SLAC-TN-12-018

# A Comparison of Image Quality Evaluation Techniques for Transmission X-Ray Microscopy

Peter J. Bolgert

Office of Science, Science Undergraduate Laboratory Internship Program

Marquette University

SLAC National Accelerator Laboratory
Menlo Park, California

August 17, 2011

Prepared in partial fulfillment of the requirement of the Office of Science, Department of Energy's Science Undergraduate Laboratory Internship under the direction of Yijin Liu at Stanford Synchrotron Radiation Lightsource, SLAC National Accelerator Laboratory.

Participant: ___________________________________
Signature

Research Advisor: ___________________________________
Signature

Work supported by US Department of Energy contract DE-AC02-76SF00515.

# TABLE OF CONTENTS

Page

## ABSTRACT

A Comparison of Image Quality Evaluation Techniques for Transmission X-Ray Microscopy. PETER J. BOLGERT (Marquette University, Milwaukee, WI, 53233), YIJIN LIU (Stanford Synchrotron Radiation Lightsource, SLAC National Accelerator Laboratory, Menlo Park, CA 94025).

Beamline 6-2c at Stanford Synchrotron Radiation Lightsource (SSRL) is capable of Transmission X-ray Microscopy (TXM) at 30 nm resolution. Raw images from the microscope must undergo extensive image processing before publication. Since typical data sets normally contain thousands of images, it is necessary to automate the image processing workflow as much as possible, particularly for the aligning and averaging of similar images. Currently we align images using the “phase correlation” algorithm, which calculates the relative offset of two images by multiplying them in the frequency domain. For images containing high frequency noise, this algorithm will align noise with noise, resulting in a blurry average. To remedy this we multiply the images by a Gaussian function in the frequency domain, so that the algorithm ignores the high frequency noise while properly aligning the features of interest (FOI). The shape of the Gaussian is manually tuned by the user until the resulting average image is sharpest. To automatically optimize this process, it is necessary for the computer to evaluate the quality of the average image by quantifying its sharpness. In our research we explored two image sharpness metrics, the variance method and the frequency threshold method. The variance method uses the variance of the image as an indicator of sharpness while the frequency threshold method sums up the power in a specific frequency band. These metrics were tested on a variety of test images, containing both real and artificial noise. To apply these sharpness metrics, we designed and built a MATLAB graphical user interface (GUI) called “Blur Master.” We found that it is possible for blurry images to have a large variance if they contain high amounts of noise. On the other hand, we found the frequency method to be quite reliable, although it is necessary to manually choose suitable limits for the frequency band. Further research must be performed to design an algorithm which automatically selects these parameters.

## INTRODUCTION

Beamline 6-2c at the Stanford Synchrotron Radiation Lightsource (SSRL) is capable of Transmission X-ray Microscopy (TXM) at 30 nm spatial resolution [1]. A transmission X-ray microscope is conceptually identical to a familiar optical microscope, consisting of a light source, a condenser to focus light on the sample, a sample stage, and an objective to focus the transmitted light onto a digital sensor (Figure 1). In this context though, the word “light” refers to synchrotron X-ray radiation, ranging from about 5 to 14 keV per photon. Since X-ray wavelengths are shorter than those of visible light, the X-ray microscope is capable of much higher resolution.

Users from around the world have come to Beamline 6-2c to image a variety of biological and inorganic samples. While it depends on the specific TXM technique being used, the user will typically take thousands of images of a single sample. For example, the user might take images at 180 different angles, taking 20 identical pictures per angle, resulting in 3600 images. If the sample is larger than the field of view (FOV) of the microscope, it becomes necessary to construct a mosaic of sub-images at each angle. If it takes five exposures to cover the sample, the total increases to $5 * 3600 = 18{,}000$ images. Additionally, the user will periodically move the sample off to the side and take some reference images (perhaps several hundred). All of these images must be extensively processed before presentation-quality images and 3D models can be obtained. While the details of the image processing depend on the specific TXM technique used, a typical processing workflow is shown in Figure 2.

With such massive amounts of data, it is necessary to automate the image processing workflow as much as possible, especially for the alignment of similar images with each other. Image alignment is a recurring part of our workflow, as it appears in steps two (Aligning and

Averaging of Multiple Images), three (Mosaic Stitching), and five (Tomographic Reconstruction). As mentioned, the user will typically take 20 to 30 repeated images of the sample at each angle. In between every exposure, the sample stage will jitter slightly, causing these repeated exposures to be identical except for a small translation. Step two of our workflow consists of aligning these repeated images and then taking the average, which increases the signal-to-noise ratio (SNR), a step known as the "advanced average." Not only is it tedious to align images by eye, but a computer should be able to align images much more accurately than a human can. By automating tasks like this, users can focus more on their experiment and less on repetitive image processing.

One commonly used algorithm for aligning two images is known as phase correlation, which manipulates the two images in the frequency domain. First, we calculate the phase correlation function (*PC)* for images *a* and *b*, given by

$$PC = \frac{\mathcal{F}(a)\,\overline{\mathcal{F}(b)}}{|\mathcal{F}(a)||\mathcal{F}(b)|}, \tag{1}$$

where $\mathcal{F}(a)$ is the Fourier transform of image *a* and the overbar denotes complex conjugation [2]. For two images which are identical except for a pure translational shift, the phase correlation function will be given by

$$PC = e^{-2\pi i(\boldsymbol{u}\cdot\boldsymbol{f})}, \tag{2}$$

where $\boldsymbol{u}$ is a vector whose components are the horizontal and vertical shifts (in pixels) from image *a* to *b*, and ***f*** is the position vector in the *PC* matrix. Figure 3 shows the real part of the *PC* function for two images with $\boldsymbol{u} = (10\,pixels\ to\ the\ right, 10\,pixels\ down)$. This cosine wave points in the direction of $\boldsymbol{u}$ and the spacing is inversely proportional to the magnitude of $\boldsymbol{u}$. Since the Fourier transform of a complex exponential is a Dirac delta function [3], we find that $\mathcal{F}^{-1}(PC)$ contains a single spike whose coordinates correspond to the offset, $\boldsymbol{u}$. In other words,

$$offset = coordinates\ of\ \max\big(\mathcal{F}^{-1}(PC)\big). \tag{3}$$

By calculating the offset between the first image and every other image in the stack, we can accurately align all the images to the first image, resulting in a sharp average.

For images containing noise, the situation described above is not completely accurate. In particular, for two images containing high frequency noise, sometimes the algorithm will attempt to align noise with noise, yielding an unsatisfactory result. For example, consider a stack of 30 images of the same sample, which differ by a small translation. However, each image contains a sharp scratch in the exact same location (see Figure 4 for an example). In the case of this data set (which we call the "scratched data set"), there was a scratch on the digital sensor of the microscope. In the FFT, the scratch corresponds to high frequency power, and the phase correlation will align the scratches together, resulting in an offset of $\boldsymbol{u} = (0, 0)$. Since the alignment failed, the resulting average will be blurry (Fig. 4), tainting the rest of the workflow.

To remedy this problem, it is helpful to slightly blur the images during the phase correlation process. We start by multiplying the *PC* function by a two-dimensional Gaussian function of the form

$$f(x, y) = e^{-\frac{g}{2}(x^2+y^2)}, \tag{4}$$

where *g* is a blurring parameter. The greater *g* is, the narrower the Gaussian peak. Multiplying *PC* by a Gaussian masks the outer region of the *PC* function, which corresponds to high frequency noise in the original images. Once the parameter *g* passes a critical value, the maximum of $\mathcal{F}^{-1}(PC)$ shifts to the desired offset. Figure 5 shows the evolution of the *PC* function for two scratched images as *g* is increased. As *g* increases, the center spot begins to look more and more like the ideal sinusoid of Figure 3. Figure 5 also shows the evolution of $\mathcal{F}^{-1}(PC)$ with increasing *g*. You can see that there are two blobs competing for attention. The off-center blob corresponds to the features of interest (FOI). As $g$ increases, both blobs fade, but the off-center blob fades more slowly. By $g$ = 8, the maximum of $\mathcal{F}^{-1}(PC)$ occurs in the off-center blob, resulting in a correct alignment. Finally, Figure 5 also shows the evolution of the resulting averages when phase correlation is performed for the entire stack of the twenty “scratched” images. As you can see, when $g$ crosses a critical value, the resulting average is as sharp as any of the starting images, but with much less noise (high SNR).

The process of blurring during phase correlation works well, but it requires manual user input for choosing $g$, slowing down the workflow. In order to make this process fully automatic we need an algorithm to evaluate image quality, particularly sharpness. If the averaged image is not sharp enough, the algorithm should adjust the blurring so that the alignment is optimized. All of this should occur without any input from the user. There are many examples of sharpness algorithms in the literature, many of which are surveyed in Ferzli and Karam [4]. In this paper, we will explore a variance-based algorithm and an FFT-based algorithm in an attempt to further automate our image processing workflow.

## Materials and Methods

All of the work for this project was performed using MATLAB R2010b, equipped with the Image Processing Toolbox. Every step in the TXM image processing workflow (Fig. 2) can be performed with TXM Wizard, a suite of Matlab software developed by Yijin Liu, Florian Meirer, and Phillip Williams. TXM Wizard can be downloaded for free at http://sourceforge.net/projects/txm-wizard. The program used to align and average images (step two in Fig. 2) is known as Advanced Average.

In order to test different sharpness algorithms, I used two test data sets. The first data set is the “scratched data set” described above. It consists of thirty 1024 x 1024 pixel TIFF images taken of an industrial catalyst. The second data set, consisting of fifty 1024 x 1024 pixel TIFF images, is known as the “root data set” since the images are of tiny plant roots (Figure 6). This data set has been doctored to contain both high and low frequency noise. The high frequency noise consists of small artificial scratches, which have the same position for each image. The low frequency noise consists of broad bright and dark patches which shift randomly from image to image. Figure 6 also shows the evolution of alignment for this data set. In the computer, these images are stored as 1024 x 1024 matrices with integer values ranging from 0 (black) to 255 (white).

Next I will describe the two sharpness algorithms which were used. The first is called the variance method. The variance of a data set (such as matrix of integers) is the square of the standard deviation and is equal to

$$var = \sum_i (x_i - \langle x \rangle)^2, \tag{5}$$

where $x_i$ represents a data point and $\langle x \rangle$ is the mean of the data set. Now, it seems intuitively obvious that a sharp “advanced average” should have a larger variance than a misaligned, blurry “advanced average.” In a sharp image, there will be very light pixels and very dark pixels, both deviating far from the mean. In a blurry image, the features are averaged out. The light pixels are not as light, and the dark are not as dark, resulting in a lower variance. This is demonstrated in Figure 7 for a random 20x20 matrix.

The second sharpness algorithm used is related to the Fast Fourier Transform (FFT) and is called the frequency threshold method. A 2D FFT is a pictorial depiction of the frequency distribution in a signal (Figure 8). Low frequencies appear in the center, and high frequencies away from the center. In the case of a digital image, we are concerned with the spatial frequency, that is, how quickly pixels values change as we move across the image. The highest frequency occurs when a black pixel (value 0) is adjacent to a white pixel (value 255), and this would show up at the fringes of the FFT. A quality image will contain sharp edges, with pixel values changing quickly from light to dark. This corresponds to high frequency power in the FFT. Thus we expect sharp images to contain more high frequency power than a blurry image.

To quantify the sharpness, we choose a suitable radius and then sum up all the absolute values of the FFT matrix which lie outside this value (technically, the term “power” refers to the absolute square of the FFT values, but we will use “power” to refer to the absolute value in this paper). However, we do not want to sum all the way to the outer boundary. While the features of interest in an image are sharp, it will generally take at least a few pixels to transition from light to dark. Any FFT power more than a certain distance from the origin corresponds to undesirable high frequency noise. Thus to quantify the sharpness of an image, it suffices to sum up the FFT power in between suitably sized concentric circles (Figure 8). The radii of the threshold circles ($R_{in}$ and $R_{out}$) must be experimentally determined.

One way to determine suitable values of $R_{in}$ and $R_{out}$ is to simply compare FFT’s of blurry and sharp images. Figure 8 shows FFT’s of two “advanced averages” for the root set. In the blurry average ($g$ = 0), you can see that while most power is concentrated in the center, there is power scattered all the way out to the edges, corresponding to the scratches. The sharp average ($g$ = 5) has more power in the mid-range region of the FFT. Choosing $R_{in}= w/32$ and $R_{out}=w/16$ would be an appropriate choice for this data set, where $w$ is the width of the cropped “advanced average.”

Another way to choose $R_{in}$ and $R_{out}$ is to plot the power in the FFT's as a function of the distance from the center. Figure 9 shows such a plot for the root data "advanced averages" in Figure 8. For each pixel in the image, its distance from the origin was calculated and rounded to the nearest integer (say $r = r_0$). By adding together all elements at this rounded distance, we obtain $|FFT(r_0)|$. From Fig. 9 we can see that most of the power corresponding to the features of interest lies between $R_{in}= w/30$ and $R_{out} = w/10$, generally consistent with the values mentioned previously. In reality, choosing radii is somewhat arbitrary and it pays to look at the problem in more than one way.

In order to carry out my research I designed and built a Matlab graphical user interface (GUI) called Blur Master. A screenshot is shown in Figure 10. Blur Master is used to create "advanced averages" for a customizable range of $g$ (blurring) values. After writing all the "advanced averages" to file, Blur Master automatically quantifies the sharpness of the averages using both the variance and frequency threshold methods. All images are cropped by 20% on each side before evaluating the sharpness. I also designed a sub-GUI, called Power Plotter to plot FFT power as a function of the distance from the center, which can be used to choose appropriate values for $R_{in}$ and $R_{out}$. A screenshot of Power Plotter is shown in Figure 11.

## RESULTS OF SHARPNESS TESTING

Table 1: Scratched Data Set
($R_{in}= w/32$ and $R_{out} = w/16$ for freq. method)

| g (blurring) | variance method | frequency method |
|---|---|---|
| 0 | 270.7 | 1.012 |
| 1 | 270.7 | 1.015 |
| 2 | 270.9 | 1.019 |
| 3 | 270.9 | 1.021 |
| 4 | 261.0 | 1.028 |
| 5 | 263.3 | 1.190 |
| 6 | 282.0 | 1.580 |
| 7 | 286.5 | 1.667 |
| 8 | 313.9 | 1.987 |
| 9 | 313.9 | 1.987 |
| 10 | 313.9 | 1.985 |
| 11 | 314.0 | 1.984 |
| 12 | 314.0 | 1.989 |
| 13 | 314.0 | 1.989 |
| 14 | 314.0 | 1.988 |
| 15 | 314.0 | 1.988 |
| 16 | 314.0 | 1.988 |
| 17 | 314.1 | 1.990 |
| 18 | 314.1 | 1.991 |
| 19 | 314.1 | 1.990 |
| 20 | 314.1 | 1.990 |
| 30 | 314.1 | 1.991 |
| 40 | 314.1 | 1.990 |
| 50 | 314.0 | 1.986 |
| 60 | 314.0 | 1.985 |
| 70 | 313.9 | 1.983 |
| 80 | 313.8 | 1.977 |
| 90 | 313.8 | 1.975 |
| 99 | 313.7 | 1.966 |

Table 2: Root Data Set
($R_{in}= w/32$ and $R_{out} = w/16$ for freq. method)

| g (blurring) | variance method | frequency method |
|---|---|---|
| 0 | 118.2 | 2.867 |
| 1 | 118.2 | 2.867 |
| 2 | 118.2 | 2.867 |
| 3 | 118.2 | 2.867 |
| 4 | 104.5 | 6.736 |
| 5 | 104.8 | 6.752 |
| 6 | 104.9 | 6.755 |
| 7 | 105.0 | 6.760 |
| 8 | 105.1 | 6.760 |
| 9 | 105.2 | 6.763 |
| 10 | 105.3 | 6.766 |
| 11 | 105.3 | 6.766 |
| 12 | 105.3 | 6.766 |
| 13 | 105.3 | 6.766 |
| 14 | 105.3 | 6.766 |
| 15 | 105.3 | 6.766 |
| 16 | 105.3 | 6.766 |
| 17 | 105.3 | 6.766 |
| 18 | 105.3 | 6.766 |
| 19 | 105.3 | 6.766 |
| 20 | 105.3 | 6.766 |
| 30 | 105.1 | 6.764 |
| 40 | 104.0 | 6.737 |
| 50 | 102.1 | 6.687 |
| 60 | 99.2 | 6.596 |
| 70 | 96.6 | 6.482 |
| 80 | 91.7 | 6.226 |
| 90 | 87.0 | 5.896 |
| 99 | 83.2 | 5.552 |

## DISCUSSION

The tables above show the sharpness results for both the scratched and root data sets. For each value of *g*, we created an "advanced average" and quantified its sharpness. For the scratched data set, we see that both the variance and frequency threshold values start low, increase dramatically from $g = 5$ to 8, and then level off. These results are consistent with our visual sense of sharpness. The "advanced averages" start out blurry, start aligning between $g = 5$ to 8, and then look the same from that point on.

For the root data set, the results are mixed. While the frequency threshold algorithm performed well, the variance algorithm finds the *misaligned* averages to be the "sharpest"! In this case the variance contributed by the noise was greater than the variance contributed by the sharp features of interest. The variance method is not expected to perform consistently from one data set to another because it treats sharp features and high frequency noise on equal footing. Note that the same frequency parameters were the same for both data sets ($R_{in}= w/32$ and $R_{out} = w/16$). These values were found experimentally through trial and error. We conclude that these parameters should work for "typical" TXM images, although certainly more tests are needed.

Now that we have gained some insight into image quality evaluation, there are several directions for continued research. The literature is rich with various sharpness algorithms, including several which claim to be immune to noise [4], [5]. There also exist "no-reference" sharpness metrics which can align images without any prior knowledge of the images (i.e. it is not necessary to manually adjust parameters) [4], [5]. By implementing more sophisticated and robust algorithms, we can improve the ease and reliability of our GUI's.

In addition to adding more sophisticated algorithms, we would like to completely automate the advanced average process. In the ideal scenario, when images are loaded into the

Advanced Average GUI, the computer would automatically find the ideal *g* value, and then return only the sharp average. One way to achieve this would be to 1) calculate the "advanced average" at a default *g* (say *g* = 8), 2) evaluate the sharpness, 3) calculate the "advanced average" at adjacent *g*'s (*g* = 8 and *g* = 9), 4) evaluate these sharpnesses, and 5) continue outward until reaching a local maximum or a plateau. A plateau is reached when adjacent sharpnesses differ less than a specified amount (say 0.1%).

The main problem with this scheme is that is it is difficult to choose $R_{in}$ and $R_{out}$ ahead of time. Values that work for one data set may not work for another data set. It seems foolish to rely on "typical values" such as $R_{in} = w/32$ and $R_{out} = w/16$. We would like to choose these parameters automatically for each data set. We can potentially solve this problem by looking at the FFT vs. radius plot for one of the original images (see Figure 12). Since the features of interest are obviously sharp in the original image, this plot should display a local maximum at these frequencies. By evaluating the properties of this maxima (e.g. the full-width-at-half-max), suitable values of $R_{in}$ and $R_{out}$ could be chosen automatically.

While the "advanced average" blurring process is not yet automatic, the Blur Master GUI could still be useful for users in its current form. For example, if the user has multiple sets of similar data (e.g. 20 images for one angle, then 20 images for the next angle, etc.), he/she could run Blur Master on the first 20 images to find the best *g* value, and then apply the same blurring factor to all subsequent data sets.

In conclusion, two sharpness algorithms were used to evaluate the quality of alignment in TXM images. The frequency threshold method was found to be more reliable; however, it requires manual input from the user. Further research must be performed in order to automatically select these frequency parameters. In addition, “noise immune” metrics and “no-reference” metrics should be investigated.

## Acknowledgments

This work was supported by SLAC National Accelerator Center, the U.S. Department of Energy, and the Office of Science. I would like to give special thanks to Yijin Liu, who guided me, and to Allison Hume, who provided a stimulating working environment. Finally, I would like to thank anybody who took the time to read this research paper.

## References

[1] J.C. Andrews et al., “Nanoscale X-Ray Microscopic Imaging of Mammalian Mineralized Tissue,” *Microscopy and Microanalysis,* vol. 16, pp. 327 – 336, 2010.

[2] R. Szeliski, “Image Alignment and Stitching: A Tutorial,” *Foundations and Trends in Computer Graphics and Vision*, vol. 2, pp. 1 – 104, 2006.

[3] R.N. Bracewell, *The Fourier Transform and its Applications*, 3rd ed., New York: McGraw-Hill, 1999.

[4] R. Ferzli and L.J. Karam, “No-Reference Objective Wavelet Based Noise Immune Image Sharpness Metric,” *IEEE International Conference on Image Processing*, vol. 1, pp. 405 – 408, 2005.

[5] X. Zhu and P. Milanfar, “A No-Reference Sharpness Metric Sensitive to Blur and Noise,” *International Workshop on Quality of Multimedia Experience*, pp. 64 – 69, 2009.

## FIGURES

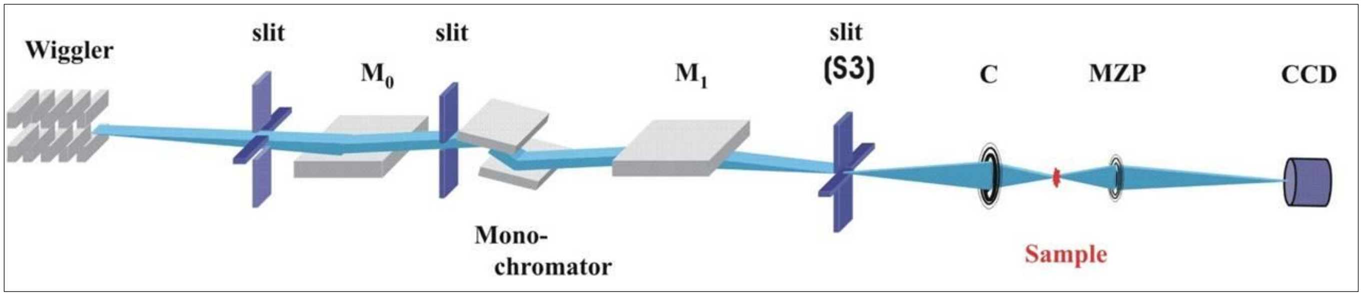


Figure 1: Schematic of the Transmission X-Ray Microscope at SSRL Beamline 6-2c. Synchrotron radiation, which contains a wide range of frequencies, is passed through the monochromator to select a single frequency. The condenser (C) focuses the X-rays on the sample, and the micro-zone plate (MZP) focuses the transmitted X-rays onto the Charge Coupled Device (CCD). The microscope also contains various slits and mirrors (M0, M1) to collimate and direct the beam.

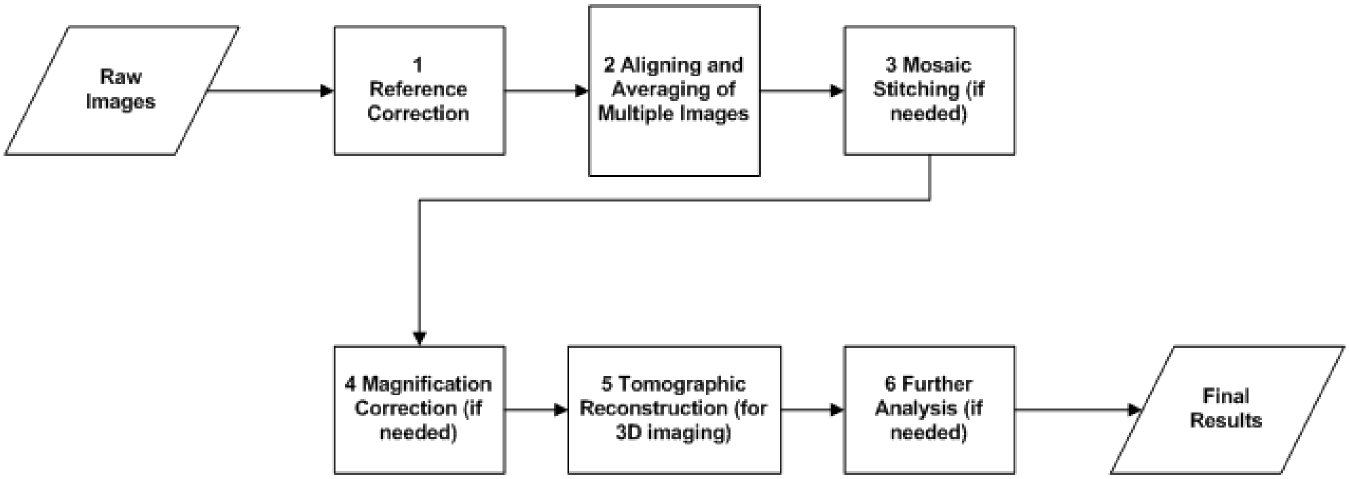

Raw Images
1 Reference Correction
2 Aligning and Averaging of Multiple Images
3 Mosaic Stitching (if needed)
4 Magnification Correction (if needed)
5 Tomographic Reconstruction (for 3D imaging)
6 Further Analysis (if needed)
Final Results

Figure 2: Image Processing Work Flow. Starting with raw images, the first step is to subtract references images, which are images taken with the sample moved outside the Field of View. Subtracting the references should eliminate the background and other types of noise (due to the digital sensor for instance). In step two, we align nearly identical images (see text for explanation) and then averaging them. Automating this process is the subject of the present paper. In step three, we stitch images together to create a larger mosaic. This is necessary when the sample is larger than the Field of View of the microscope. Step four, magnification correction, is needed when we image the sample at multiple X-ray energies. Magnification is a function of energy, and so we need to scale all images to a common size. Step five is used for tomographic techniques, in which we take images at multiple angles. The reconstruction algorithm converts these angled views to a series of cross-sectional slices, much like an MRI. Finally, further analysis may be necessary for more advanced techniques. In order to completely automate this workflow we need automatic image quality control, as discussed in the text.

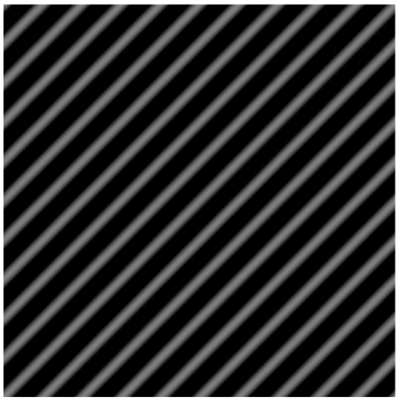


Figure 3: The real part of the phase correlation function for two images differing by a pure translational offset.  The offset u is given by u = (10 pixels to the right, 10 pixels down).  In the ideal situation, the real part of this function is a cosine pointing in the direction of the offset.  The spacing, or wavelength, is inversely proportional to |u|.

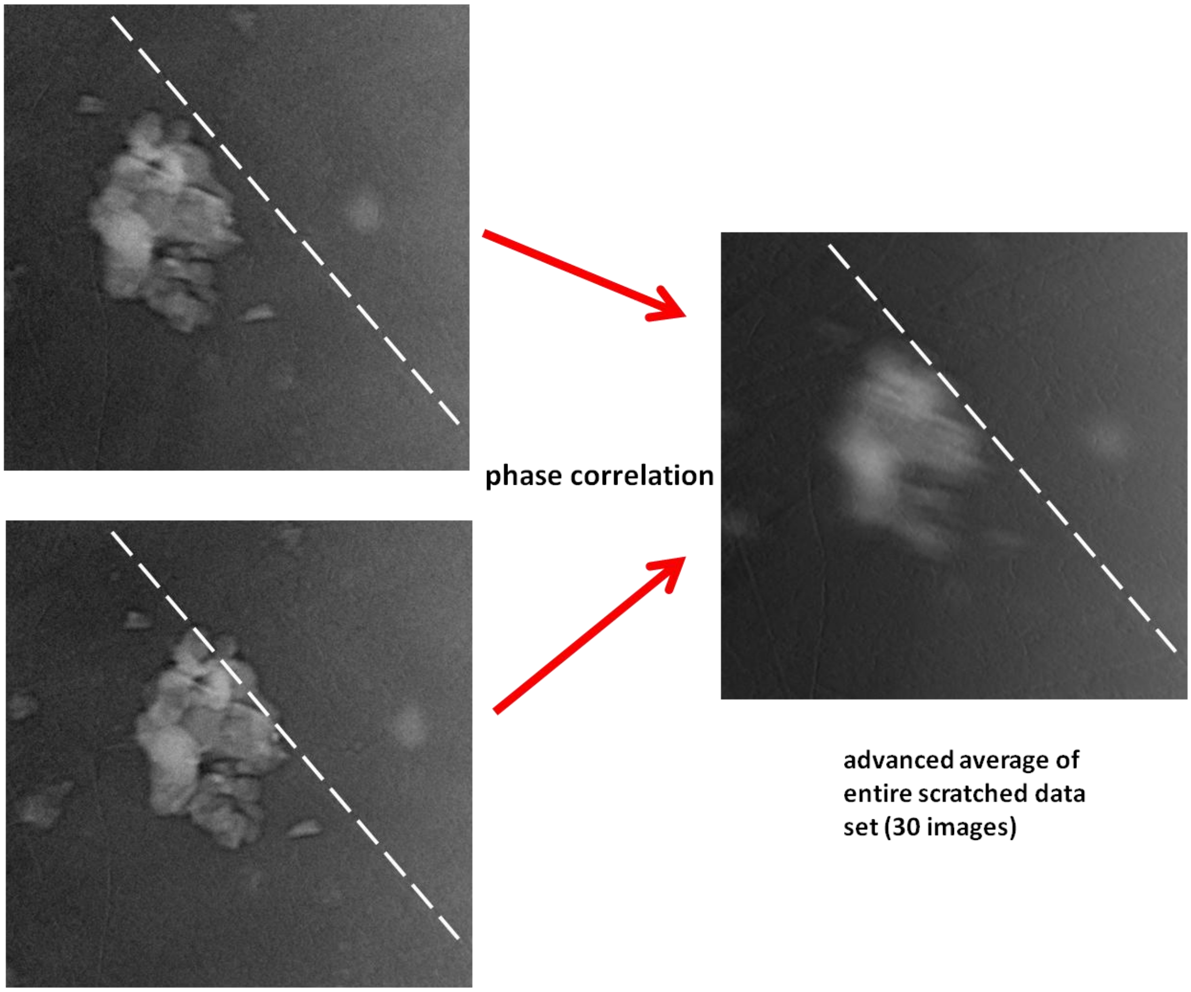


Figure 4: (LEFT) Two of the 30 images in the scratched data set.  Both images have a faint scratch in the same location, while the catalyst is offset to the right in the second picture. Since the scratch is quite faint, its position is indicated by the dotted white line.  (RIGHT) The "advanced average" of the entire image stack using phase correlation.  The scratches all align, resulting in a sharp scratch and a blurry catalyst.

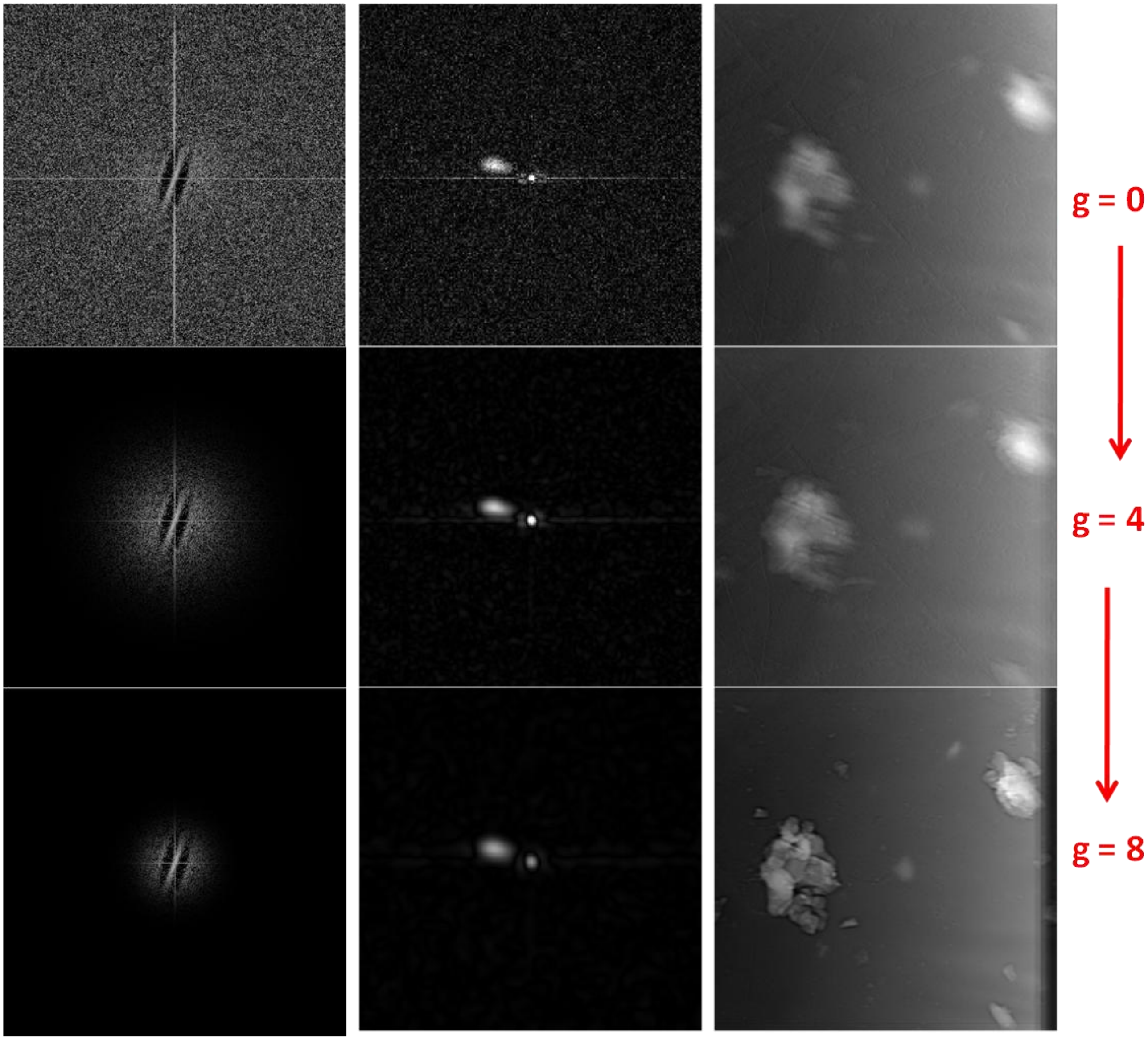


Figure 5: (LEFT) The evolution of the PC function (real part) between two scratched images as g increases. As g increases, the PC function begins to look like a cosine, reminiscent of Figure3. (MIDDLE) The evolution of the corresponding $\mathcal{F}^{-1}(PC)$ functions. By g = 8, the maximum is located in the off-center blob. (RIGHT) The evolution of the "advanced average" of the entire image stack with increasing g.

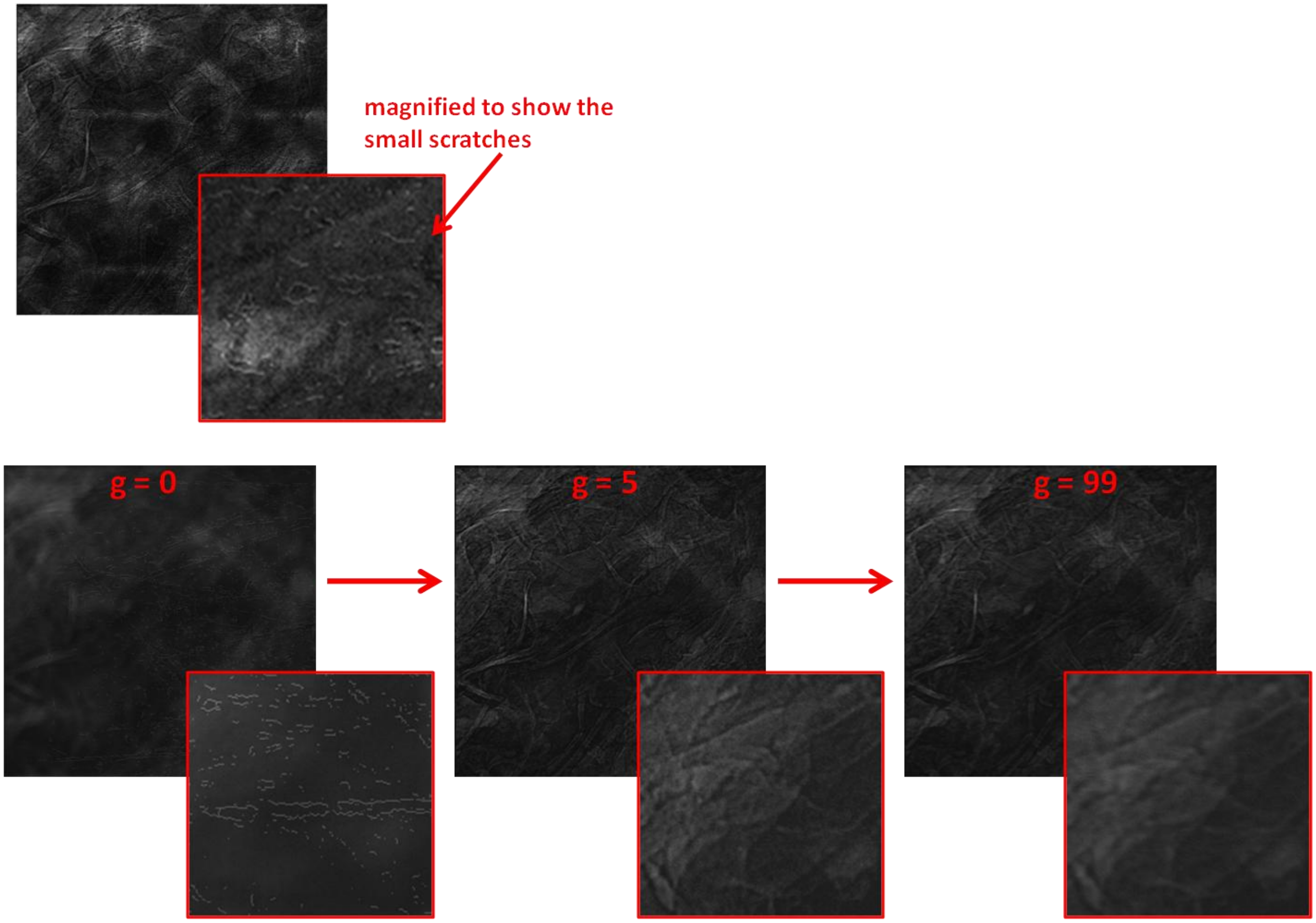


**Figure 6: (TOP) One image from the root data set, with a blown up inset to show the high frequency artificial scratches. (BOTTOM) The evolution of alignment with increasing g for the root data set. For g = 0, the artificial scratches are aligned, resulting in blurry features. By g = 5, the features of interest are aligning properly. By g = 99, the "advanced average" is starting to get slightly blurred, which can be seen in the inset.**

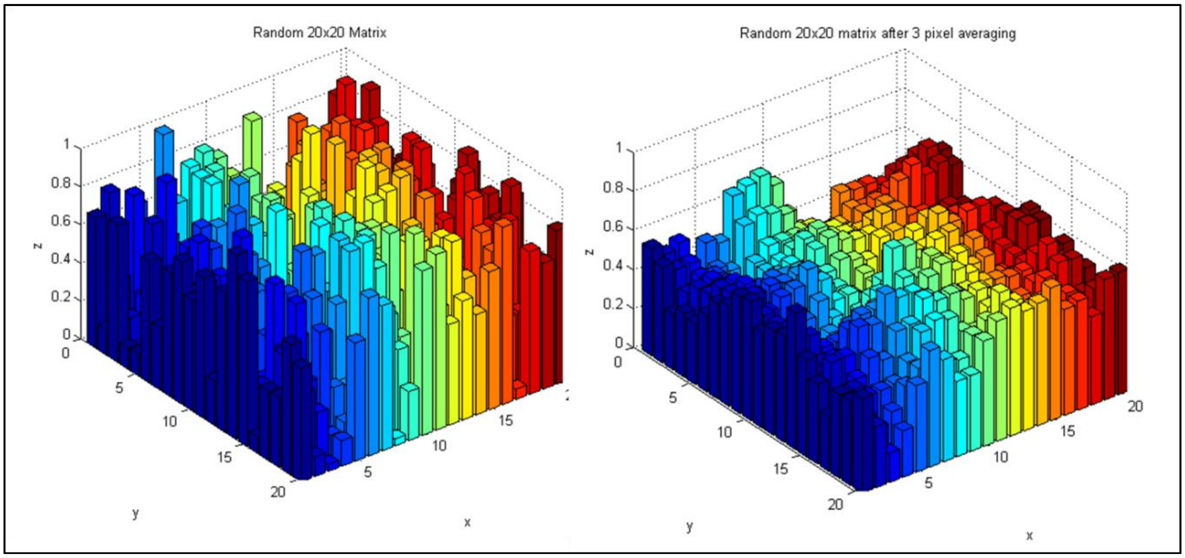


Figure 7: (LEFT) A random 20x20 matrix to represent a "sharp" image; (RIGHT) The same matrix after a 3x3 pixel averaging filter has been applied. This matrix represents a blurred image, and its variance has decreased.

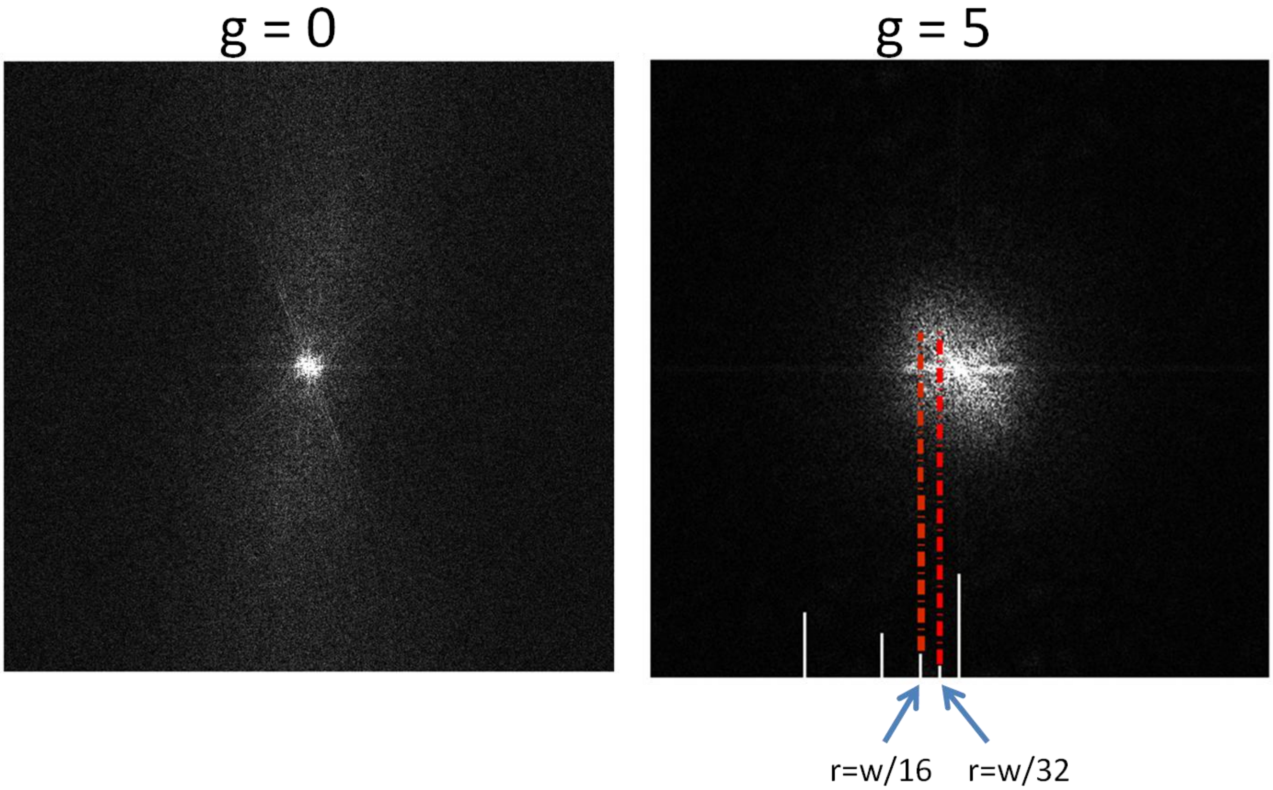


Figure 8: A comparison of the FFT's of a blurry "advanced average" (g = 0) and a sharp average (g = 5). Most of the frequency power corresponding to the features of interest lies between r=w/32 and r=w/16, where w is the width of the cropped "advanced average."

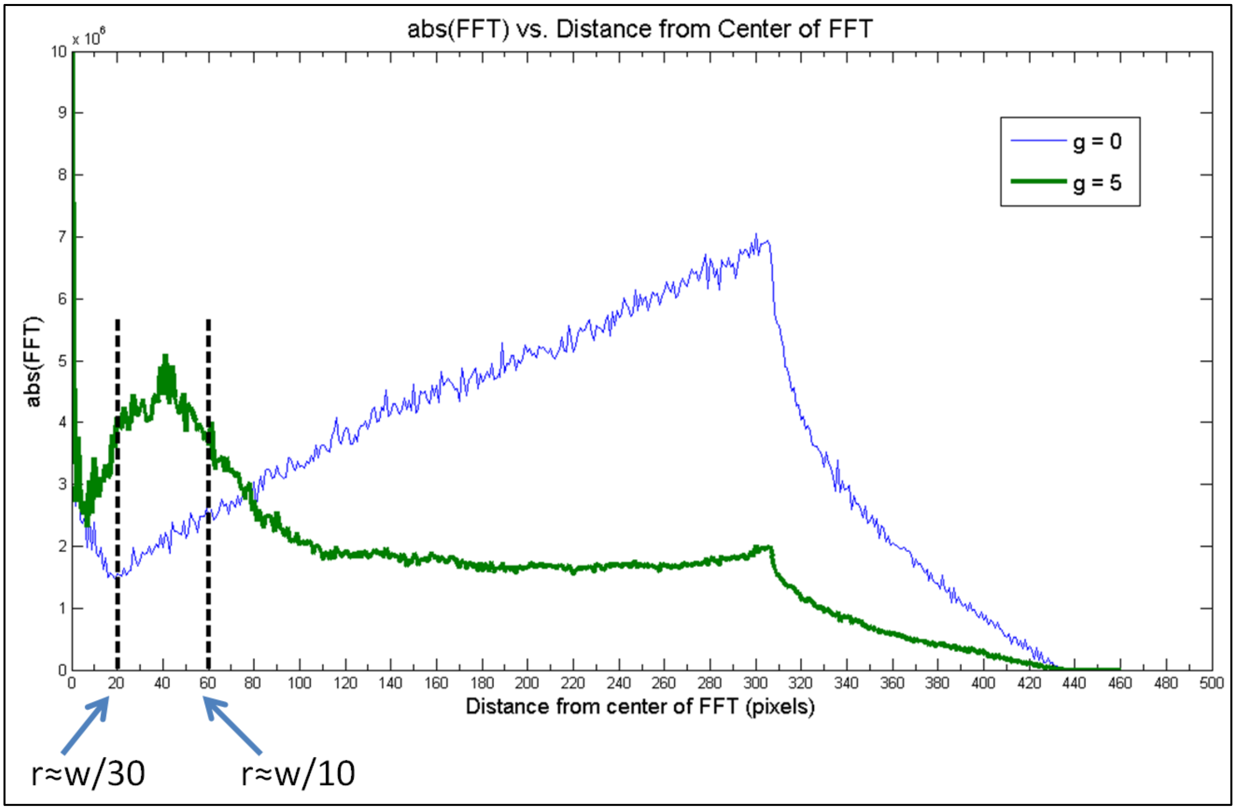
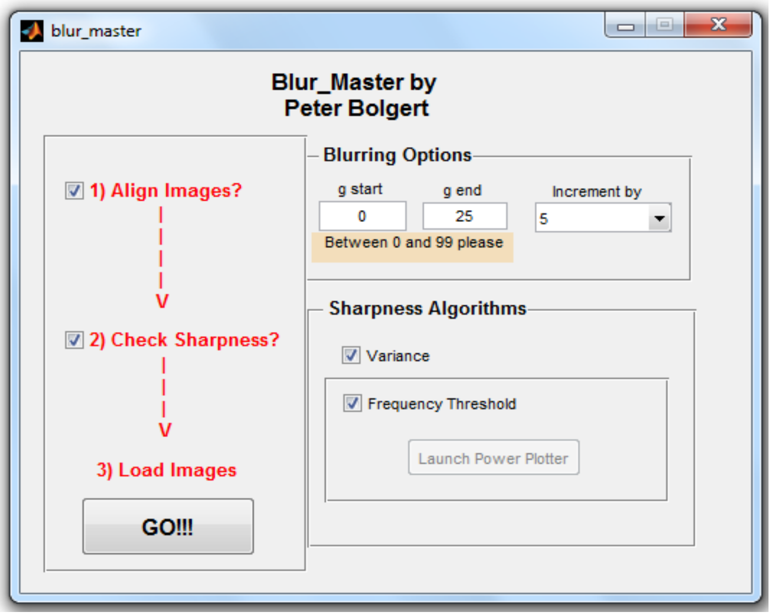


Figure 9: Plots of abs(FFT) vs. distance radius for "advanced averages," a blurry average (g = 0) and a sharp average (g = 5). These plots are for the root data set. In the sharp image (g = 5), there is a peak between r=w/30 and r=w/10, where w is the width of the cropped "advanced average." This peak corresponds to the sharp features of interest in the sharp average. The power drops off after about 300 pixels because we are starting to sum up the power in the corners of the FFT's, where the area is less.



blur_master

**Blur_Master by Peter Bolgert**

☑ 1) Align Images?

☑ 2) Check Sharpness?

3) Load Images

Blurring Options

g start: 0

g end: 25

Between 0 and 99 please

Increment by: 5

Sharpness Algorithms

☑ Variance

☑ Frequency Threshold

Figure 10: Screen shot of Blur Master. This GUI is used to produce "advanced averages" for a range of g (blurring) values, and also to quantify their sharpness using the two algorithms described in the text.

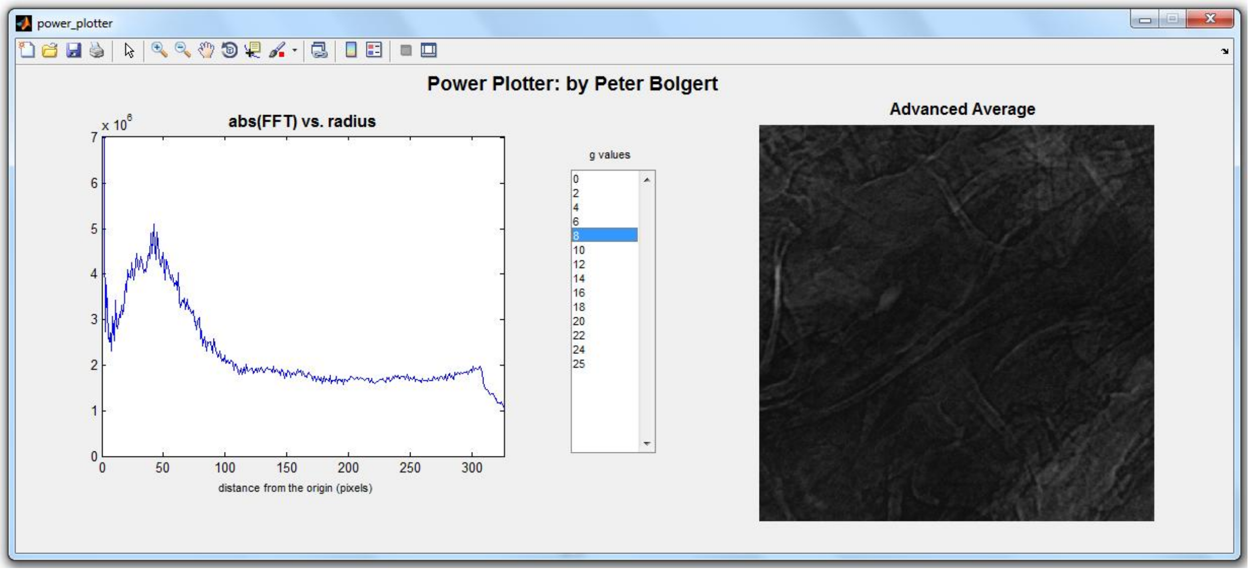


Figure 11: Screenshot of Power Plotter. This sub-GUI allows the user to simultaneously see each "advanced average" alongside a plot showing its FFT power vs. the distance from the FFT center, similar to Figure 9. This sub-GUI could be used to pick appropriate frequency bands to quantify the sharpness of subsequent data sets.

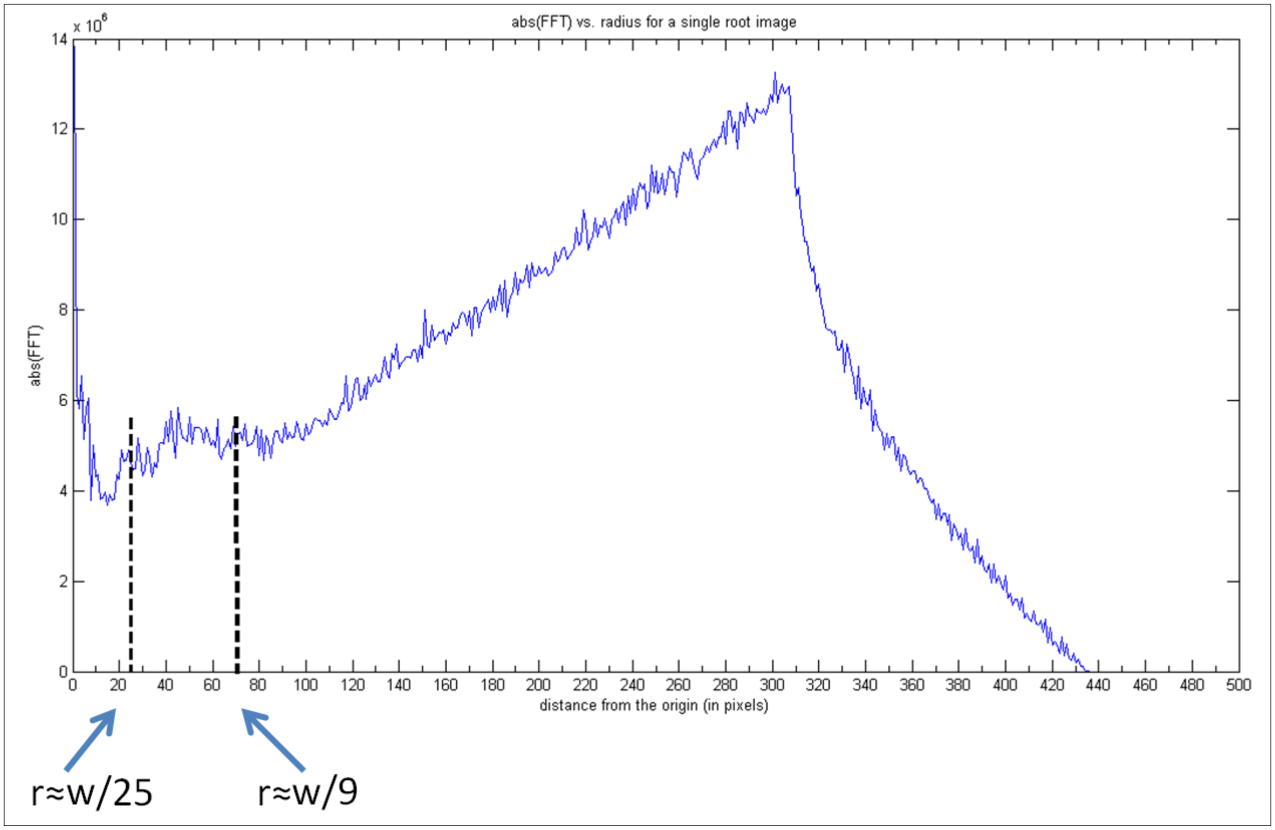


Figure 12: Plot of abs(FFT) vs. radius for a single root image. While this image contains high frequency noise, you can see that there is a small local max between r=w/25 and r=w/9, where w is the width of the cropped image. This local maximum corresponds to the sharp features of interest in the original image. We expect any sufficiently sharp "advanced average" to show a similar hump.