design of circular dark-field illumination and a parabolic mirror. At first, after sequentially illuminating the LEDs and capturing 88 images, the misalignment calibration of the BI and DI was conducted using the mcFPM algorithm, which has the segment-independent property in our FPM system. The mcFPM was applied to a sub-image with a 190 \times 190 pixels size, cropped from the raw image illuminated on the on-axis LED. The BI misalignment error was estimated first and then the results were used for estimating DI dislocation. The GPU took 27 s to complete the two processes (11.6 s of BI, 15.4 s of DI), about 2.7 times faster than the CPU which needed 72.9 s to finish them (34.3 s of BI, 38.6 s of DI).

The reconstructed HR image is reconstructed by three steps after DF and BF calibration, as shown in Fig. 2. Firstly, CPU slices LR image into sub-images and transfers them sequentially to GPU. Secondly, GPU processes the FP recovery kernel functions in parallelly in order to reconstruct a sub-region HR image. Finally, the recovered subregion HR image is sent back to the CPU to be stitched together to create the FOV HR image. Two major time consumers are computing FP recovery of sub-images in GPU and transferring images between CPU and GPU.

As discussed in Section II.C, the FP recovery is very sensitive to sub-image pixel size because it involves a Fourier Transform computation. Fig. 4(a) shows the FP recovery calculation time related to the pixel size (N \times N pixels) of the sub-image when running on the GPU. To calculate FFT and IFFT of FP recovery, the CUDA Fast Fourier Transform (cuFFT) in the GPU highly optimized for input size that can be written in the form $N = 2^a \times 3^b \times 5^c \times 7^d$ [22]. In general, the smaller the prime number, the faster the speed. Therefore, the GPU computing time is usually very long when N is an odd number and is reduced a lot when N is an even number. Choosing an optimal pixel size of the sub-image will save a great amount of time. When the size of the sub-image was 250 \times 250 pixels, it took only 110.4 s to recover 12 \times 12 of the sub-images. On the other hand, it took 309 s to recover the same number of sub-images when the size of the sub-image was 257 \times 257 pixels.

Fig. 4(b) indicates how the number of segments affects GPU processing time. For segmenting, FP recovery, and stitching processes, the optimal segmentation of the raw image was 3 \times 3 sub-images (1000 \times 1000 pixels). The GPU only took 61.29 s with 3 \times 3 sub-images (1000 \times 1000 pixels), while it needed 160.78 s in the case of 16 \times 16 sub-images (184 \times 184 pixels). The computational time dominates when the number of segments is less than nine sub-images (3 \times 3). When the number of segments exceeds nine, the processing time is increased as the number of segments increases since the transferring time becomes dominant. Lastly, the optimal number of segments...
Fig. 5. (a) High-resolution image using the state-of-the-art method, (b) High-resolution image of the proposed method, (c) Amplitude of the HR image with the proposed method, zoomed in on elements 9 and 10.

is 9, which makes the summation of FP computing time and transferring time the shortest.

The power of the AS application in the sub-image FP recovery can be seen in the recovered full FOV HR amplitude and phase images. Thanks to the global convergence property of the AS, these images were completely continuous without the average process of the overlapping areas in the sub-images. Fig. 5(b) shows the continuity in HR amplitude of the full FOV USAF target when using the proposed approach, compared with the discontinuity of the one in Fig. 5(a) when using state-of-the-art methods. We observed that the overlapping area required in this approach could be reduced to 4%, compared to 10% used in the online strategy method [16]. Minimizing the overlapping area relaxes the complexity of calculation in the GPU and makes it run faster. The pattern in group 11 and element 1 of USAF target is resolved in the Fig. 5(c), which is consistent with the theoretical solution to this system.

In addition, the adaptive step-size also converged very fast so that the FP recovery saved a lot of time. As mentioned above, the FP recovery process required only 61.29 s, compared to 917.19 s in the result in the previous study [12] to get a full FOV HR image (8095 × 8095 pixels). When enlarging the amplitude and the phase of the HR images, the image quality of this approach was much better than the quality of the recovered HR image using the online strategy method [16]. The online strategy method did not estimate misalignment error well, leading to the poor HR recovery image.

C. The Second Case study: Integrated Circuit (IC) FPM Experiment

An IC of a smartphone device was put into the reflective FPM for surface inspection. The amplitude of the full FOV HR image, shown in Fig. 6(a), had many more improvements compared to that of the 100 × objective lens, with about 1 × 1 mm² FOV. When the AS method was applied to the FP recovery of sub-images, and the recovered sub-images were stitched together, Fig. 6(a) did not contain any discontinuities.

The speed of FP recovery was remarkably improved by applying the AS method and GPU acceleration. To achieve an 8095 × 8095 pixels HR image corresponding to about 1 × 1 mm², the proposed method required 88.29 s in total, 27 s for misalignment estimation and 61.29 s for FP recovery. This is about an 11.1 times reduction compared to the method proposed by H. Lee et al., which took 980.27 s [12]. This method is capable of taking less time by applying multiple GPUs instead of a single GPU, as in this experiment.
IV. CONCLUSION

To achieve a full FOV and HR image while working with the bulk data of LR images, FPM still required a considerable amount of time, which is a noticeable problem that has to be overcome. Using the state-of-the-art method, the FPM system was able to achieve half pitch resolution less than 250nm in 980.27 s, or about 16 min with a full FOV of approximately 1mm2.

In this paper, we introduced a novel method, which accelerates the full FOV HR image reconstruction. This was accomplished by combining GPU computation and a smart optimization strategy which uses the incremental gradient descent with an AS. This approach is very effective when segmenting a large raw image into many sub-images, and when individually applying the FP recovery process to the sub-images. On one hand, the GPU has the power of parallel computing, which is especially productive to FFT calculations in FP recovery. On the other hand, the AS method ensures global convergence will be achieved in the FP optimization problem, which provides the continuous property to the stitched HR image. We found that there is the optimal segmented number and pixel size which result in the minimum processing time. We concluded that this is because of two factors affect the full FOV HR image recovery time: one is transferring and re-transferring time between CPU and GPU, and the other is sub-image processing time in GPU.

In the USAF target experiment, patterns could be resolved in Element 1 in Group 11, corresponding to a full pitch resolution of 488 nm. Recently, our approach took only 88.29 seconds to obtain a full FOV high resolution image of about 1 × 1 mm2 (8095 × 8095), which is the fastest task time for reflective FPM. Of course, the reconstruction time of more than 60 s is still not a short time to measure live cells which require real-time reconstruction. Instead, our method can be a useful tool for surface inspection of semiconductors and displays in the industry, which depends on scanning electron microscope (SEM) or atomic force microscope (AFM) taking more than several hours to obtain the 1 × 1 mm2 image.

REFERENCES

[1] 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.
[2] K. Guo, S. Dong, P. Nanda, and G. Zheng, “Optimization of sampling pattern and the design of Fourier ptychographic illuminator,” Opt. Exp., vol. 23, no. 5, pp. 6171–6180, Mar. 2015, doi: 10.1364/oe.23.06171.
[3] J. Sun, Q. Chen, Y. Zhang, and C. Zuo, “Efficient positional misalignment correction method for Fourier ptychographic microscopy,” Biomed. Opt. Exp., vol. 7, no. 4, pp. 1336–1350, Apr. 2016, doi: 10.1364/bioe.7.001336.
[4] S. Dong, Z. Bian, R. Shiradkar, and G. Zheng, “Sparsely sampled Fourier ptychography,” Opt. Exp., vol. 22, no. 5, pp. 5455–5464, Mar. 2014, doi: 10.1364/oe.22.005455.
[5] A. Zhou, W. Wang, N. Chen, E. Y. Lam, B. Lee, and G. Situ, “Fast and robust misalignment correction of Fourier ptychographic microscopy for full field of view reconstruction,” Opt. Exp., vol. 26, no. 18, pp. 23 661–23 674, Sep. 2018, doi: 10.1364/oe.26.023661.
[6] C.-K. Liang, T.-H. Lin, B.-Y. Wong, C. Liu, and H. H. Chen, “Programmable aperture photography: Multiplexed light field acquisition,” ACM Trans. Graph., vol. 27, no. 3, pp. 1–8, Aug. 2008, doi: 10.1145/1360612.1360684.
[7] M. Kellman, E. Bostan, M. Chen, and L. Waller, “Data-driven design for Fourier ptychographic microscopy,” in Proc. IEEE Int. Conf. Comput. Photography, 2019, pp. 1–8, doi: 10.1109/ICCPHOT.2019.8747339.
[8] B. Lee et al., “Single-shot phase retrieval via Fourier ptychographic microscopy,” Optica, vol. 5, no. 8, pp. 976–983, Aug. 2018, doi: 10.1117/1.optica.5.000976.
[9] H. Lee, B. H. Chon, and H. K. Ahn, “Reflective Fourier ptychographic microscopy using a parabolic mirror,” Opt. Exp., vol. 27, no. 23, pp. 34 382–34 391, Nov. 2019, doi: 10.1364/oe.27.034382.
[10] K. Guo, S. Dong, and G. Zheng, “Fourier ptychography for brightfield, phase, darkfield, reflective, multi-slice, and fluorescence imaging,” IEEE J. Sel. Topics Quantum Electron., vol. 22, no. 4, pp. 77–88, Jul./Aug. 2016, doi: 10.1109/JSTQE.2015.2504514.
[11] S. Pacheco, G. Zheng, and R. Liang, “Reflective Fourier ptychography,” J. Biomed. Opt., vol. 21, no. 2, 2016, Art. no. 26010, doi: 10.1117/1.jbo.21.2.026010.
[12] H. Lee, B. H. Chon, and H. K. Ahn, “Rapid misalignment correction method in reflective Fourier ptychographic microscopy for full field of view reconstruction,” Opt. Lasers Eng., vol. 138, Mar. 2021, Art. no. 106418, doi: 10.1016/j.optlaseng.2020.106418.
[13] E. Baber and B. Diri, “Performance analysis and CPU vs GPU comparison for deeplearning,” in Proc. 6th Int. Conf. Control, Eng. Inf. Technol., Istanbul, Turkey, Oct. 2018, pp. 25–27.
[14] H. Bahri, F. Sayadi, R. Khemiri, M. Chouchene, and M. Atri, “Image feature extraction algorithm based on CUDA architecture: Case study GFD and GCFD,” IET Comput. Digit. Techn., vol. 11, no. 4, pp. 125–132, Jul. 2017, doi: 10.1049/iet-cdt.2016.0135.
[15] C. Wang, S. Chandrasekaran, and B. Chapman, “cusFFT: A high-performance sparse fast Fourier transform algorithm on GPUs,” in Proc. IEEE Int. Parallel Distrib. Process. Symp., 2016, pp. 963–972, doi: 10.1109/IPDPS.2016.95.
[16] G. Zhou, S. Zhang, Y. Hu, L. Cao, Y. Huang, and Q. Hao, “Parallel Fourier ptychography reconstruction,” 2022, arXiv:2203.02507.
[17] 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.
[18] X. Ou, G. Zheng, and C. Yang, “Embedded pupil function recovery for Fourier ptychographic microscopy,” Opt. Exp., vol. 22, no. 5, pp. 4960–4972, Mar. 2014, doi: 10.1364/oe.22.004960.
[19] C. Zuo, J. Sun, and Q. Chen, “Adaptive step-size strategy for noise-reducing Fourier ptychographic microscopy,” Opt. Exp., vol. 24, no. 18, pp. 20724–20744, Sep. 2016, doi: 10.1364/oe.24.020724.
[20] J. M. Rodenburg, A. C. Hurst, and A. G. Cullis, “Transmission microscopy without lenses for objects of unlimited size,” Ultramicroscopy, vol. 107, no. 2/3, pp. 227–231, Feb. 2007, doi: 10.1016/j.ultramic.2006.07.007.
[21] J. W. Cooley and J. W. Tukey, “An algorithm for the machine calculation of complex Fourier series,” Math. Comput., vol. 19, pp. 297–301, May 1965, doi: 10.1090/s0025-5718-1965-0178586-1.
[22] “CUCFFT library user’s guide,” Vers. 11.7, NVIDIA Corp., Santa Clara, CA, USA, May 2022.