Quantifying Defocus: Controlling for Unintended Artifacts during TEM Image Analysis

Redford Hudson, Justin Mulvey, Joe Patterson

Introduction

Transmission electron microscopy is the ideal technique for studying nanomaterial. However, due to the physical properties of electronic radiation, artifacts are introduced to samples that obfuscate elements of the sample and make feature analysis much more difficult. Here, I propose a procedure for quantifying these artifacts and controlling for their effects to improve the performance of feature analysis.

Background

Transmission Electron Microscopy (TEM)[1] is a high-powered imaging methodology in which an electron beam is sent through a sample and received at a sensor. Electrons are blocked where there is material in the sample, and the signal composition detected by the sensor captures the structure of the sample. TEM yields much higher resolution images than light microscopy, and is therefore very popular among materials scientists[2]. Since electrons behave as waves, they undergo **constructive and destructive interference** in which perpendicular phases either amplify or neutralize each other. Because of this, some electrons are not detected at the sensor, meaning information is lost, artifacts are introduced, and the image produced is an imperfect replica of the sample.

Electron signals are lost according to their respective frequencies. The frequencies which are susceptible to wave interference are represented in the **Contrast Transfer Function (CTF)**[3] (Figure 1), which is a curve describing the relationship of wave interference and frequency. The CTF of an image—and thus the type of information lost—changes with the **defocus** value of the lens used to focus electrons onto the sensor. Researchers can adjust the defocus of their microscope to capture features of variable scales and sizes, and it is usually the case that researchers collect datasets using multiple values of defocus. This is a problem when using image processing algorithms to analyze features, because the properties of the artifacts change across samples. Additionally, since the defocus value recorded by the microscope is oftentimes inaccurate due to high sensitivities and unpredictable lab conditions, the shifting properties of these artifacts are difficult to measure.

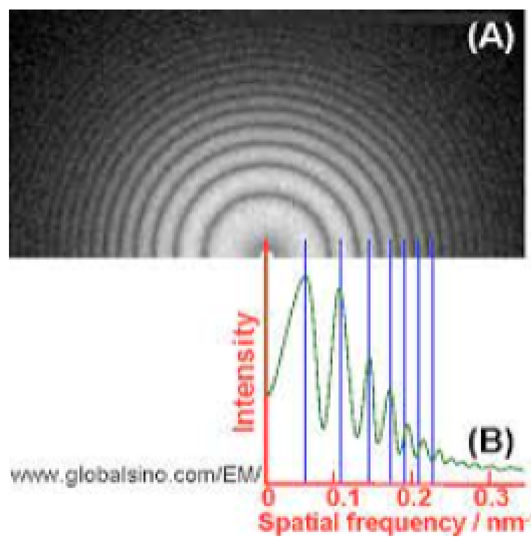

Figure 1. Contrast Transfer Function illustrating the intensity of signals according to frequencies.

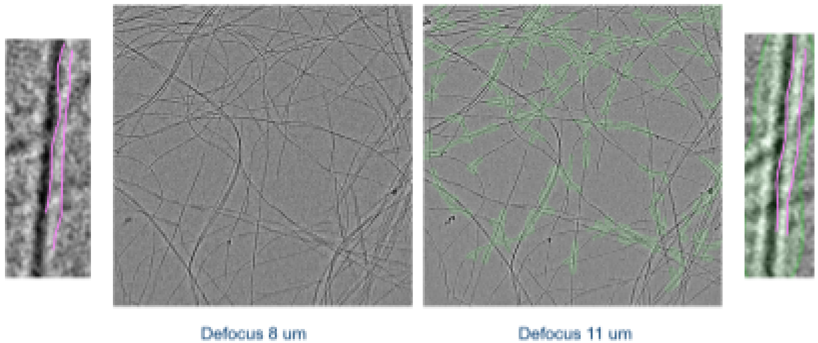

Figure 2. The defocus fringe generated in two contexts at different defocus values.

Figure 2 shows one such artifact, called the **defocus fringe**—a region of contrast surrounding features. The brightness of the defocus fringe varies according to defocus, and segmentation algorithms cannot consistently differentiate between features.

I claim that there is a way to quantify frame-to-frame variations in the CTF, and furthermore resolve this issue demonstrated in Figure 2. The CTF of an image can be estimated using the **Fourier Transform** (Figure 3), a widely-used image science algorithm that translates an image into its compositional frequency components. By applying the fourier transform to images, image scientists can quantify many useful patterns, and can visualize the CTF for TEM images. Once the CTF is calculated, the defocus value can also be quantified. Once the defocus value is quantified, it can be fed into subsequent image processing and feature segmentation algorithms as input, enabling the obfuscation of meddling artifacts to be controlled for. Finally, once effects of artifacts are controlled for, reliable results can be obtained for further constructive research.

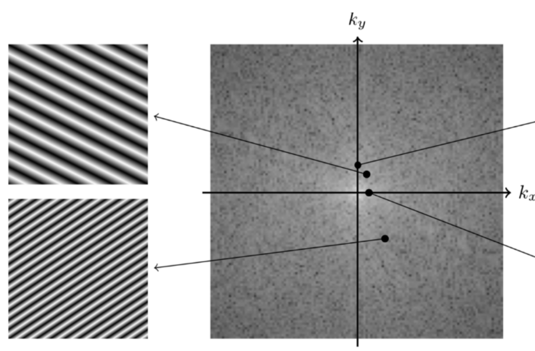

Figure 3. A generic Fourier Transform, with illustrations of patterns that are represented by various 2-dimensional values in the image.

Purpose

The purpose of my research project is to implement a general-purpose, data science-driven methodology for quantifying defocus artifacts to improve the performance of feature segmentation techniques. This will supplement research projects in the Patterson Lab that utilize TEM, and help my fellow researchers to extract insightful data.

Objective

The objective of my research project is to write a script that quantifies the defocus of an image and returns an FFT index to be used in other image processing scripts.

Approach

This research project concerns image and data science, and so all my studies will be done *in silico*.

I will write a script for quantifying the defocus value in **Python**, because: I am most familiar with this language; Python is the optimal language for data science, and there are many pre-existing Python packages available for public use that will streamline my development of the auxiliary functions. The primary Python packages I will be using are: **NumPy** (logic); **MatPlotLib** (graphing and data visualization); **Pandas** (data framing and management); **OpenCV**[4] (computer vision); and **SciPy**[5] (linear algebra and fourier operations).

The script will take in a **Medical Research Council (MRC)** image file provided by fellow researchers, and will return a scalar segmentation index. This high-level transformation will be accomplished by chaining together many low-level transformations involving unique systems of logic. Therefore I will break up the script into individual modules, each of which will perform a unique and necessary transformation.

To streamline the debugging process and give my script added flexibility, I will punctuate the signal chain with "junctures" in which the results of the transformation will be stored in a **Comma-Separated Values (CSV)** file. This will allow me to isolate certain portions of the signal chain rather than having to run the entire script, which would be very time-consuming.

I will run these functions on a dataset collected by the Patterson Lab, which consists of 1070 MRC images at unknown defocus values. To compress the run-time, I will run these scripts on the **High-Performance Cluster (HPC)**, a cloud computing infrastructure, operated and distributed by UCI's Research Cyberinfrastructure Center, that supports and catalyzes scalable computations. The HPC[3] runs on a pay-per-cycle model, and as such its usage will constitute the majority of my budget.

Once my algorithm quantifies the defocus values, this information will inform the existing segmentation algorithm being used by the Patterson Lab. Accounting for the defocus will result in more accurate image segmentation, and improve the accuracy of the analysis. All code will be made publicly available via the Patterson Lab Github page, and should be directly applicable to additional datasets in the TEM community.

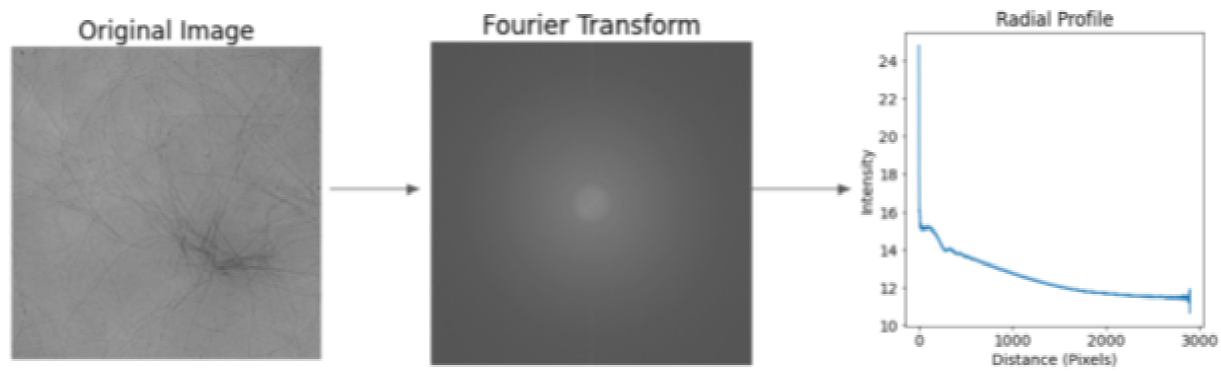

Figure 4. Preliminary results of Fourier Transform on raw data. Left) Raw data; Center) Fourier Transform of raw data; Right) radial integration of Fourier Transform

I have already implemented a rudimentary script that applies Fourier Transform and radial integration algorithms to MRC files. The results are shown in Figure 4.

Timeline

Timeframe	Tasks
Weeks 1-2	• Write script-A that renders MRC images in Reciprocal Space via the Fourier Transform ○ Debug

Weeks 3-4	• Write script-B that transforms reciprocal images into CTF's (as 1-dimensional arrays) via Radial Integration
	    ○ Debug
	• Merge script-A and script-B into script-C
	• Run script-C on entire dataset of 1070 images harvested by fellow materials science researcher using the HPC[3]
	    ○ Yield CSV-C (containing CTF's for 1070 images)

Weeks 5-6	• Write script-D that preprocesses and smooths the CTF via a Gaussian Blur
	    ○ Debug
	• Write script-E that quantifies the distance in pixels of the 2nd peak in the CTF via a Peak-Finding Algorithm (The distance and intensity of the 1st peak will most-likely be distorted by high-frequency components present in the fourier transform as a product of the fibers in the image, so referring to the 2nd peak is more reliable)
	    ○ Debug

	• Merge script-D and script-E into script-F • Run script-F on CSV-C using the HPC^3 ○ Yield CSV-F (containing peak distances for 1070 images)
Weeks 7-8	• Map peak distances from CSV-F to other properties of respective images (date, average intensity) to discover statistical correlations • Write script-G that transforms peak distance and other properties into a segmentation index via a one-to-one formula based off statistical correlations ○ Debug
Weeks 9-12	• Run script-G on CSV-F using the HPC^3 ○ Yield CSV-G (containing peak distances for 1070 images) • Run segmentation algorithm using values from CSV-G to quantify features • Compare the results with control data in which the defocus is known, and further optimize script-G • Once results are sufficiently accurate, transfer CSV-F to fellow researcher

Itemized Budget

Items	**Price**
HPC^3: 40k core hours	$400
HPC^3: 1TB of storage	$100
Visual Studio Professional 2022	$200
Computer Monitor	$300
Total	**$1000**

References

1. Williams, D. B. & Carter, C. B. *Transmission electron microscopy: a textbook for materials science*. (Springer, 2008).
2. Rizvi, A., Mulvey, J. T., Carpenter, B. P., Talosig, R. & Patterson, J. P. A Close Look at Molecular Self-Assembly with the Transmission Electron Microscope. *Chem. Rev.* **121**, 14232–14280 (2021).
3. Schramm, S. M., Pang, A. B., Altman, M. S. & Tromp, R. M. A Contrast Transfer Function approach for image calculations in standard and aberration-corrected LEEM and PEEM. *Ultramicroscopy* **115**, 88–108 (2012).
4. Widodo, C. E., Adi, K. & Gernowo, R. Medical image processing using python and open cv. *J. Phys. Conf. Ser.* **1524**, 012003 (2020).
5. Virtanen, P. *et al.* SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nat. Methods* **17**, 261–272 (2020).