Multi-Angle Plasma FIB Curtaining Artefact Correction using a Fourier-based Linear Optimization Model

Christopher W. Schankula[†] schankuc@mcmaster.ca
Christopher Kumar Anand[†] anandc@mcmaster.ca
Nabil D. Bassim[‡] bassimn@mcmaster.ca

Department of Computing and Software[†], and Department of Material Science and Engineering[‡], McMaster University
Hamilton, Ontario, Canada L8S 4K1

We present a flexible linear optimization model for correcting multi-angle curtaining effects in Plasma Focused Ion Beam Scanning Electron Microscopy (PFIB-SEM) images. PFIB-SEM is a serial sectioning tomography technique capable of imaging large 3-dimensional volumes quickly, providing rich information in the critical 10-100nm feature length scale. During tomogram acquisition, a "rocking polish" is often used to reduce straight-line "curtaining" gradations in the milled sample surface. While this mitigation scheme is effective for deep curtains, it leaves shallower line artefacts at two discretized angles. Segmentation and other automated processing of the image set requires that these artifacts be corrected for accurate microstructural quantification. Our work details a new Fourier-based linear optimization model for correcting curtaining artifacts by targeting curtains at two discrete angles. We demonstrate its capabilities by processing images from a concrete tomogram. We present methods for selecting the parameters which meet the user's goals most appropriately. Compared to previous works, we show that our model provides effective multi-angle curtain correction without introducing artefacts into the image, modifying non-curtain structures or causing changes to the contrast of voids. Our algorithm can be easily parallelized to take advantage of multi-core hardware.

Computing and Software Report CAS-18-01-CA

The authors thank NSERC for financial support.

Introduction

In order to gain a proper understanding of structure-processing-property relationships in materials, it is crucial to characterize the three-dimensional (3D) topology by means of tomography and image processing. Topological features within the 10 to 100nm length range and their contexts in relation to other structures have a large influence on the material properties and are therefore of interest to microscopists studying a large range of materials (**??**).

Focused Ion Beam Scanning Electron Microscopy (FIB-SEM) tomography is based on alternating FIB slicing and SEM imaging, in order to obtain a stack of two-dimensional (2D) cut-face images, forming a three-dimensional (3D) tomogram (**?**). Unlike non-destructive 3D imaging techniques, e.g. X-ray computed tomography (CT), positron emission tomography (PET) and nuclear magnetic resonance imaging (MRI), FIB tomography is capable of providing sufficient contrast and spatial resolution to resolve this important length size range (**?**). Additionally, since the probing beam for the tomogram is an electron beam, there is the possibility of obtaining a rich variety of signals showing composition (e.g. backscatter electrons, energy-dispersive X-ray spectroscopy — EDS) and crystal orientation (electron backscatter diffraction — EBSD) using FIB-SEM tomography (**?**).

Once a 3D tomogram is obtained, further quantitative and qualitative analysis is often performed, including statistical analysis of phase size and distribution or 3D visualizations of the phases (e.g. those described in **?**). Further, secondary and backscatter electron images can be simultaneously complemented by other imaging modes to obtain 3D distributions of elements in a specific volume (3D elemental maps) and three dimensional texture analysis using electron backscatter diffraction (EBSD) (**?**).

Emerging Xe^+ Plasma FIB (PFIB) beam replaces the traditional Ga^+ ion beam on the microscope, promising much faster removal rates — thereby greatly increasing the feasible volume which can be analyzed in a given timeframe (visualized in figure **??**). By using proper imaging and milling techniques, milling rates of 60x faster can be achieved with a minimal amount of imaging artefacts (**?**).

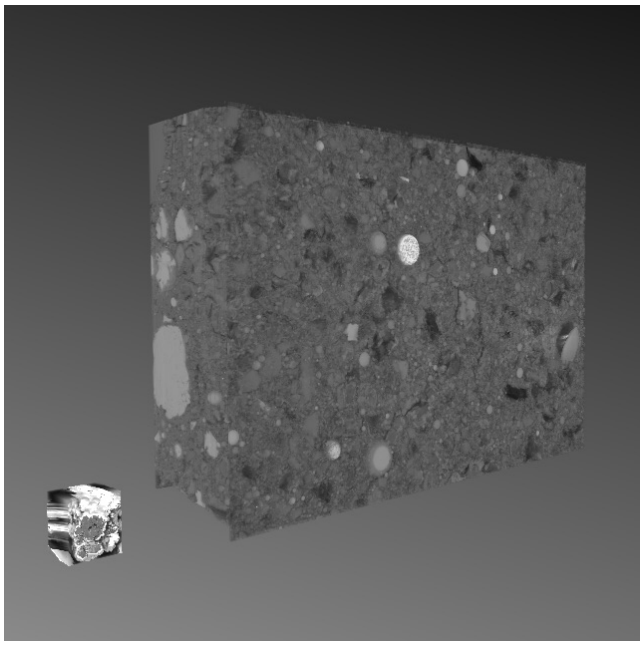


Figure 1: *Plasma FIB provides much faster milling rates than traditional FIB, allowing imaging of much larger volumes in comparable timeframes. A meteorite dataset (**?**) of of 6,110μm^3 imaged by traditional FIB is compared against our PFIB concrete dataset of 529,000μm^3, both having been imaged in approximately 72h.*

This allows microscopists to quantify heterogeneous multiphase structures like rocks, concrete and biomaterials at the mesoscale (100nm - 200μm).

Curtaining

However, due to the inhomogeneous nature of the phases in the sample, milling rates vary within a specified cut face, causing vertical ripples on the surface of the sample. These curtaining effects appear as darker or lighter straight lines on the secondary and backscatter electron images (see figure **??**). Porous PFIB samples exhibit particularly deep curtaining effects downward from the pores due to a higher milling rate attacking the bottom of the pore (**?**). These curtaining effects pose challenges to further quantitative analysis of the 3D tomogram as many of these algorithms are reliant on accurately detecting edges within the images.

There are several means to reduce curtaining artefacts, including polishing the free face of the sample

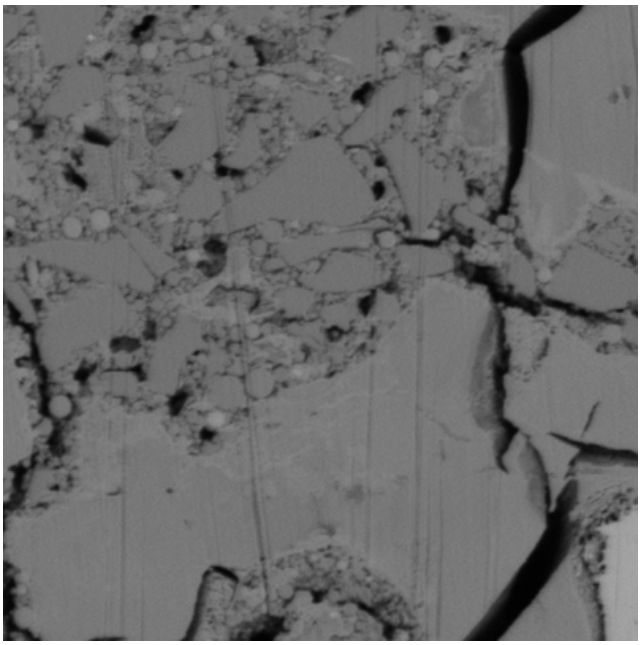


Figure 2: *Curtains appear on images as lines darker or lighter than the surrounding pixels. Due to the rocking mill technique, they appear at two discretized angles in our sample.*

cube using a low ion current (**????**), applying a Pt mask layer or sputter-coated gold films to the sample (**??**), and using a "rocking polish" or multiple-angle ion beam technique to counteract curtains as they form (**??**). The latter method is of interest to us in this paper.

UHPC Test Sample

In order to test our algorithm, we imaged a sample of ultra high-performance concrete (UHPC) prepared one month prior to image acquisition (3D tomogram is shown in figure **??**). UHPC is a low-water, low-porosity structure exhibiting high strength and other favourable mechanical and durability properties (**?**). This makes UHPC an ideal candidate in addressing concerns about the deterioration, repair and replacement of modern infrastructure (**?**). Quantifying the 3D microstructure of materials like concrete is crucial to understanding how to further improve their strength, durability and service life, thereby lowering their financial and environmental costs.

The sample was imaged with an FEI-Thermo Fisher PFIB using a backscatter electron (BSE) detector at 2 keV. The initial cutting surface was prepared with a cleaning cross-section and tungsten was deposited on the top of the sample surface to protect the sample from curtaining. The sample was periodically aligned to fiducial markers to align the beam. A rocking polish was used to to reduce major curtaining while the cut face remained perpendicular to the beam, leaving minor curtains at approximately 7° & -1° counter-clockwise from the vertical direction (see figure **??**).

The resulting dataset contains 705 16-bit grayscale images at 3072px by 2048px each. Each pixel on the x-y plane has a scale of approximately 49nm with a 50nm slice thickness. This gives a total dimension of 150μm by 100μm by 35.3 μm, for a total volume of approximately 529,000μm^3. The dataset was re-sampled to 8-bit grayscale using ImageJ (**?**) for more efficient data processing. Containing both complicated heterogeneous areas with multiple phases and many different shapes of pores as well as large homogeneous areas, it is an ideal dataset in order to test the ability of our algorithm to correct curtains in many situations.

Although not required for the algorithm described in this paper, to account for small alignment errors among slices during image acquisition, and to allow for comparison with **?**'s algorithm which requires an aligned dataset, the ImageJ StackReg plugin (**?**) was used in "Rigid Body" mode to fix both translational and rotational misalignments.

Linear Model Design Principles

Our model makes the following assumptions about curtaining artefacts, although the resulting algorithm may work in more general situations:

A1. Curtains do not deviate from their straight-line path along one of the specified angles.

A2. The deviation of the height of the curtain from the cut surface is much less than the distance between cut surfaces.

A3. There are no "real" structures which look like and are parallel to curtains.

A4. Curtaining artefacts never appear on dark voids.

Assumption **A1** is a valid assumption in most cases. In rare instances the curtains may deviate slightly. Assumption **A2** is required for a 2D algorithm, which operates on one slice at a time. Assumption **A3** limits the types of samples that can be processed with our algorithm. If there are structures in the sample that are parallel to the curtaining direction and of similar width (i.e. nanoscale channels or printed metal lines in a semiconductor), these may be effected. Finally, **A4** encodes the fact that curtains are a surface artefact and therefore do not show up in the dark void sections of the image.

In order to design an algorithm that is useful in a pipeline of image processing and quantification steps, it must satisfy several criteria, against which we will later analyze our algorithm:

C1. It should be able to correct curtaining artefacts at arbitrarily specified angles.

C2. It should cause minimal change to image structures which aren't curtains.

C3. It should not lighten or reduce the contrast of voids, which show up as very dark areas.

Criterion **C3** is critical not only because of **A4** but because the study of void distribution and networks using FIB is very important. For example, pore analysis is critical in the characterization of gas shales (**????**), geological material properties analysis (**?**) and the analysis of the structural properties of concrete.

Previous Works

Although physically reducing curtaining through improved sample preparation and techniques like rocking mill is an important first step, artefacts may still remain despite those efforts. Therefore, there have been many past works aimed at removing curtaining artefacts post-acquisition with computational methods. These include works aimed specifically at correcting the curtaining problem as well as those aiming to remove all types of striped noise, of which curtaining is a subset. Several authors, including **?** and **?**, have proposed general frequency-domain-based methods of removing stripe artefacts from images. **?**, have proposed wavelet-based filtering techniques. **?** combines Fourier and wavelet filtering.

? builds on the variational models of **?** and **?** and proposes a 3D convex variational model aimed at removing curtaining effects as well as the "laminar" artefact, which we do not target in this work.

FEI's Avizo 9.0 software has a tool for aligning a stack of FIB-SEM images and correction of curtains at specific angles; however, to target multiple angles the wizard must be run multiple times.

To the best of our knowledge, no algorithm has been proposed aiming specifically to correct rocking mill PFIB imaging artefacts at multiple angles.

Methods

Our overall approach consists of three linear optimization models solved in succession. The first linear optimization model (**??**) corrects curtaining using a rotated univariate Fourier basis to correct curtains along a specific angle. This problem is run on many small boxes of the image in order to reduce memory usage and computation time, and to account for the case where some curtains begin and end vertically. The second linear optimization model (**??**) reduces any boxiness left by the first optimization model. Finally, a third optimization model (**??**) corrects for contrast loss after the first two steps.

Curtaining Correction Optimization Model

The first step in our algorithm is the curtaining correction model. This step's input is an uncorrected 2D image and its output is an array of overlapping corrected boxes. For example, when solving with a box size of 10 pixel by 10 pixels and an overlap of 2 pixels, we solve the first 10-by-10 box, then move left or down by 8 pixels and solve the model again on those 10-by-10 pixels. These will be recombined into a single corrected image later. For the remainder of this subsection, "image" will refer to the one small piece ("block") of the overall image.

Rotatable Univariate Fourier Basis

In order to target curtains in a certain direction, we set up a rotatable univariate Fourier basis (visual-

ized in figure **??**):

$$F_\theta(x,y) = a_{\theta_0} + \sum_{i=1}^{N} \left(a_{\theta_i} \cos\left(\frac{2\pi i}{N}(\vec{u}_x x + \vec{u}_y y)\right) + b_{\theta_i} \sin\left(\frac{2\pi i}{N}(\vec{u}_x x + \vec{u}_y y)\right)\right) \tag{1}$$

where

- $F_\theta(x,y)$ is the multiplicative contribution to the change in brightness of the pixel at (x,y) for the given θ.
- $a_{\theta_0} \dots a_{\theta_N}$ are the Fourier co-efficients on the real Fourier terms
- $b_{\theta_1} \dots b_{\theta_N}$ are the Fourier co-efficients on the imaginary Fourier terms
- N is a parameter controlling the number of frequencies to which the image block is fit
- $\vec{u}$ is a unit vector perpendicular to the curtain's direction, θ. E.g. we define a completely vertical curtain as having $\theta = 0^\circ$ and $\vec{u} = (1,0)$

Any straight-line structure can be constructed as weighted sums of these sines and cosines. This allows the model to target only specific structures. Note that, for $\vec{u}$ with an irrational ratio $\vec{u}_x/\vec{u}_y$, it would be possible to overfit the data by using as many variables as there are pixels in the block, and even values of N approximately equal to the box width can produce "beat" patterns in which stripes are alied at different angles, see figure **??**. Extreme problems are shown in the 16th and 20th waves.

Optimization Problem Description

Taking advantage of the properties of the above Fourier basis, we set up the first linear optimization model. The model is computed in such a way so as to apply a multiplicative change to the image in order to account for situations where curtains are up against voids. In early tests of a similar but additive algorithm, curtains which terminated at voids were problematic. The following is a description of this optimization problem.

Within each box of pixels, we seek a corrected image, J, of the form

$$J(x,y) = I(x,y) \cdot \sum_\theta F_\theta(x,y) \tag{2}$$

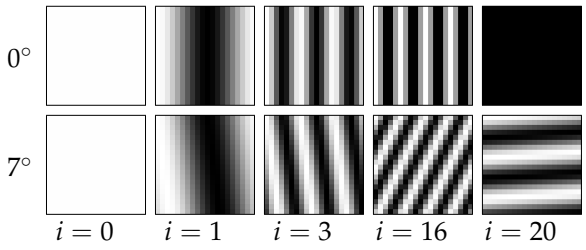


Figure 3: *A rotatable univariate Fourier basis is used to correct curtains in specific directions without affecting other structures. Visualized here with block size = 20px, $a_{\theta_i} = 1$ and $b_{\theta_i} = 0$. Higher frequencies behave differently than expected, creating what look like beat patterns, due to the sampling of a continuous function at discretized pixel co-ordinates. These frequencies can cause unwanted changes to the image by giving the optimization model the ability to cause changes to non-curtain structures. This can be mitigated by reducing N, the number of frequencies used to fit the data. Reducing the number of frequencies also reduces processing time, but also reduces effectiveness for thin curtains. In this paper, we use $N = 8$ frequencies.*

in which each pixel in the original image I is multiplicatively corrected. The differences between I and J are restricted by (**??**) to be a sum of waves parallel to the expected curtain directions. We expect that the true image has a lower variation in the direction perpendicular to the curtains, and that changes are not too large.

We express this trade-off in the form of an optimization problem

$$\min_{F_\theta} \sum_{(x,y),(x+1,y)\in box} |J(x+1,y) - J(x,y)| + \lambda \sum_{(x,y)\in box,\theta} |1 - F_\theta(x,y)| \tag{3}$$
$$\text{subject to } |1 - F_\theta(x,y)| \le \alpha$$

where

- λ controls the trade-off (with changes being discouraged by higher values) (see fig. **??**)
- α limits the maximum percent change of any pixel

The first term in the cost function provides the most important penalty: reducing the ℓ_1-difference

norm (i.e. the sum of absolute adjacent column differences) in the x-direction. Since rocking mill curtains are assumed to be along directions that are close to the vertical (i.e. $|\theta| \ll 45°$), penalizing horizontal variation is a good approximation to penalizing changes in the directions perpendicular to the curtains. The second term penalizes changes to the original image I. This term's importance relative to the first term is controlled by the experimental constant λ. Increasing λ will result in a lower difference between the original and corrected images. This can be adjusted by the user in order to improve image quality, by limiting changes to non-curtain pixels. In similar problems, image quality is usually not very sensitive to changes in λ. See figure **??** for a comparison of λ values.

Per-pixel constraints encode prior knowledge and assumptions about the dataset. A constraint of α is applied to limit each pixel's change. This value represents the maximum allowed percent change due to a curtain. A higher value for α may result in deeper curtains being corrected; however, it may also lead to an increased change of the image in undesired ways. Therefore, care should be used to set this value to the lowest value that gives adequate curtaining correction in order to minimize unwanted changes.

The linear optimization problem is solved on images in small, overlapping blocks, resulting in small optimization problems, which reduces memory requirements and increases throughput. Additionally, since the Fourier basis is univariate and therefore independent of the curtain's y-direction, solving in many small blocks helps to account for occasional variation along the curtain's direction, sometimes disappearing and reappearing entirely.

Although our aim was to correct curtaining at two angles, an arbitrary number of angles can be combined into the optimization problem to correct many angles at once should the instrument operator choose to apply many angles during a tomogram acquisition.

Box Normalization Step

Since the curtain correction optimization problem is solved in boxes independently of one another, small changes at the edges of boxes are noticeable. A second optimization problem is used to correct this.

Optimization Problem Description

To correct blockiness, the second optimization problem searches for the best constant value by which to multiply each block in order to normalize the difference between block edges. This optimization problem is similar in structure to the first optimization model. The aim of this optimization problem is to find a multiplier $m_{i,j}$ to correct the decurtained image J for each box from the first optimization problem:

$$K_{i,j}(x,y) = m_{i,j} \cdot J_{i,j}(x,y)$$

where we use indices (i,j) for the boxes, and cordinates (x,y) for pixels within each box.

The chosen $m_{i,j}$ values should minimize the difference in the average value of the overlapping sections:

$$\begin{aligned} \min_{m_{i,j}} \sum_{i,j=0}^{w,h} \Bigg(& \sum_{\substack{(x,y) \text{ in overlap of} \\ \text{boxes } (i,j) \text{ and } (i+1,j)}} \left| K_{i,j}(x,y) - K_{i+1,j}(x,y) \right| \\ & + \sum_{\substack{(x,y) \text{ in overlap of} \\ \text{boxes } (i,j) \text{ and } (i,j+1)}} \left| K_{i,j}(x,y) - K_{i,j+1}(x,y) \right| \Bigg) \\ & + \sigma \left\| K - J \right\|_{\ell^1} \\ & \text{subject to } \left| 1 - m_{i,j} \right| \leq \alpha \end{aligned} \tag{4}$$

where

- σ controls the amount of deviation from the curtaining corrected box mean, $\overline{I_{i,j}}$, and
- α limits the maximum percent change to each block (same value as that used before).

Recombination Step

In the next step, brightness-corrected boxes are combined into a single image, L. In order to further avoid box boundaries, a weighted combination of the overlapping boxes is used. The weighting of each pixel depends on its distance from the edge of the box in which it resides. If the overlap of the curtaining correction optimization problem is set to n, the minimum contribution weight is $\frac{1}{n+1}$ and the

maximum pixel contribution weight is $\frac{n}{n+1}$. A value of $\frac{2}{n+1} \dots \frac{n-1}{n+1}$ is then used for the pixels in between. For areas with four overlapping boxes, pixel values are similarly interpolated using bi-linear interpolation.

Brightness-Adjustment Step

The original curtaining correction step tends to have a bias towards 0, i.e. the brightest sections of the image are brought down, leading to a loss of contrast. This final step helps to fix this problem, increasing contrast and brightness of the top-level pixels.

Optimization Problem Description

This optimization model fits the curtaining- and box-corrected image to a quadratic function which aims to minimize the difference between the original image and the corrected one.

$$\min \sum_{(x,y)\in S} \Big| I(x,y) - (a \cdot L(x,y)^2 + b \cdot L(x,y)) \Big| \tag{5}$$

where

- S is a subset of pixels, we take the centres of the boxes used in the first step;
- I is the uncorrected image;
- L is the curtaining- and box-corrected image;
- a and b are the co-efficients of the quadratic function.

The quadratic function is then applied to the corrected image, L, to obtain the final image M:

$$M(x,y) = a \cdot L(x,y)^2 + b \cdot L(x,y),$$

for every pixel (x,y).

Implementation

Our algorithm was implemented in Ubuntu 17.04 using the Anaconda Python 4.1.1 distribution, which is based on Python version 3.5.2. The scikit-image image processing library (**?**) was used to read in our 8-bit image files as NumPy arrays (**?**) in order to perform calculations. The PuLP linear optimization library (**?**) was used to define each optimization model and call the GNU Linear Programming Kit (GLPK), version 4.62. GLPK is an open-source C implementation of primal and dual simplex methods used to solve linear programs (**?**).

Results

The proposed algorithm was compared to **?**'s algorithm and Avizo version 9.0's FFT curtaining removal wizard. The **?** and Avizo algorithms were run on the first 200 slices of the data. Our algorithm was used to process 1000x1000px arrays from selected slices. Slices were cropped to 972x972px for display. Identical regions were selected from each of the outputs and are compared side-by-side in figure **??**. Our algorithm was run using block size 20px and an overlap of 2px with the following parameters, described in the aforementioned optimization problems: $\alpha = 0.09, \sigma = 0.5, \lambda = 0, \theta_1 = 7^\circ, \theta_2 = -1^\circ$.

Canny Edge Detection

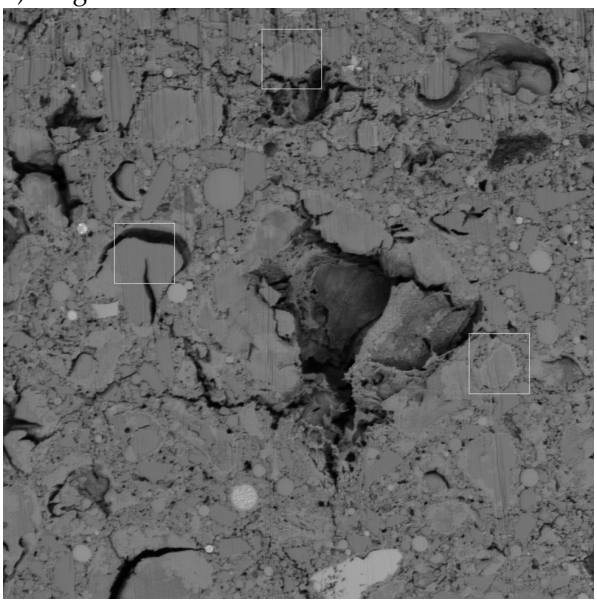
Figure **??** shows the result of the Canny edge detection applied to the blow-ups (figure **??**) of the images in figure **??**. Canny edge detection is an image gradient-based edge detection algorithm that uses a low and high threshold to determine whether a pixel is part of an edge. Otsu's method (**?**) is a way of automatically determining a good threshold factor, σ_b^2. The Python package scikit-image 0.12.3 was used to compute both the Otsu threshold and the Canny edge detection, using $\frac{1}{4}\sigma_b^2$ and $\frac{1}{2}\sigma_b^2$ as the low and high thresholds. These thresholds were selected to attempt to find a balance between finding subtle features and detecting noise. With these thresholds, if a curtain remains in the image, it will be detected by Canny edge detection as an edge. As a precursor to many segmentation routines, Canny edge detection is a good visual indicator of curtaining removal effectiveness.

Discussion

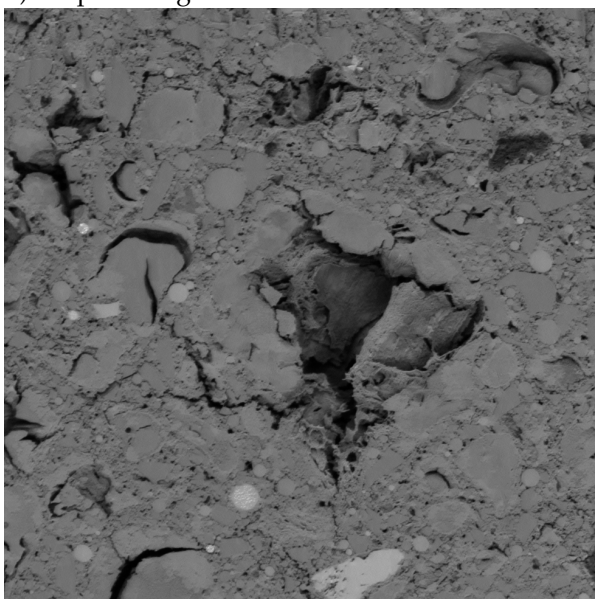
Choice of Block Size

The choice of block size determines the performance of the algorithm, both in terms of image quality and processing time. Internal testing found that if a block was the same size as features (for

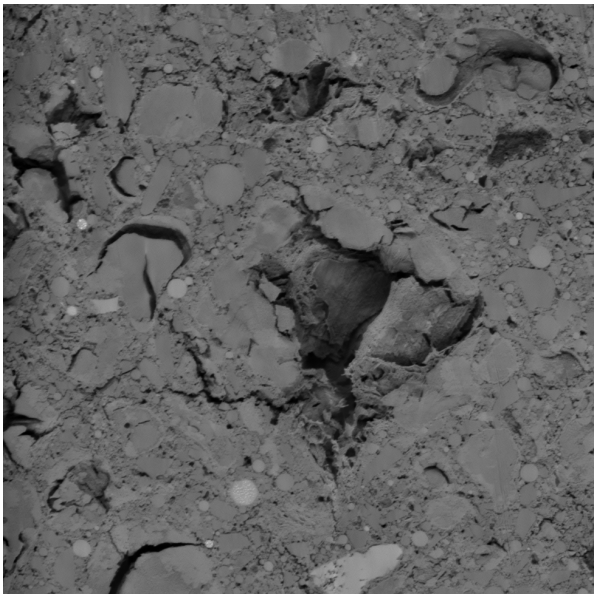
a) Original



b) Proposed algorithm



c) **?**



d) Avizo Curtaining Wizard

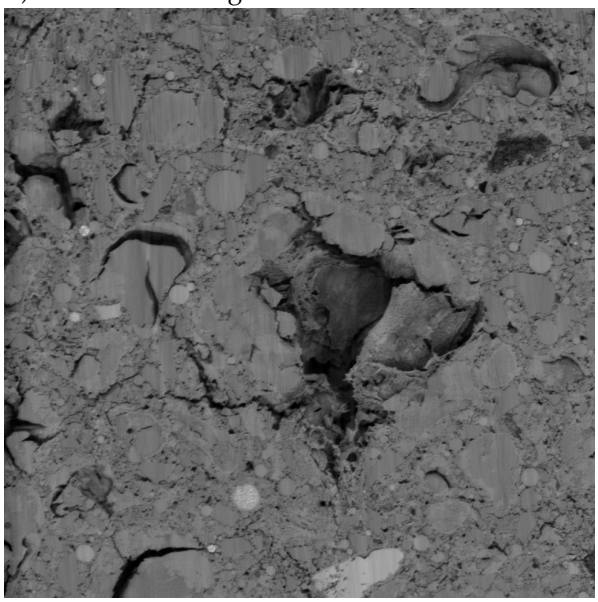


Figure 4: *A comparison of curtaining correction algorithms. a) is the original image of a pore in our concrete dataset, b) is the result of our proposed method (block size = 20px, overlap = 2px, N = 8, $\alpha = 0.10, \lambda = 2, \sigma = 0.5, \theta_1 = 7°, \theta_2 = -1°$), c) is the image processed by* **?***'s vertical curtaining algorithm, d) is the result of Avizo's FFT Curtaining Filter run once for each angle (97° & 89°– measured from the positive x-axis), using the default 3° tolerance setting.*

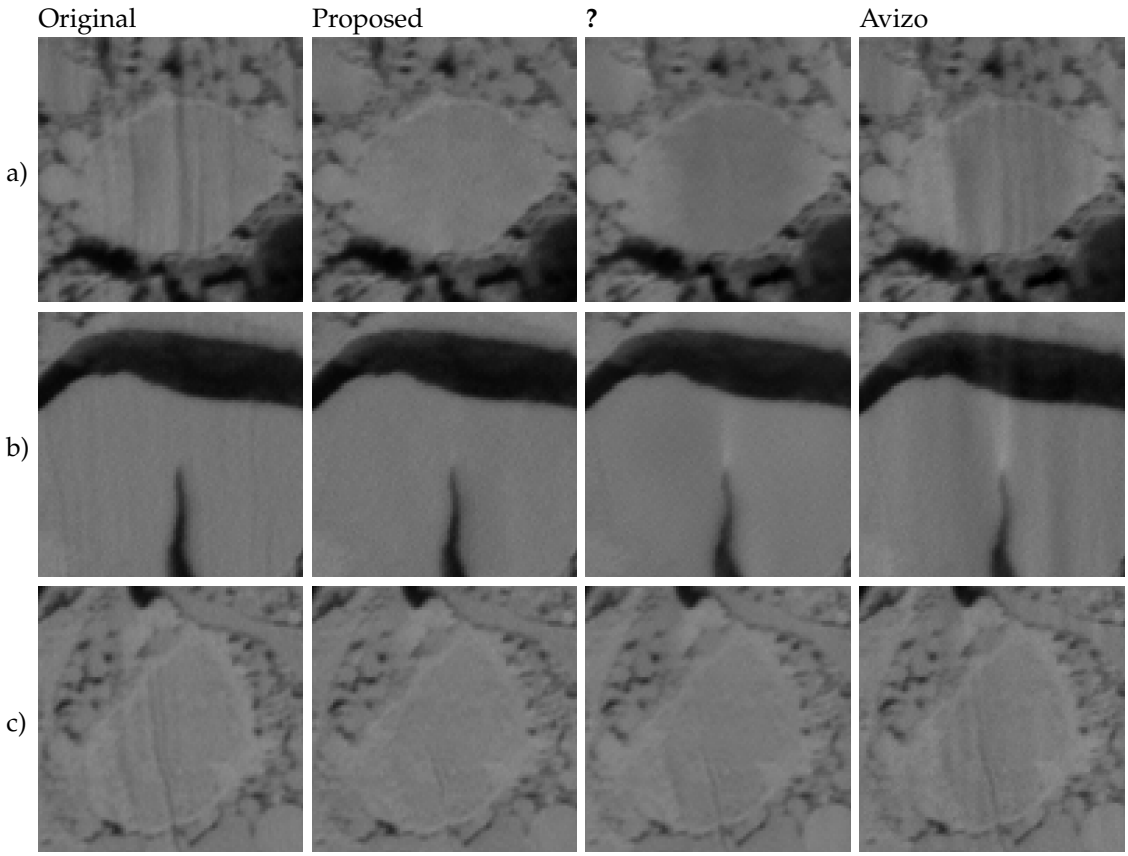


Figure 5: *A comparison of blown up sections shown in figure* **??** *a).*

example, the same width as the diameter of circular features), this can cause unwanted changes to the image because the optimization problem will attempt to reduce the contrast between the feature and its surroundings. In general it was found that increasing the block size reduced the number of unwanted changes to the image, however this also increases computation time as the linear optimization problems are bigger. Therefore, some experimentation is required to find a good block size, though 20px seems to work well.

Another advantage to smaller blocks is that although we assumed **A1**, it is true that there is some vertical deviation and changes of darkness, sometimes disappearing or reappearing altogether. Since this method lacks a physical model to describe these behaviours, keeping the block size small means that the vertical change across the small block is minimal and curtains are corrected effectively.

Choice of λ

The parameter λ controls the strength of the second penalty in the optimization problem described above. A larger value of lambda penalizes changes to the image, preserving structure and contrast from the original image. This allows the user control over image quality as desired, as the optimization problem can cause undesired changes if there is a vertical structure partially intersected by the optimization problem box. The user is free to select the value of λ which suits their goals, but there is a general method of selecting a best value called the L-curve method (**?**), representing an efficient trade-off between fitting the raw data and having the desired property, which in our case is a reduction in the ℓ_1-difference in the horizontal direction. Figure **??** shows the corrected images with five different values of λ, and the corresponding L-curve, plotting the ℓ_1-x-difference of the corrected image

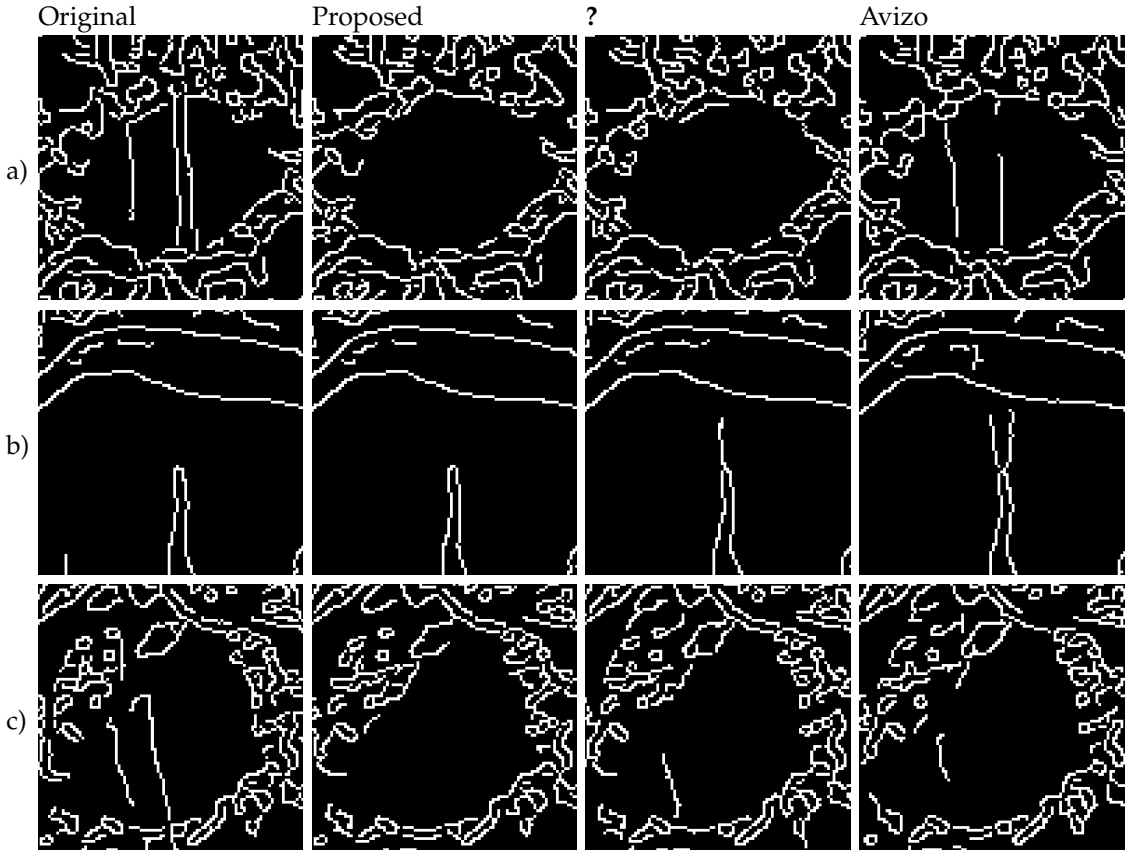


Figure 6: *A comparison of the Canny edge detection results in the blown up sections shown in figure* **??** *a).*

versus the logarithm of the mean energy of the correction.

Finding a value for λ is more or less an experimental procedure; however, a value in the 0-5 range was found to provide a good balance of preservation of image structures and good curtaining artefact correction.

Comparison to Other Methods

A comparison of our method to two other curtaining correction models is presented in figure **??**. The method presented by **?**, while intended for vertical-only curtains, was able to remove most of the curtaining artefacts; however, as a result of the fact that this method expects vertical curtains, it leaves behind what can be described as a lower-frequency waviness in the image, as well as occasionally imparting some extra, unwanted artefacts.

When filtered without aligned images, the method tends to lighten voids and cause more changes to the overall image, which is expected given that the method is 3-dimensional in nature and therefore requires aligned images to function properly. One advantage of our proposed model is the fact that it's a 2D technique, which means the dataset does not need prior alignment nor do we need information about prior or future slices to complete the filtering process. This also means our algorithm could be implemented in realtime when imaging on the microscope.

Perhaps more surprisingly, commercially available Avizo's FFT curtaining removal filter, while targeted at specific angles, imparted a similar, yet much more noticeable effect to the image, while also noticeably missing the curtaining effect in certain areas. Additionally, while the **?** method appears to avoid large changes to voids (when used properly with aligned images), the Avizo filter ac-

tually imparts the same low-frequency effect onto the darker void regions. While this implies that the Avizo method is perhaps more general-purpose, it is clearly targeting parts of the image we know should not change (**A5**). These kinds of changes are expected when using a traditional FFT filter they are known to potentially introduce additional artefacts when curtains do not cover the entire cross-section of the image, in this case due to the presence of voids (**?**).

Compared to these two methods, the method presented in this paper is able to effectively remove the curtaining effect without imparting this new, lower-frequency structure. However, our algorithm does have the tendency to reduce contrast of unintended parts of the image. This doesn't greatly affect image segmentation (see figure **??**), but can affect the image visibly to the human eye. Choosing a good value for λ can help to combat this, however the model should be explored further to attempt to mitigate this effect.

Curtains occasionally appear to deviate from their straight-line paths. However, our algorithm is still able to handle these situations fairly well. This is likely because our linear model is solved in blocks, meaning curtains tend to approximate straight lines well enough within those blocks.

Figure **??** shows some regions of interest from the images in figure **??**. In a), we see an area that is heavily affected by curtains in the original image. The proposed method removes these curtains very effectively, even across the boundaries of different phases of the image. **?**'s method is also effective at removing these, but produces a darkened area in the middle of the phase. Avizo's curtaining filter removes some of the curtains, but overall leaves most of the effect.

In b), there are some subtle curtains in the original image. Both our method and the **?** method are effective at removing the curtains in this location. However, a lighter artefact directly above the crack is created by the latter. This is likely an attempt by the filter to increase the brightness of the dark area, which is avoided by our method's multiplicative factor. Avizo's filter imparts this same artefact in both directions and to a greater extent, even causing a lightening of the void pixels, and transforms the curtains into shaded regions.

In c), the method proposed here causes some reduction in contrast around the edge of the phase shown. This can be seen in other areas where there are subtle differences between adjacent gray values. Additionally, the curtain slightly deviates here, violating the assumption **A1**, which causes those curtains to not be fully removed. **?** does a better job at preserving contrast but does not fully remove the angled curtain, since it is not designed to do so. Avizo also does a good job at preserving contrast, but does not fully remove the curtains either.

Future Work

Ongoing and future work is aimed at increasing the computational efficiency of the algorithm, taking advantage of the "embarrassingly parallel" nature of the curtaining correction optimization problem. Each block can be solved in separate threads, allowing a speedup roughly proportional to the number of CPU cores on the user's machine. Additionally, the Fourier basis matrix depends only on the experimental parameters (θ_i, N) . Therefore, it can be generated once and shared amongst the blocks.

The PuLP library was advantageous in implementation and testing due to its human-readable problem setup, ability to write direct linear program files for debugging purposes and its ability to interface with multiple solvers. However, its internal design makes it significantly slower than other libraries, especially when dealing with a series of many small linear programs. Therefore, it would be worthwhile to implement the algorithm using a faster library such as CVXOPT, which interfaces directly with GLPK's C library.

Additionally, future projects involve incorporating phase information to gain a better understanding of the formation and propagation of curtains, including "hoodoos". This information can be used to more intelligently target and correct the artefacts. New microscope control APIs allow programatically capturing information from multiple detectors. This additionaly information should help in the detection and correction of curtains.

Acknowledgments

The authors thank Yasamin Sartipi for her work in the early stages of this project, Dr. Kathryn Grandfield and

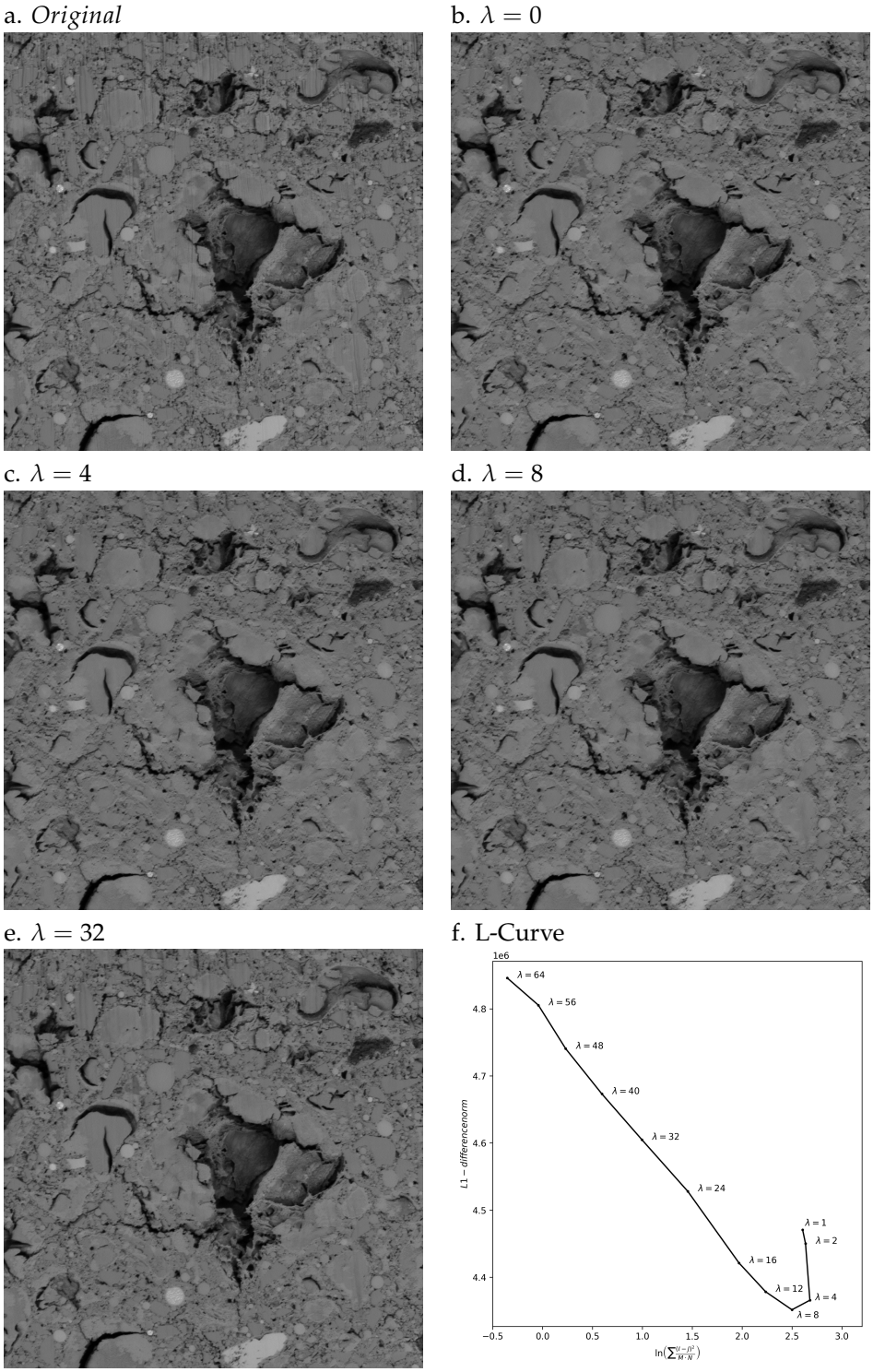


Figure 7: *a) through e) show the effect of a range of λ values. Higher values penalize changes to the image more heavily, resulting in an image which retains more of the original detail, but with the side effect of correcting curtaining artefacts less effectively. The value of λ can therefore be tailored to the application. f) Is an LCurve plotting the sum of the absolute values of x-direction differences (ℓ_1-difference) versus the mean-energy (ℓ_2) difference between the original and corrected image. This plot suggests that $\lambda \approx 8$ provides an optimal balance between curtain correction and changes to the image (block size = 20 px, overlap = 2px, $N = 8, \alpha = 0.10, \sigma = 0.5, \theta_1 = 7^\circ, \theta_2 = -1^\circ$)*

group for providing their support and use of the group's image processing computers, and the many people at the Canadian Centre for Electron Microscopy (CCEM) for their ongoing help and infinite technical expertise. We also thank Prof. Kay Wille of the University of Connecticut and his group for the UHPC test sample, Ron Kelley and Brandon Van Leer of FEI-Thermo Fisher for the rocking-stage PFIB image acquisition as well as Dr. Jan Henrik Fitschen and his supervisors for running their algorithm on our data and agreeing to allow us to include their results in this paper. This project was supported financially by the Natural Sciences and Engineering Research Council of Canada (NSERC) and the McMaster University Department of Materials Science and Engineering.