Drift Correction for Scanning-Electron Microscopy

by

Michael T. Snella

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Engineering in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2010

Author . . . .
Department of Electrical Engineering and Computer Science
September 13, 2010

Certified by . . . .
Karl K. Berggren
Associate Professor of Electrical Engineering
Thesis Supervisor

Certified by . . . .
Vivek K Goyal
Esther and Harold E. Edgerton Associate Professor of Electrical Engineering
Thesis Supervisor

Accepted by . . . .
Dr. Christopher J. Terman
Chairman, Masters of Engineering Thesis Committee

Drift Correction for Scanning-Electron Microscopy

by

Michael T. Snella

Submitted to the Department of Electrical Engineering and Computer Science
on September 13, 2010, in partial fulfillment of the
requirements for the degree of
Master of Engineering in Electrical Engineering and Computer Science

Abstract

Scanning electron micrographs at high magnification (100,000x and up) are distorted by motion of the sample during image acquisition, a phenomenon called drift. We propose a method for correcting drift distortion in images obtained on scanning electron and other scanned-beam microscopes by registering a series of images to create a drift-free composite. We develop a drift-distortion model for linear drift and use it as a basis for an affine correction between images in the sequence. The performance of our correction method is evaluated with simulated datasets and real datasets taken on both scanning electron and scanning helium-ion microscopes; we compare performance against translation only correction. In simulation, we exhibit a 12.5 dB improvement in SNR of our drift-corrected composite compared to a non-aligned composite, and a 3 dB improvement over translation correction. A more modest 0.4 dB improvement is measured on the real image sets compared to translation correction alone.

Thesis Supervisor: Karl K. Berggren
Title: Associate Professor of Electrical Engineering

Thesis Supervisor: Vivek K Goyal
Title: Esther and Harold E. Edgerton Associate Professor of Electrical Engineering

Acknowledgments

There are many people who have helped me make this project a reality and to whom I owe my deepest thanks.

First and foremost, I would like to thank my advisors Karl and Vivek, whose constant engagement and encouragements helped see me through to the end. You were two sides of the same coin, bringing together vastly different expertise to help tackle any challenges I had.

I would also like to thank my advisor András Vladár at NIST, who was as much a life advisor as a research advisor. Thank you for showing me that the historic Polish-Hungarian bond is as strong as ever.

A special thank you goes to Petr Cizmar, your help over that first summer springboarded my own research direction and your attentive answers to my questions helped keep me on the right track.

I would like to thank Mark Mondol and Jim Daley at MIT and Bin Ming at NIST, whose help in sample prep and instrument operation ensured that I had actual results to showcase and not just simulations.

I would like to acknowledge the help of the Nanometer-Scale Metrology Group at NIST, who generously took me on as a summer researcher and gladly showed me how to use their equipment, especially the Orion microscope.

Thank you to my labmates Donny Winston and Joshua Leu, for some great discussions and sanity checks early and late in the project, respectively.

Thanks to Sam Glidden, for knowing me all too well and being so damn dependable, making sure that I started writing this thesis on time.

And to all my friends, whose good natured ribbings about when I was actually going to graduate spurred me to eventually do so. To all of you, I say: Hah!

Last but certainly not least I thank my mother, who has served as an unwavering foundation from which I could always draw strength, even during the most trying times. I cannot express my gratitude to her and the rest of my family. Without their support I would not be.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The scanning electron microscope (SEM) is the workhorse of most nanotechnology laboratories in industry and academia. While it does not have the atomic-scale resolution of the transmission electron microscope, the ability to scan bulk samples with minimal sample preparation and the intuitively understandable images produced by the SEM make it a very practical general use instrument. High-resolution metrology conducted on the SEM is critical in semiconductor fabrication, where the critical dimension of transistors must be controlled [1], and for emerging methods of nanostructure fabrication such as templated self-assembly, which often involve sub-10 nm features [2]. However, the ability of the SEM to accurately render the surface topology of such nanostructures is complicated by sources of noise and degradations both internal and external to the instrument—lens aberrations and sample charging are the most significant effects caused by the physical process of scanning an image, while sample drift is the most significant external noise. Choosing optimal instrument settings will minimize lens aberrations, and sputter coating the sample with a conductor or other appropriate preparation will take care of sample charging, but short of additional vibration isolation equipment drift distortion is difficult to compensate. In this work we propose and develop a general-purpose algorithm to correct for drift distortion by creating a composite image from multiple quickly-scanned frames. The algorithm and imaging procedure is applicable to a wide range of imaging conditions on the SEM and similar scanning-beam microscopes.

1.1 Image Distortions in Scanning Electron Microscopy

A SEM works by probing a specimen with a beam of electrons and detecting the various products that are emitted from the specimen surface. In an optical microscope all of the information in the micrograph of the sample is acquired simultaneously. In contrast, SEM micrographs are created as the electron beam scans across the surface, with the result that portions of the image are generated at different times. Despite the different imaging mechanism, SEM micrographs are subject to errors that are generic to microscopy: SEMs suffer from chromatic aberration (differences in energy of the electrons in the probing beam), spherical abberation (gradients in the magnetic field used for focusing the beam), and diffraction, all of which impact image resolution. There are also distortions which are unique to scanning-charged-particle microscopy. One effect of the statistical nature of the electron-sample interaction is shot noise. The number of electrons impinging on the sample is a random process in time, which causes the number of products measured at a point in the sample to fluctuate. The solution is to spend more time integrating data at each point, performing a slower scan. However, the drawback of a long scan is two-fold. A longer scan time means more probing electrons are injected at each point in the sample, which charges the specimen and deflects subsequent electrons in the beam. Secondly, a slower scan increases the likelihood that during the scan the sample will move with respect to the spot where the beam is focused [3]. We refer to this motion as drift, and it degrades the quality of the acquired image by integrating mispositioned information at each point during the scan.

Drift is a non-linear, time-varying process that warps the features in an image and can cause blurring. Figure 1-1 provides a clear example of the effect of drift, with the sinusoidal waving along the vertical edge of the structure caused by a mechanical vibration coupling into the instrument, distorting the true position of the wall edges by upwards of 35nm. Mounting an instrument on an isolation stage does minimize the effect of vibrations coupling to the instrument (for example, the operator walking

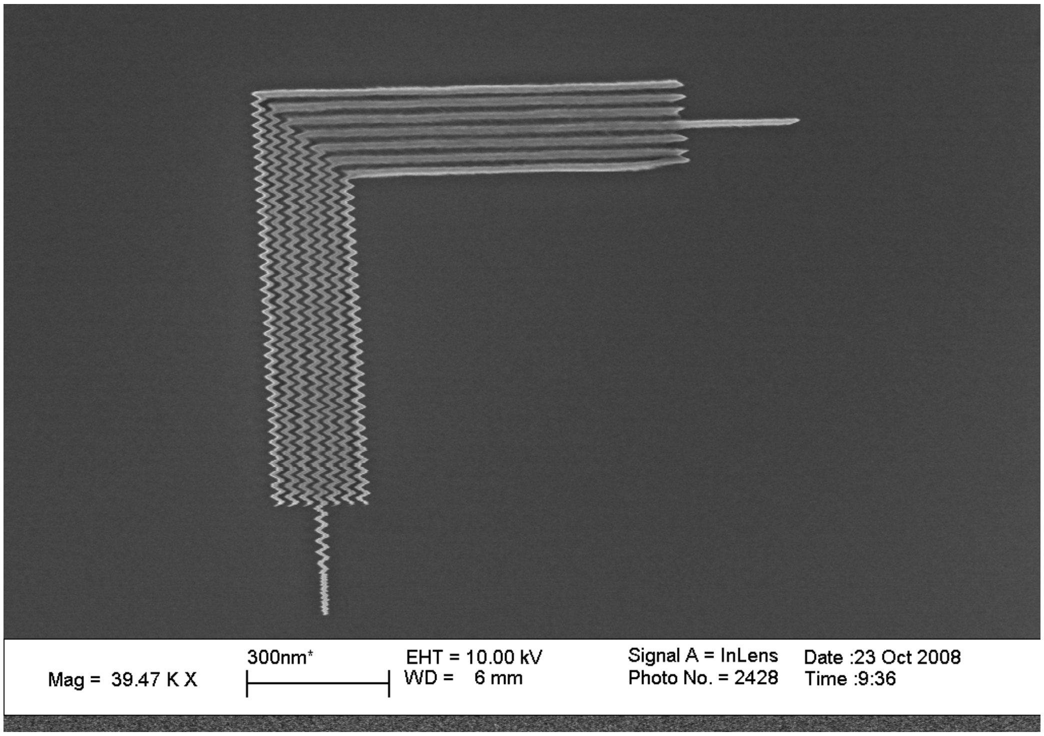


Figure 1-1: An example of drift-distortion in a scanning electron image. The distortion can be seen along the vertical edge of the sample. This very prominent effect was caused by a nearby cherry-picker [3].

across the room, or talking while the SEM is scanning), but other sources of drift are internal to the instrument itself. Mechanical settling of the sample-stage after positioning the specimen, thermal expansion caused by ohmic heating and fluctuations of the position control voltages all contribute to misalignment between where we expect to be scanning and the actual position of the probing electron beam of the sample surface. This form of drift introduces a time-dependent offset between images scanned in a sequence on a timescale of minutes [4], while mechanical vibration within the optics column leads to drifts which act on a timescale of seconds. The amplitude of drift in Figure 1-1 is atypical, with more typical effects being displacement on the order of <10 nm [5].

1.2 Overview of Proposed Solution

The problem we wish to solve can be summarized as follows: In order to produce good quality, non-noisy SEM images we need to perform a slow scan. The longer the scan, the more pronounced the effect of drift within the image. So is there a way to gain the benefits of a slow scan while minimizing the effect of drift within an image?

One solution typically implemented on more sophisticated instruments, such as scanning-beam-lithography systems, is to include a laser interferometer stage to actively adjust the position of the sample. Adding an interferometer is a costly investment for SEMs which are not used to write samples. Modern commercial SEM software allows users to average several scans together, either frame-by-frame or line-by-line, integrating the information from 2 to 16 scans to create a composite image. Since no alignment occurs during the averaging, the effect of drift is still present and manifests as a blurring of the image.

We propose a drift-correction method which aligns a series of quickly scanned images to produce a drift-free composite image. The images are scanned sequentially, one after another, and the drift affecting each image is estimated by comparing adjacent pairs of images. The main idea behind the correction method is illustrated in Figure 1-2. The images are spaced along the horizontal time axis, with the image axes vertical and out of the page. The real drift, drawn in green, will move the area being viewed in each image. Comparing adjacent images allows us to measure the shift between them and form an estimate of the drift, drawn in red. We use the drift estimates and information about the scan time for each image to correct the effect of drift in individual images, and combine the information in the corrected and aligned images into a composite. The composite contains information from the total scanning time of the series without drift-distortion, allowing us to retain the benefits of a long scan while producing an image of the specimen much closer to the ground truth.

The development of our correction method draws on previous drift-correction algorithms for scanning microscopy. We incorporate a simple model for drift which is used to correct individual images during the registration.

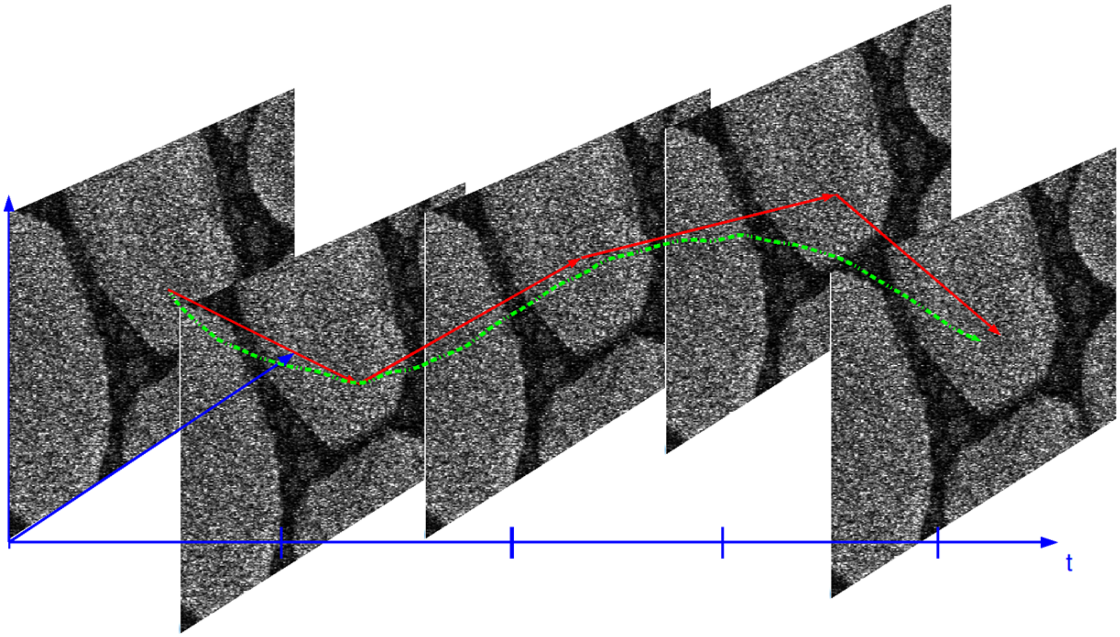


Figure 1-2: Illustration of the proposed correction between a series of images scanned in time. The true motion of the sample is illustrated in dashed green, while piecewise linear estimates are in solid red.

1.3 Background

1.3.1 Image Registration

Image registration is the process of aligning a set of two or more images, which were taken at different times, with different sensors, or from different viewpoints. The characteristics of the images to be registered and their particular deformations determine the most appropriate registration method. Images with pronounced features such as contours, edges, corners, or line intersections may be processed to detect them and have corresponding features matched between images. Alternatively, area-based methods may match similar areas between images without attempting to detect features, by comparing intensity values directly. Area-based methods are often used in medical imaging, where distinct objects or contrast is lacking. Due to the method of matching, these methods are often called “correlation-like methods” [6].

Area-based detection has been used previously in registering images acquired from scanning-probe microscopes, and are the core of the digital-image-correlation (DIC)

[7] method used on the atomic force microscope (AFM). The basis of the DIC is a cross-correlation performed between pairs of images. The normalized cross-correlation for images I and R, both size MxN, is

$$C(d_1, d_2) = \frac{\sum_m \sum_n I(m + d_1, n + d_2) R(m, n)}{\sqrt{\sum \sum [I(m, n)^2] \sum \sum [R(m', n')^2]}}. \tag{1.1}$$

The cross-correlation can be used to determine the translation between the tested image and a reference image. When images differ by translation, the peak of the cross-correlation occurs at the coordinates of the offset between them. This is because the cross-correlation is maximal when both images have identical values at corresponding image coordinates. This peak will occur even under conditions where some rotation and skewing are present, which distorts the purely translational correspondence between the images. Owing to the high spatial resolution of the AFM, DIC methods are used to evaluate local translations in the tested image caused by in-plane deformations on a strained sample [8]. Images are segmented and similar segments between the images are cross-correlated to calculate in-plane displacements, producing a displacement field over the surface of the sample.

After the geometric relationship between the modified and reference image is measured—translation, rotation, scaling, or some non-linear correspondence—this relationship is used to estimate the transformation to align the test and reference images. The transformation is a mapping function which maps the coordinates of the test image onto those of the reference image. This mapping function is estimated from a model for the measured distortions. For image values at non-integer coordinates, interpolation is used to compute the appropriate integer values.

1.3.2 Image Registration on the SEM

Sutton proposed a similar DIC based method to estimate and correct drift distortion within images from SEM-based strain measurements [4]. Sutton's method is based on an image model similar to the one derived in Section 2.3. For pairs of images, DIC is performed in the neighborhood of several pixel positions. Drift velocities are calculated from image acquisition times and integrated along the scan pattern to produce an estimate of the drift in time. This drift function is used to correct the drift distorted positions in the odd numbered images. Sutton demonstrated the performance of the drift correction on images between 200x and 10,000x magnification, corresponding to object magnifications ranging from 1.25 μm/pixel to 20 nm/pixel [9, 10]. For the highest magnification measurements, 16x frame averaging was used, for an acquisition time of 75 seconds per image.

Cizmar proposed an alternative method for compensating drift distortion in much higher magnification images, those typical used in nanofabrication [5, 11]. Cizmar's algorithm is used to process images typically taken with fields of view of 500 nm to 1 μm, corresponding to object magnifications of 1 nm/px to 2 nm/pixel. Dwell times were set to instrument minimums, typically 100 ns, for acquisition times of $\approx$100 ms per image, with no frame or line averaging—the final settings are a tradeoff between good image oversampling and low image acquisition time. Such rapid acquisition times minimize the effect of all but the highest frequency drift. The majority of drift occurs during the time between image scans, which can be removed by aligning images using a simple translation estimated by DIC methods. Only one global drift vector needs to be calculated per image pair.

1.4 Overview of Chapters

The remaining chapters are organized as follows. Chapter 2 describes the components of the SEM, and how an image is formed in the instrument. An analytical model for drift distortion is formulated which serves as the foundation for the proposed correction algorithm.

The drift correction procedure is described in Chapter 3. The correction algorithm is described step-by-step including implementation details for the shift estimation and image registration stages.

Chapter 4 describes the datasets and metrics used to evaluate the drift correction algorithm. We discuss the simulated images used in the development of the algorithm, as well as how samples were prepared for real datasets. We describe some common image quality metric to quantitatively evaluate algorithm performance. Finally, we detail the data acquisition process, namely the generation of artificial datasets using our simulation software, and the procedure for acquiring a dataset on the SEM. We discuss sample degradations that occur during measurements in the SEM.

In Chapter 5 we evaluate the performance of the drift correction algorithm with both simulated and real datasets. The performance of the proposed correction method is investigated by parametric sweeps with simulated datasets. Several datasets measured on different scanning beam instruments are corrected and we examine and compare the corrected composite images.

Chapter 6 summarizes our results and what was accomplished in this project. Possible directions for further investigation are outlined and improvements to the proposed algorithm are discussed.

Chapter 2

Mechanics of Image Formation

This chapter describes how a scanning electron microscope image is formed. We begin by describing the components of the microscope. Next we describe the contrast mechanism of a secondary electron image. Lastly, we develop a model for the image under drift distortion and examine the case of linear drift.

2.1 SEM Schematic

The SEM consists of three main sections: an electron gun, which emits electrons either through thermionic or field emission; a set of electro- and magneto-static lenses, which focus the electrons emerging from the gun into a beam which strikes the surface of the sample; and the lower chamber, which houses both the sample-stage to which the specimen is mounted and detectors for measurement of the specimen. The sections are illustrated in Figure 2-1.

2.1.1 Electron Gun

The upper portion of the SEM houses the electron gun, which is the source of the electron beam used to image the sample. The beam is formed from electrons emitted from an electron source. Two mechanisms are responsible for electron emission - thermal emission (TE) and field emission (FE). In thermal emission the source, typically

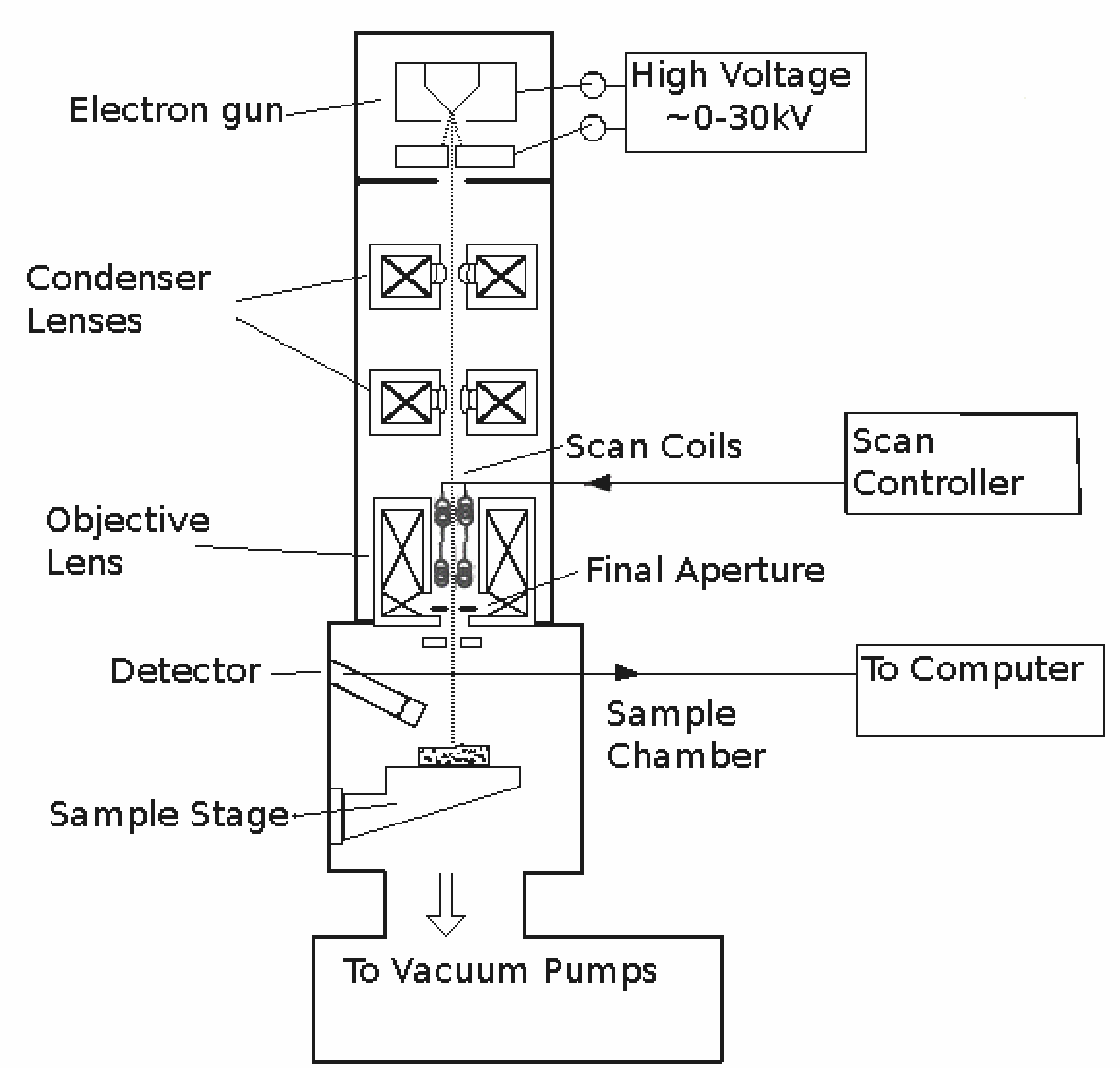


Figure 2-1: Diagram of a Scanning Electron Microscope. The three main segments are illustrated: the electron gun with electron source, the focusing column with electron optics, and the sample chamber with stage and detector.

a tungsten filament of LaB_6 tip, is heated directly or indirectly allowing electrons to overcome the work function of the material. In field emission an applied electric field lowers the potential barrier at a sharp tungsten tip enough to allow electron to tunnel through and escape. The source is held at a negative potential (typically between -0.1 kV and -30 kV) and emitted electrons are accelerated towards a grounded anode plate below the source to form a collimated beam. The electron source is one factor of the maximum instrument resolution. After passing through several sets of lenses, the electron beam is focused onto a small area of the sample. The diameter of this electron-probe spot depends on the brightness of the source, β, defined as current density/solid angle. By conservation of brightness [12, 13] we find the minimum diameter of our probing beam

$$D_0^2 = \frac{\sqrt{4I/\pi^2\beta}}{\alpha}, \tag{2.1}$$

where I is the current of our electron beam, and α is the angle of convergence between the edge of the final lens and the probe spot. The advantage of a brighter beam is that for a fixed spot size and angle of convergence, a brighter source allows more current which creates more detectable signal - FE sources are typically two orders of magnitude brighter than TE sources. This minimum probe size is what we can achieve with perfect lenses, but as is often the case, aberrations from the electron lenses will limit what is practically achievable.

2.1.2 Optical Column

After the collimated electron beam exits the electron gun, it passes through and is focused by one to three electromagnetic lenses. These first condenser lenses demagnify the cross-section of the beam by a factor of 10 to 100. The demagnified beam is focused onto a small spot on the sample surface by the objective lens. The lenses are solenoids encased in large ferrous casings, which create a magnetic field in the path of the beam [12]. The geometry of the lens is such that the magnetic field causes electrons on the outside of the beam to be deflected and spiral towards the center. Unlike optical lenses, the focal length of electromagnetic lenses is adjusted by varying the strength of the magnetic field, so changing the current in the solenoids adjusts the focus of the final beam spot. The position of the spot is controlled by a set of deflection coils, which deflect the beam further before the objective lens. These coils are controlled by a scan controller outside of the instrument, which generates the scanning pattern used to image an area of the sample. The amount of deflection caused by the coils determines the magnification of the final image—a smaller deflection results in a smaller area being displayed in the measured image.

2.1.3 Sample Chamber

Underneath the SEM column is the sample chamber. The sample is mounted on a motor-controlled stage connected to the chamber. The stage motors are used to adjust the sample at low magnifications, and the stage can be tilted to improve signal collection by the secondary electron detectors within the chamber. The sample is firmly secured to the stage in order to provide a path to ground for the electrons deposited by the probe beam. This sample current is one type of signal collected in the SEM. Secondary electrons and other products of the beam-sample interaction are collected by an in-chamber Everhart-Thorley Detector [13]. More advanced SEMs also include an in-lens SE detector as well as a chamber mounted detector, which affords an increased image signal while decreasing noise, the mechanics of which are discused below.

2.2 Electron-Sample Interaction

Now that we have discussed how the probe beam is generated, we describe how the interaction of the primary, beam electrons allows us to make measurements of the sample.

As the highly energetic primary electrons enter the sample they undergo elastic and inelastic scattering as they interact with the atomic elements of the sample. Some of these primary electrons are scattered backwards and reemitted from the

Material	L_s [nm]
Au	2
Au-Pd	1.5
Si	6
C	12

Table 2.1: Characteristic range for Secondary Electrons in materials examined in this thesis [12].

sample, called backscattered electrons (BSE). As BSE are preferentially scattered by heavier elements they are detected to create a composition map of the sample. The primary electrons also scatter inelastically as they travel through the sample, creating secondary electrons. Secondary electrons (SE) are loosely bound outer shell electrons that gain sufficient energy during the inelastic collision to overcome their binding energy. All electrons emitted from collisions with primary electrons with energy below 50 eV are considered SE [14, 13]. Their low energy means that only those SE created near the sample surface will escape to be detected.

Figure 2-2 illustrates the emission mechanism for both SE and BSE. Secondary electrons generated at the incident beam are labelled SE1, while those emitted by backscattered electrons travelling through the sample are labelled SE2. Backscattered electrons that strike another spot in the chamber release more secondaries, SE3, not shown in the figure. The characteristic range of SE, L_s drawn in red, is material dependent. Table 2.1 lists L_s for several common specimen materials. Because of their limited range, SE1 provide information about the topology of the sample. The secondaries SE2 and SE3 emitted by backscattered electrons are undesirable because they increase the spacial region from which the signal originates.

While there are several other products that can be detected from the beam-sample interaction, SE are the most commonly measured signal in scanning electron microscopy. The photograph-like contrast of SE micrographs is familiar to us and they can be understood intuitively, leading to their widespread use.

The contrast of a SEM image can be understood geometrically. As the incidence angle of the beam with the surface increases, the section of the primary beam trajectory within L_s of the surface also increases. At the extremum when the beam

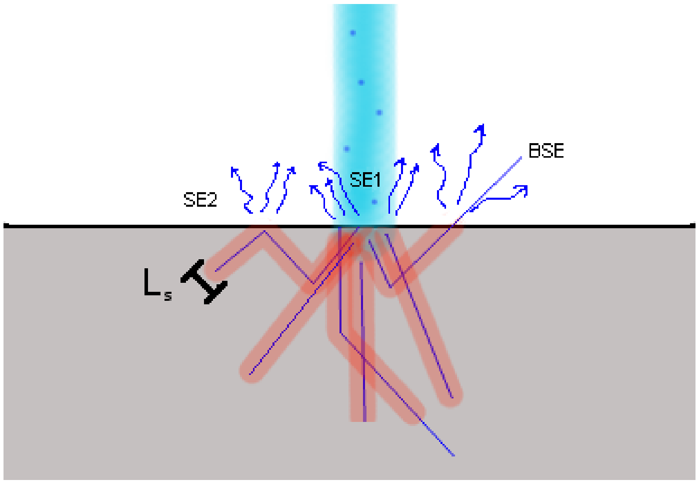


Figure 2-2: Interaction of the electron beam and sample near the spot position. The red regions indicate the extent of secondary electrons generated along the path of primary and backscattered electrons, dawn in blue within the sample. Secondary electrons generated at the probe-spot (SE1) escape the sample and are detected. Secondaries generated when backscattered electrons pass close to the surface away from the probe-spot (SE2) are also detected.

trajectory is parallel to the surface, secondaries generated along the entire path will be able to escape. The increase in the yield of SE is described by

$$\delta_{\mathrm{SE}}(\theta) = \delta_0 \sec(\theta), \tag{2.2}$$

where θ is the angle between the beam and surface normal, and δ_0 is the yield of SE when the surface is perpendicular to the beam [13]. Equation 2.2 suggests that sharper sloped sample regions will appear brighter than flat regions due to the larger number of SE emitted. The sharp increase in SE yield along vertical edges explains why sidewalls and thin features on the order of L_s appearing much brighter than their surroundings in SE micrographs. Figure 2-3 shows the relationship between the measured pixel value and changes in the surface height which generates contrast in the image.

When motivating this contrast mechanism we assumed perfect detection of SE. Factors such as detector position and the angular distribution of emitted SE will influence the contrast by decreasing the number of SE detected [13, 14]. However, for

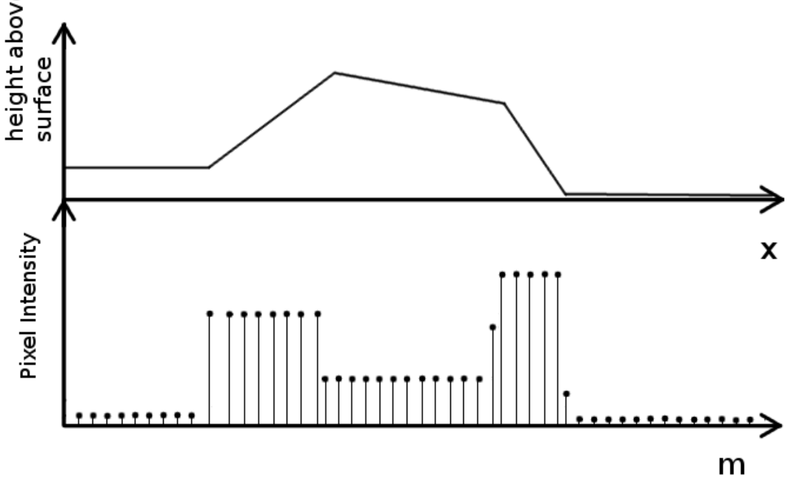


Figure 2-3: Changes in height along the sample surface (top) lead to a difference in pixel values in a row (bottom) due to enchanced SE emission along areas of higher slope.

our purposes the basics of SE contrast are sufficient to continue.

2.3 Image Formation Model

When collecting an image, the probe-spot sweeps across the surface of the sample in a raster pattern, as illustrated in Figure 2-4. This pattern is composed of a series of horizontal line-scans across the sample, with a fast return sweep occurring when the beam reaches the end of the horizontal scan range. As the beam returns it moves by a small vertical step, and begins to sweep the next line. The area bounded in red in Figure 2-4 is called the field of view, which is the region of the sample that is scanned in one image.

The line-scan is done as either a continuous sweep—in which case the signal from the detector is binned at fixed intervals in time—or the scan has discrete steps with the probe-spot remaining stationary at one point for a set dwell time, which we call t_{dwell}, before stepping to the next position. In either case, the resolution of the image in pixels determines the number of bins or steps in a linescan and the pixel values are the detector signal measured for that row. The final image is then an intensity map of detected electrons at these discrete steps.

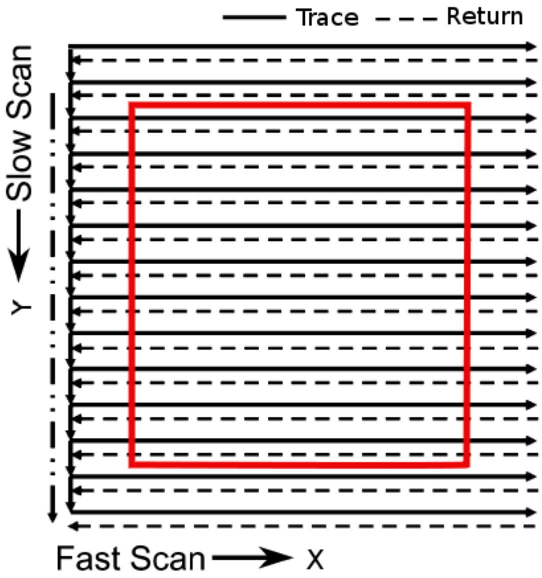


Figure 2-4: The raster pattern traced during an image scan. The scan begins in the upper left corner and ends in the lower right. The pattern of the electron beam across the surface is composed of rapid horizontal scans and vertical steps, which can be thought of as a slow vertical scan. The area being imaged, marked in red, is typically smaller than the area scanned by the electron beam.

2.3.1 Image Coordinates and Time Index

In order to understand how to correct for drift distortion, we need to understand how it affects our image. Before we can do that, we must define our image and sample spaces. Let's assign a coordinate system to the area we are scanning on our sample. We'll call this our sample coordinate, expressed as $(x, y)^T$, where x is the horizontal and y is the vertical position of points on the sample surface. Since the surface of our sample is continuous, so is our sample coordinate. Similarly, let's define the image or pixel coordinates, expressed $[m, n]^T$, where m and n are integer values for the horizontal and vertical pixel position, respectively. Figure 2-5 details the notation convention for a typical sample and corresponding image.

In addition to the image coordinates assigned to each pixel, pixels are ordered by the raster pattern traced by the probe-spot across the sample surface. If we couple this ordering with additional time information about the image we can use a time index to alternatively identify each pixel. We write the expression for the time index

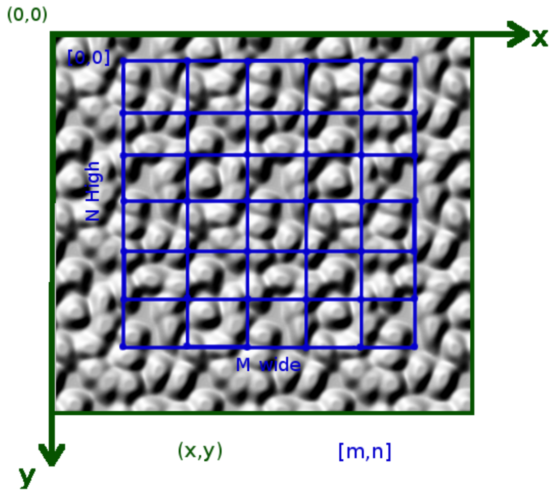


Figure 2-5: Image and sample coordinates superposed over the same area. The origin of both coordinate systems is located in the upper left with axes increasing to the right and downward.

of a pixel as:

$$t_{\text{index}}[m,n] = m\, t_{\text{dwell}} + n\, t_{\text{row}}, \qquad (2.3)$$
$$t_{\text{row}} = t_{\text{line}} + M\, t_{\text{dwell}},$$

where t_{dwell} is the dwell time per pixel mentioned before, t_{row} is the time to scan one row of the image, which includes the delay for the beam to reset between rows, t_{line}, and M is the width of an image in pixels. The index time represents the time at which the information for the pixel intensity begins to be acquired, so t_{index} is a monotonically increasing discrete function in m and n.

For a series of same sized images, acquired sequentially, the time index is similar, with an additional term specifying the image being indexed:

$$t_{\text{index}}[m,n,\ell] = M\, t_{\text{dwell}} + n\, t_{\text{row}} + \ell\, t_{\text{frame}}, \qquad (2.4)$$
$$t_{\text{frame}} = N\, t_{\text{row}} + \Delta_{\ell}$$

where t_{frame} is the time between the start of the scan of two sequential images, and ℓ is the image number (0,1,2,...) in the series. t_{frame} is the time to scan the N rows of an image plus a factor Δ_ℓ that includes any additional time between images, for example the time for the beam to reset between scans. Starting at 0, the time index measures the total time the instrument has spent scanning an image (or series), up to pixel $[m, n]$. We make the reasonable assumption that t_{dwell} and t_{line} are constant, since they are fixed by the SEM scan-controller. While t_{frame} often varies, we assume it is fixed for the remainder of the chapter.

The expression for image coordinates as a function of time is not as simple as the time index. We need to select a proper quantization from continuous time to discrete pixel positions. We use the convention chosen for our time index, identifying pixels by the time at which we start acquiring information for that pixel. One such set of expressions for m and n of image ℓ in a series is:

$$n_\ell(t) = \left\lfloor \frac{t_\ell}{t_{\text{row}}} \right\rfloor \tag{2.5}$$

$$m_\ell(t) = \left\lfloor \frac{t_\ell - \left\lfloor \frac{t_\ell}{t_{\text{row}}} \right\rfloor t_{\text{row}}}{t_{\text{dwell}}} \right\rfloor \tag{2.6}$$

$$t_\ell = t - \ell\, t_{\text{frame}} \tag{2.7}$$

where the floor function, $\lfloor\cdot\rfloor$, rounds down to the nearest integer, and the subscript ℓ denotes a parameter in image ℓ. These functions look like a staircase, with intervals of time corresponding to each pixel.

2.3.2 Drift

Now that we have an expression for where we are in time, we can relate our pixels with physical points on the sample in time. In order to derive this expression between image and sample coordinates, we will first consider the ideal case of no drift. With no drift, the field of the image does not change its position in time with respect to the sample. A pixel does not have a dimension of length, but we can use the field of view and pixel dimensions to calculate the step size between pixels as we scan the sample. Alternatively, in modern SEMs this step size is a programmable parameter.

In either case, the relation between the coordinate systems is time-independent:

$$x = m\, d_x, \qquad y = n\, d_y, \tag{2.8}$$

where d_x and d_y are the distance on the sample between adjacent horizontal and vertical pixels, respectively. If we wish to trace the scan across the surface as the beam moves from pixel to pixel, we replace m and n in (2.8) with the time indexed positions from (2.5) and (2.6). This gives us a time indexed scan of the surface of the sample, which will be identical for all images in a series (so we drop the subscript):

$$x(t) = m(t)\, d_x, \qquad y(t) = n(t)\, d_y. \tag{2.9}$$

Now let us consider the case with drift. How do we define drift with our coordinates? If the sample stage moves with respect to the optical column, the position of the probe-spot will change, resulting in a displacement of the field of view. This displacement is what we call drift. Let's write this displacement as a time-dependent vector in x and y:

$$\vec{\mathrm{Dr}}(t) = \begin{bmatrix} \mathrm{Dr}_x(t) \\ \mathrm{Dr}_y(t) \end{bmatrix}. \tag{2.10}$$

Drift is a continuous function, and as the field of view moves, the probe-spot will trace a complicated pattern as it scans across the surface sample. With drift there is no longer a one-to-one correspondence of image and pixel coordinates, and we need to keep track of the absolute time each pixel was scanned:

$$x(t) = m_\ell(t)\, d_x - \mathrm{Dr}_x(t), \qquad y(t) = n_\ell(t)\, d_y - \mathrm{Dr}_y(t). \tag{2.11}$$

This drift-distorted scanning pattern is illustrated in Figure 2-6, with the drift vectors drawn for several pixel positions.

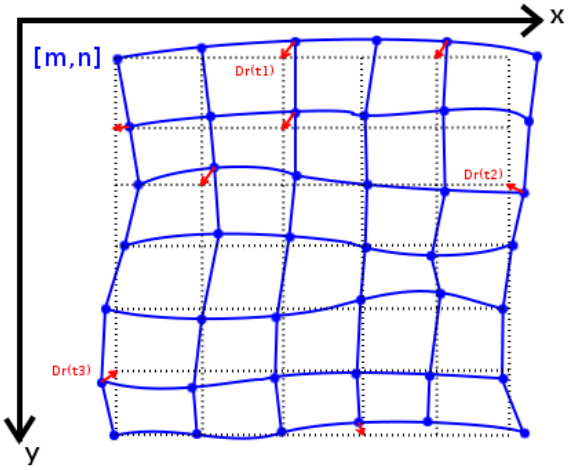


Figure 2-6: Example of the effect of drift on the relationship between image and sample coordinates. The drift vectors marked at times t_1, t_2, and t_3 exhibit the time-dependence of both drift magnitude and direction. The intersections on the dotted grid show where the image coordinates would be without drift. Drift warps the cartesian grid of pixel positions, resulting in a distorted image.

2.3.3 Estimating Drift Between Adjacent Images

We cannot know the drift for a single image without knowledge of the surface of the sample. Unless we know the exact position of objects on the sample surface, we have no basis for comparison. If we have a series of images, we can estimate the drift that occurred between pairs of images.

For instance, let's say we scan a point P at $(x, y)^T$ to the pixel at $[m_\ell, n_\ell]^T$ at time t in image ℓ, and in the next image we find point P at pixel $[m_{\ell+1}, n_{\ell+1}]^T$ at time index t'. Using (2.11) and $x_\ell(t) = x_{\ell+1}(t')$, since P does not move on the sample, we have:

$$x_\ell(t) = x_{\ell+1}(t'), \tag{2.12}$$

$$m_\ell(t)\, d_x - \mathrm{Dr}_x(t) = m_{\ell+1}(t')\, d_x - \mathrm{Dr}_x(t'), \tag{2.13}$$

$$\mathrm{Disp}_{\ell x} = [\mathrm{Dr}_x(t') - \mathrm{Dr}_x(t)] = d_x\,(m_{\ell+1}(t') - m_\ell(t)), \tag{2.14}$$

where $\mathrm{Disp}_{\ell x}$ is called the drift displacement. The calculation for the vertical displacement $\mathrm{Disp}_{\ell y}$ follows the same logic.

2.3.4 Constant Drift Velocity

Having defined drift with respect to our image and sample, how does drift distort our image? The case of linearly increasing drift - where the field of view moves at a constant velocity in some direction—is very illustrative, so let's walk through this example.

Let's assume drift in the form $\vec{\mathrm{Dr}}(t) = [v_x\, t, v_y\, t]^T$. For a positive v_x, (2.14) tells us that $\mathrm{Disp}_{\ell x}$ will be positive, meaning sample features occur at later pixels in the second image, and our field of view is drifting to the left. Rearranging (2.11) and plugging in our drift,

$$x(t) + v_x\, t = m_\ell(t)\, d_x, \tag{2.15}$$

if both $m_\ell(t)$ and $v_x\, t$ are increasing, the result is $x(t)$ increasing more slowly than in the case of no drift. So for a positive drift velocity, features in the x-direction are spaced a bit further apart in the resulting image. The spacing difference between objects on adjacent pixels a time difference t_{dwell} apart is $v_x\, t_{\mathrm{dwell}}$. An entire row is scanned in t_{row}, so at the start of the next row scan, the field of view will have shifted by $v_x\, t_{\mathrm{row}}$. After scanning the first line, the second line begins to be scanned a time t_{row} later. This results in a shearing of the image to the right, illustrated in Figure 2-7. If we instead have a positive v_y drift, our field of view moves upwards. The spacing difference between adjacent pixels in a column is now $v_y\, t_{\mathrm{row}}$, and the shear between columns is $v_y\, t_{\mathrm{dwell}}$.

In summary, the resulting image will have:

$$\Delta_x = v_x\, t_{\mathrm{dwell}}, \qquad \Delta_y = v_y\, t_{\mathrm{row}}, \tag{2.16}$$

$$\mathrm{sh}_x = v_x\, t_{\mathrm{row}}, \qquad \mathrm{sh}_y = v_y\, t_{\mathrm{dwell}}, \tag{2.17}$$

where Δ is the increase in spacing, and sh is the shearing. Since v_x and v_y are

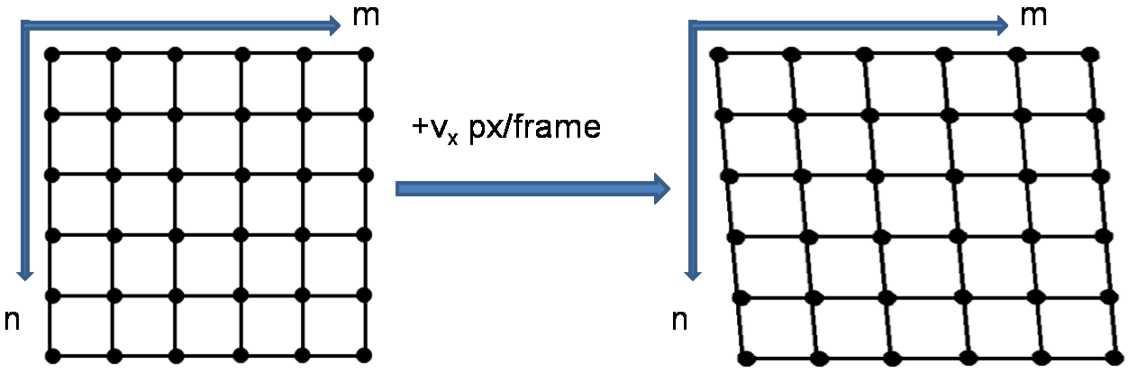


Figure 2-7: Effect of a positive drift velocity on the image of a grid. The intersections are spaced further apart in m, and subsequent lines are shifted over as n is increased.

unknown, an estimate using Disp from (2.14) is used.

$$\Delta_x = \frac{\mathrm{Disp}_x\, t_{\mathrm{dwell}}}{t_{\mathrm{frame}}}, \qquad \Delta_y = \frac{\mathrm{Disp}_y\, t_{\mathrm{row}}}{t_{\mathrm{frame}}}, \tag{2.18}$$

$$\mathrm{sh}_x = \frac{\mathrm{Disp}_x\, t_{\mathrm{row}}}{t_{\mathrm{frame}}}, \qquad \mathrm{sh}_y = \frac{\mathrm{Disp}_y\, t_{\mathrm{dwell}}}{t_{\mathrm{frame}}}. \tag{2.19}$$

Because of the large difference between t_{dwell} and t_{row}, shearing will be dominant in x and stretching in y. We expect lines running up and down in the image to be tilted due to drift, and the distance between lines running left-to-right to widen under drift.

Chapter 3

Proposed Correction Method

This chapter describes the image registration algorithm at the core of the drift correction method. We start with an overview of the flow of information in the algorithm and discuss the different operations.

3.1 Algorithm Implementation and Procedure

The proposed drift correction method is applicable to high magnification SEM images, between 100,000 to upwards of 1 million times magnification. For our application, we work at object magnifications of 1 nm/px and fields of view between 500 nm and 1 μm, depending on the image size. The method is based on Cizmar's method, with attention paid to improve the robustness of the drift compensation via a novel transformation model, while relaxing the dwell time requirement during image acquisition.

Figure 3-1 illustrates how the images in a sequence, which we call frames, are processed by our algorithm. A reference frame is loaded and prepared, with subsequent frames following step-by-step:

1. The next frame, called the test frame, is loaded.

2. The frame is cropped, windowed and filtered.

3. Test and Reference frames are rotation corrected.

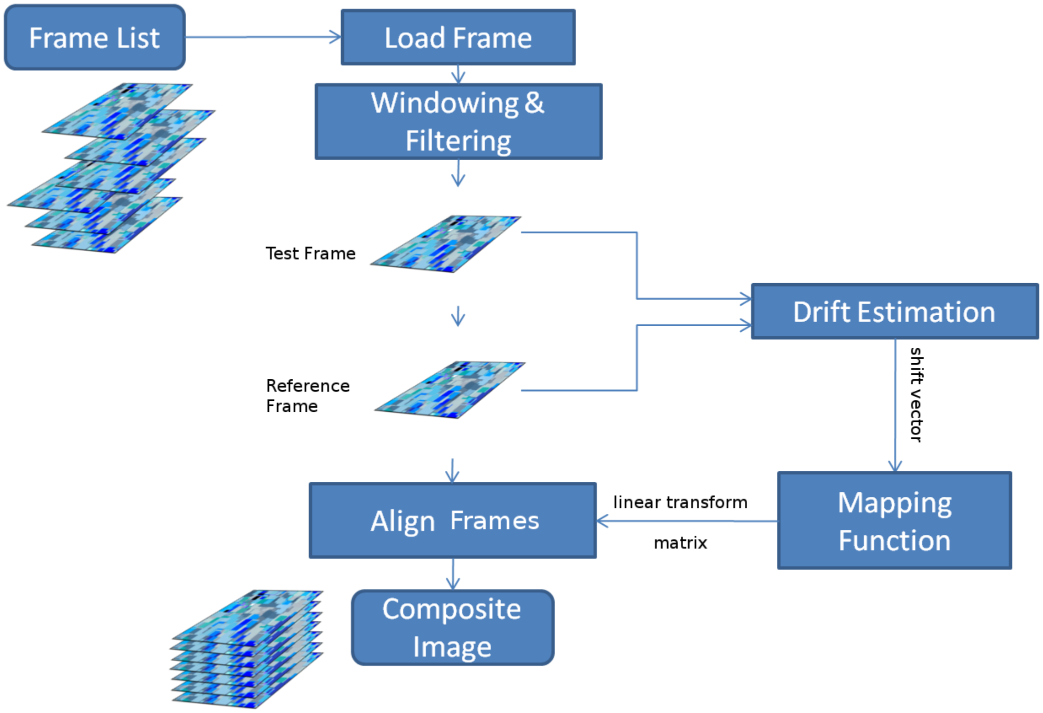


Figure 3-1: Structure of the algorithm. Arrows indicate flow of information between computation blocks. Pairs of adjacent frames are processed in series to create an aligned, composite image.

4. Test and Reference cross-correlated to calculate a shift vector.

5. The shift vector is used to estimate the mapping function to transform the test image onto the reference image coordinates.

6. The test frame is transformed and aligned with the reference, and both are added to the composite stack.

7. The composite stack is summed and averaged to produce a composite image.

The algorithm was coded in MATLAB®, with a focus on modifiability over processing speed. The procedure is called from a script, which allows for alterations and modifications to the procedure between datasets, or selective execution of individual segments of the algorithm. The script is composed of basic MATLAB commands as well as special functions created to implement each block operation of Figure 3-1. The script and functions created are included in Appendix A.

3.2 Image Preparation

After images are loaded they are filtered and windowed. For the extent of this project we limited our investigation to 512x512 images, but alternative, non-square sizes may be used. Following cropping, images are windowed with a Tukey window in order to eliminate edge effects during subsequent registration steps. The windowing is performed spatially by element-wise multiplying the image with a mask.

Depending on the severity of noise in the image, a variable low pass filter may be applied. As mentioned previously, the pixel intensity is based on secondary electron emission from the probe spot, which may be described by a Poisson process. For images with short dwell times, the signal intensity scales as t_{dwell}, while noise scales as $\sqrt{t_{\text{dwell}}}$ [15, 14]. A minimal dwell time is desirable in its ability to limit drift distortion within an image, but it will reduce the signal to noise ratio of the image. In order to reduce the impact of noise on our drift detection, we low pass filter our images with an adjustable sized finite impulse response (FIR) filter. FIR filters are commonly used when working with images—they are easy to implement as matrices of coefficients and are inherently stable. We use the windowing method described in [16] to create a circularly symmetric Blackman filter. A 5x5 pixel filter is used for lightly noisy images, but the filter size is adjustable for noisier images. Figure 3-2 shows the Tukey mask and filter kernels.

3.3 Rotation Estimate

The rotation between the reference and test frames is calculated after windowing and filtering. A rotation may occur if there is a change in the magnetic field strength during the scan, corresponding to a change of focus for the probe-spot. While modern SEMs often automatically correct for rotation caused by focal changes, slight drifts in instrumental focus are known to occur, and a rule of thumb is to check and refocus if necessary every fifteen minutes. One method for detecting the rotation estimate is proposed by Vanderwalle [17], using the property that identical but rotated images

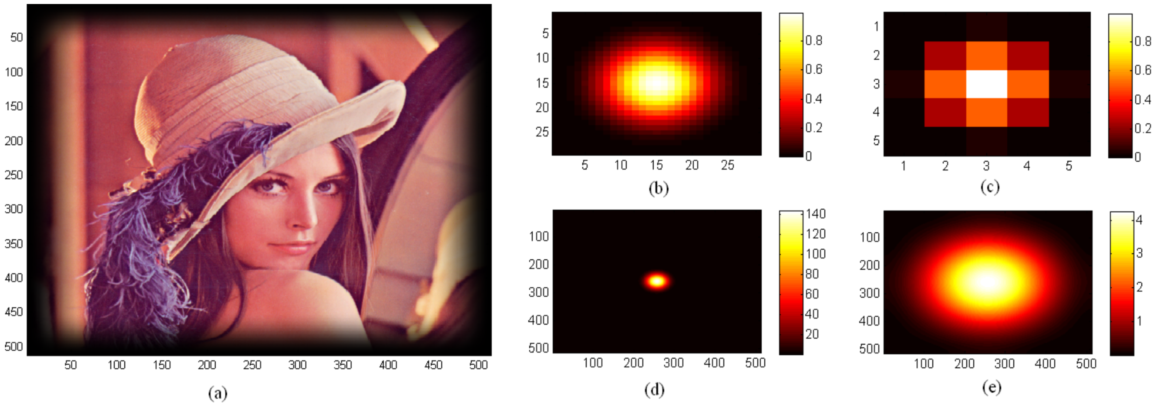


Figure 3-2: (a) Tukey Window applied to the standard test image Lena. The 29x29 pixel Blackman filter (b) and 5x5 pixel filter (c) with their 512x512 point Discrete Fourier Transforms in (d) and (e), respectively. There is a very visible difference in the concentration of energy in (d) and (e), representing the duality between the spatial and frequency domain. The larger spatial area of (b) spans a much smaller area in frequency, with the converse true in (c).

have identical but rotated Fourier transforms.

Following [17],we compute the Fourier transform of our images in polar coordinates, and integrate along angles α to get a one-dimensional function,

$$h(\alpha) = \int_{\alpha-\Delta\alpha}^{\alpha+\Delta\alpha} \int_{0.1\rho}^{\rho} |F(r,\alpha)|\, dr\, d\theta, \tag{3.1}$$

where ρ is half the frame size. Because we deal with a discrete signal, we realize (3.1) as wedges between $\alpha + \Delta\alpha$ and $\alpha - \Delta\alpha$, with $\Delta\alpha = 0.1$deg, and steps in α of 0.2 degrees. We exclude the very large and coarsely sampled low frequency content. With $h(\alpha)$ for both images the rotation can be calculated from their one-dimensional cross-correlation. Figure 3-3 demonstrates calculated rotation angles between pairs of SEM images.

3.4 Drift Vector Estimation

Once the reference and test frames are corrected for rotation, the translation between them is calculated by cross-correlation. Examining the form of the cross-correlation (3.2) reveals a similarity to the convolution operation in (1.1). A convolution in

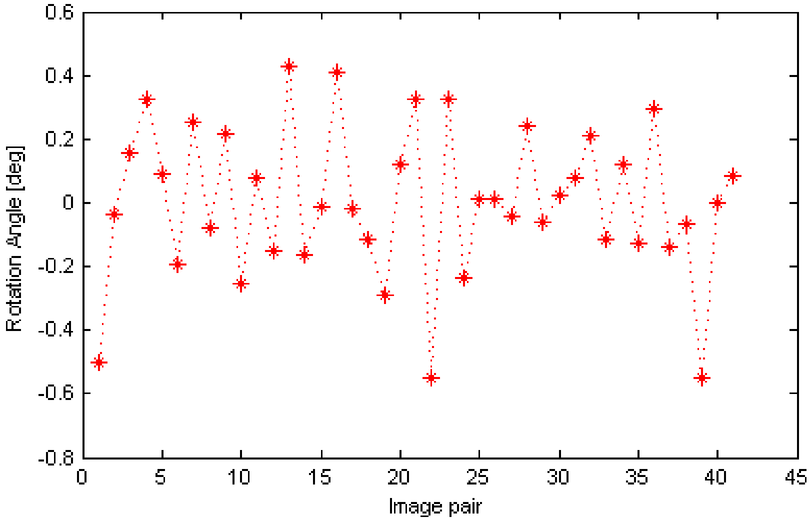


Figure 3-3: Angles calculated for 43 pairs of SEM images using the 1D correlation method. Changes in rotation fluctuate around zero through the series.

space is equivalent to multiplication in the frequency domain, and we take advantage of this property of the Fourier transform to efficiently compute the cross-correlation, where in (3.4) we use the property that an index negation $z(-m,-n)$ corresponds to a complex conjugation $Z^*(f_1,f_2)$ in the frequency domain. We use MATLAB's 2D Fast Fourier transform (FFT) operation, fft2, to perform our computation in the frequency domain, and the inverse operation, ifft2, yields the spatial cross-correlation. Care must be taken to prevent aliasing caused by the cyclical nature of the Discrete Fourier transform. One method to prevent this is zero padding the FFT to length $M+L-1$, where M and L are the sizes of each image along each axis [18].

$$CC_{t,r}(d_1,d_2) = \sum_{i=m}^{M-1}\sum_{n=0}^{N-1} t(m+d_1,n+d_2)r(m,n) \tag{3.2}$$

$$= x(m,n) * z(-m,-n) \tag{3.3}$$

$$\overset{\mathcal{DFT}}{\leftrightarrow} X(f_1,f_2)Z^*(f_1,f_2) \tag{3.4}$$

We can imagine the cross-correlation as an operation on the "center of mass" of each frame, where the concept of an image center of mass is analogous to the physical concept, with pixel values taking the role of mass. In this way, the result can be seen

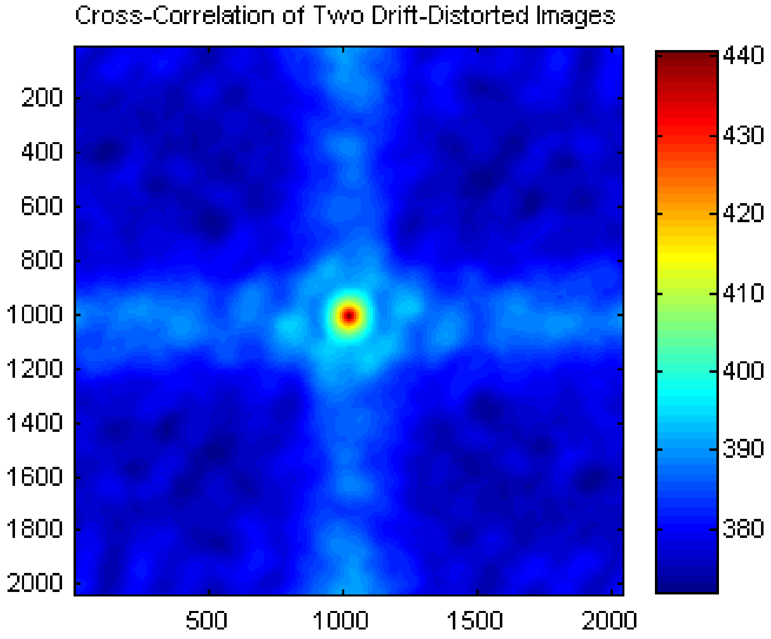


Figure 3-4: The cross-correlation between a pair of drift-distorted images, with the origin located in in the center. The position of the peak is the best estimate for the translation between both images.

as the distance between the centers of the frames. Figure 3-4 illustrates a typical result of cross-correlation.

The coordinates of the peak of the cross-correlation define the shift vector between the two frames. Because of the sampling that occurs when the SEM image is acquired, this peak can only determine the translation between the two images to an accuracy of one pixel. In order to improve the accuracy to sub-pixel estimates for the shift, we expand the two frames in order to perform the correlation across more points. The expansion is done by bilinear interpolation, where the values for the inserted interstitial pixels are determined by averaging their nearest neighbors. We interpolate 512x512 sized frames to 2045x2045, a nearly fourfold pixel increase. The finer coordinate spacing means we can estimate down to 0.25 px shifts, with additional interpolation allowing finer estimate granularity.

Figure 3-5 shows shift vectors calculated from 22 frames. The peak of the cross-correlation is determined by searching for the maximum value. The smoothness of the cross-correlation between two similar images ensures that there is only one global

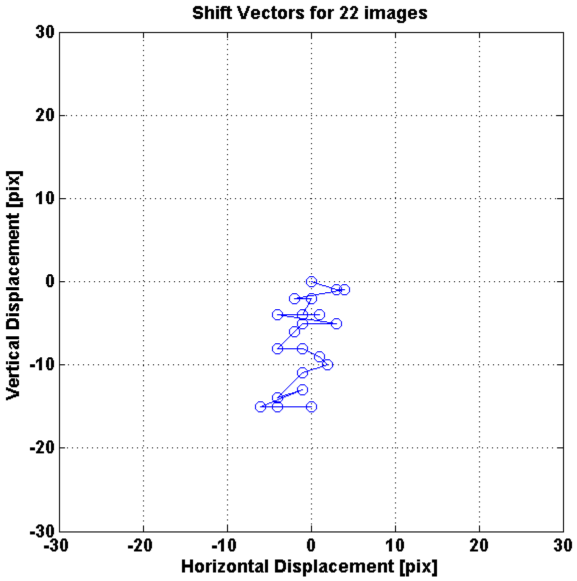


Figure 3-5: Estimates for the shifts between 22 adjacent pairs of frames. The path traced corresponds to the shift between the first and subsequent frames in pixels.

maximum, which corresponds to the desired shift.

3.5 Alignment

An assumption we make for our drift correction is that the underlying real drift is a smooth function in time. The physical pieces of the SEM which cause drift cannot have step functions in their differential motions in time. For our purposes, we approximate this smooth function as a series of piecewise linear shift vectors, which combine with the frame time information for piecewise constant drift velocities.

The equations for distortion under constant drift velocity derived in Section 2.3.4 showed that the sample and image coordinates are linearly related. This type of linear coordinate transformation is called an affine transformation. The types of affine transformations are illustrated in Figure 3-6.

We represent affine transformations using homogenous coordinates, which map the image coordinate 2-vector into a 3-vector: $[x, y]^T \rightarrow [x, y, 1]^T$. With this coordinate change the affine transformation is implemented as a matrix operation. Operating on

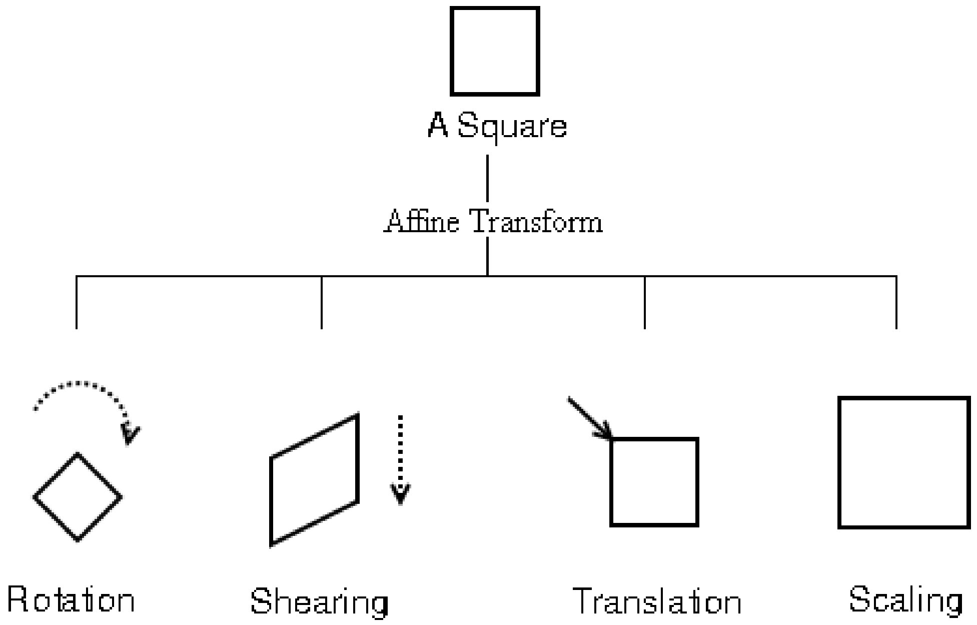


Figure 3-6: The effect of the different types of affine transformation on a square. All four categories are used in our image correction method.

the coordinate vector using the 3x3 matrices:

$$R = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}, \; S = \begin{bmatrix} 1 & \mathrm{sh}_x & 0 \\ \mathrm{sh}_y & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

$$T = \begin{bmatrix} 1 & 0 & \delta_x \\ 0 & 1 & \delta_y \\ 0 & 0 & 1, \end{bmatrix}, \; D = \begin{bmatrix} \Delta_x & 0 & 0 \\ 0 & \Delta_y & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{3.5}$$

results in all four types of transformations illustrated in Figure 3-6. The matrices R, S, T, and D correspond to a rotation, shear, translation, and scaling respectively. The factors for shear, translation, scaling and rotation are given by elements sh,δ,Δ along x and y, and angle θ respectively. So a shear and translation operation would be $[x, y, 1]^T \rightarrow ST\,[x, y, 1]^T = [x', y', 1]^T$. The distorted coordinates $[x', y', 1]^T$ can be remapped back to the original $[x, y, 1]^T$ by another affine transformation with estimates for the parameters of the correction matrices in (3.5). Intuitively, we expect

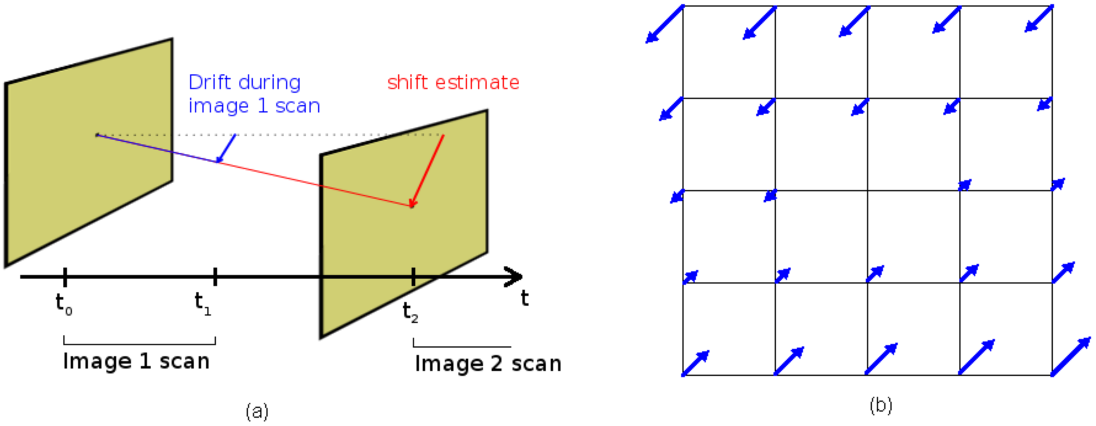


Figure 3-7: (a)The shift vector between adjacent frames. For estimating shear and scaling, we only want to correct for drift that occured during the image scan, and not during time spent between frames. (b)After removing the translation between frames, pixels further from the center need additional corrections to adjust for the effect of linear drift, namely shearing and scaling.

that negating the translation $\delta_{x/y}$ will remove it, (multiplying our correction matrix with negated translation does indeed return an identity matrix) and with a little more effort we can calculate a shear correction matrix using the shears $\mathrm{sh}_{x/y}$. Similarly, inverting $\Delta_{x/y}$ corrects for scaling.

The correction transformation begins with an estimate for θ as described in Section 3.3; the rotation is corrected before we begin to estimate the remaining transform coefficients. The transformation coefficients $\Delta_{x/y}$ and $\mathrm{sh}_{x/y}$ were given in (2.18) and (2.19), and $\delta_{x/y}$ are the components of the measured shift vector, $\vec{\mathrm{Disp}}$ (2.14). Figure 3-7 illustrates how the correction matrix is estimated from the shift vector and time information.

Chapter 4

Experimental Setup

This chapter describes the datasets used to evaluate our drift correction method. We discuss how simulated datasets were generated, as well as how we obtain timing information for the datasets and how we protect against several phenomena which degrade image quality during measurement.

4.1 Choice of Datasets

The number of images that can be processed at one time is only limited by the amount of drift. Ideally, as long as there are corresponding objects or areas within all of the images, a shift-vector may be obtained. In practice, once the field of view drifts so far as only half of the original area is scanned, any more images added to the dataset provide marginal returns. For a stable stage, datasets may number in the hundreds of images, or as few as a dozen, if taken right after sample loading when the instrument is still settling.

Both simulated and real datasets were used to evaluate the performance of the proposed drift correction method. The development of the algorithm was aided tremendously by the availability of simulated datasets with fixed noise and drift parameters. The open-source software Artimagen, created by Petr Cizmar at the National Institude of Standards and Technology in Maryland, was used to generate the simulated datasets [19, 20]. An example of a simulated image is given in Figure 4-1.

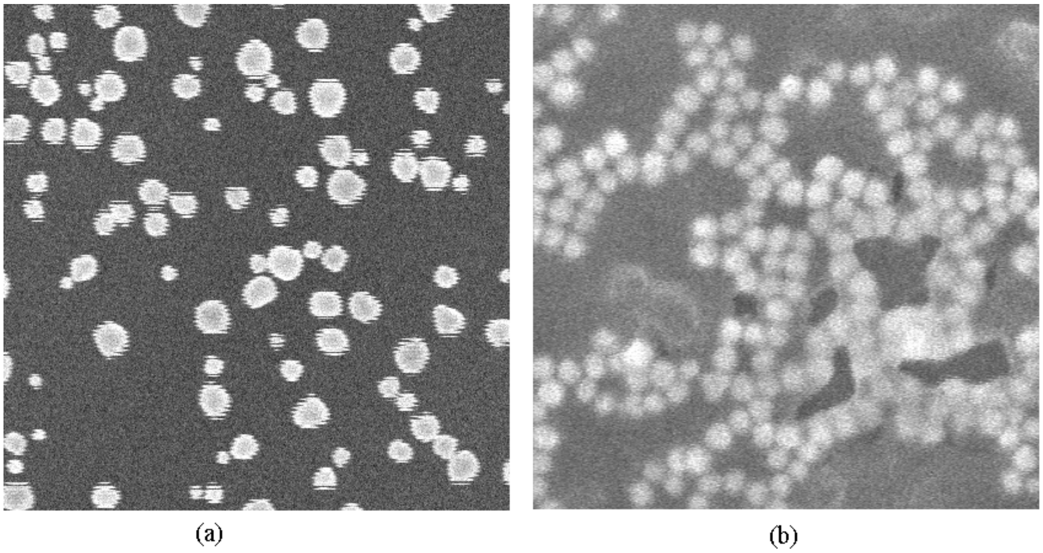


Figure 4-1: (a)Example of Artimagen generated SEM image of gold nanoparticles on carbon substrate, with 10 kHz vibration creating drift. Generated using Artimagen v.0.91 (b) Real image taken at the Nanostructures Laboratory at MIT. 100,000x magnification.

4.1.1 Simulating SEM Images Using Artimagen

Artimagen is an open source library for the C programming language that allows researchers the ability to create artificial images that have characteristics similar to images acquired on scanning charged-particle microscopes, such as the SEM or Scanning Helium-Ion Microscope. The source code may be downloaded at `http://artimagen.sourceforge.net`. Figure 4-2 shows how a simulated image is constructed by the program step-by-step. The final images emulate several characteristics of real SEM images, such as increasing pixel intensity with surface gradient, gaussian blurring of features caused by the finite extent of the probe-spot, and image graininess caused by shot noise of the SE emission. Images are also distorted by drift, causing warping of structure perimeters and misaligned edges between rows.

Artimagen implements drift by taking the values for each pixel in an undistorted image and adding a perturbation to their m and n coordinates $I' \begin{bmatrix} m \\ n \end{bmatrix} = f \begin{bmatrix} m + \Delta m \\ n + \Delta n \end{bmatrix}$, where Δm and Δn are linear combinations of sinusoids or other per-

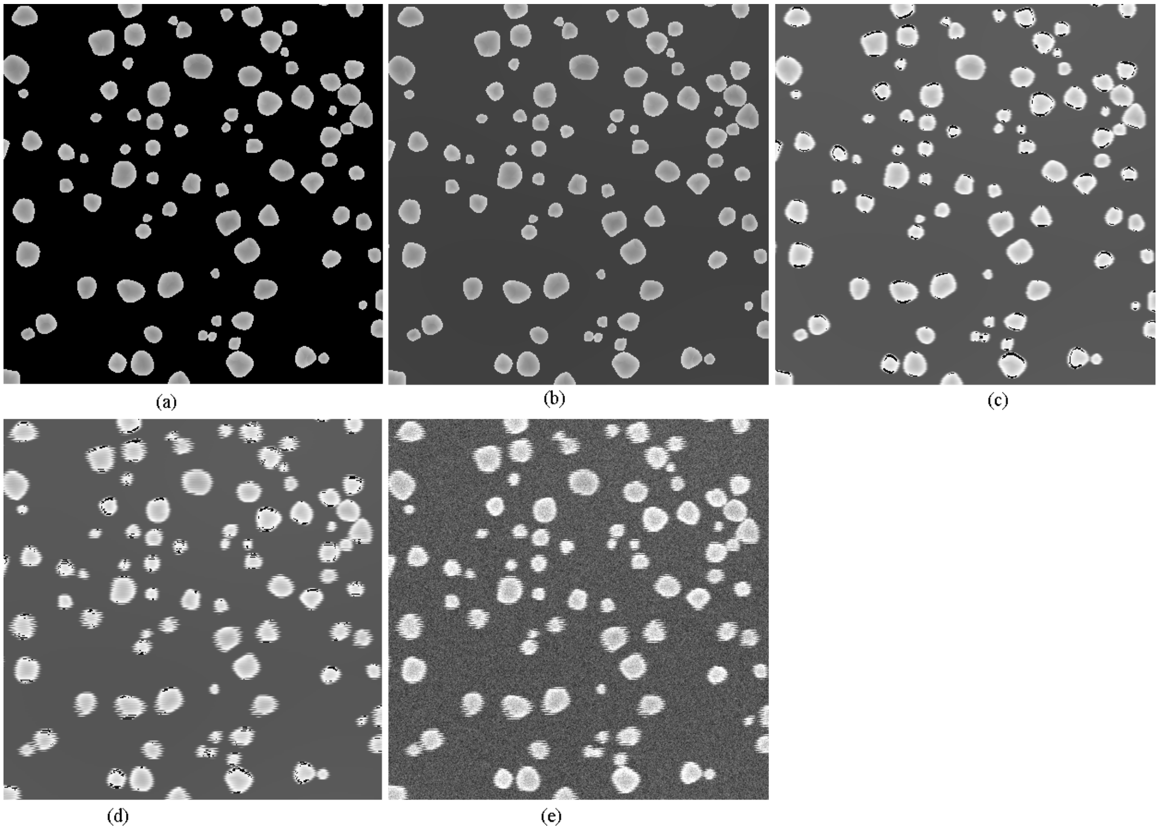


Figure 4-2: Artimagen creates images in layers: (a) the sample is first painted on, (b) and a background is applied, (c) the width of the probe-spot is simulated by convolving the iamge with a gaussian kernel, (d) a drift function remaps pixel positions to distort image features, (e) noise is applied to simulate shot-noise [19].

turbing functions. For our purposes, we care about drift that will shift of the field of view between images. We estimate this drift by large amplitude, low frequency perturbations. In order to approximate a uniform drift, we choose frequencies that will be below one-quarter wavelength over the time to scan the image. The images are cropped to 512x512 and saved as 8-bit tiff files. Appendix B includes the C code used to generate a dataset.

4.1.2 Estimating Time Information in Datasets

The compensation for shear is only as good as the available time information for the image. Three times are important—t_{dwell}, dwell time, t_{frame}, delay between images, and t_{line}, the delay between the end of one linescan and the start of the next as the beam is repositioned. For simulated datasets this information is chosen when applying drift distortion to the dataset, but for datasets acquired on the SEM it must be estimated. The time between images can be estimated by carefully recording the start and stop times during acquisition, and the dwell time is a user adjustable parameter on modern SEMs. The time t_{line} is much more difficult to estimate. In the author's experience, scanning electron microscope manufacturers are reluctant to disclose such instrumental parameters.

In order to get a rough estimate for t_{line}, an oscilloscope was used to capture the control signal sent to the x and y deflection coils in a Hitachi 5500 model SEM, located at NIST. Figure 4-3 shows the trace for one period of the x (horizontal) deflection for a scan with a dwell time of 100 μs. Note the settling time for the beam return before the next linescan, demonstrating the behavior illustrated in Figure 2-4. Assuming an equivalent behavior near the end of the row (the beam overshoots before returning and settling), t_{line} is estimated to be about 30% of the time spent scanning the row, $t_{\mathrm{dwell}}N$. A similar measurement performed one a faster scan yields an estimate of 18% of the linescan time.

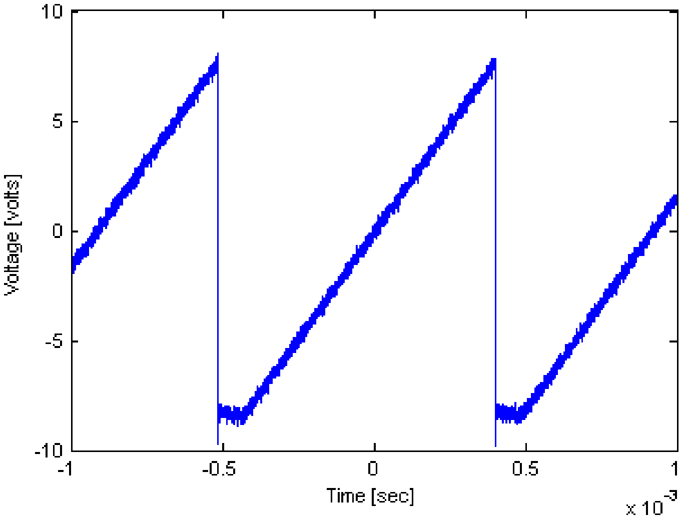


Figure 4-3: Deflection signal from the x-direction scan-controller of a Hitachi 5500 SEM for a scan with $t_{\text{dwell}} = 100\ \mu$s. The settling time between each row scan is visible when the signal resets back to -8 V.

4.2 Choosing Metrics for Evaluation

Once we have processed a dataset using its related acquisition times, we must select a metric to evaluate the performance of our method. A number of metrics for image comparison, the root mean-square error (RMSE), normalized least-square error (NLSE), peak signal-to-noise ratio (PSNR), and correlation (CORR), are given below:

$$\text{RMSE} = \sqrt{\frac{\sum_{m=1}^{M}\sum_{n=1}^{N}\left[R(m,n) - I(m,n)\right]^2}{MN}} \tag{4.1}$$

$$\text{NLSE} = \sqrt{\frac{\sum_{m=1}^{M}\sum_{n=1}^{N}\left[R(m,n) - I(m,n)\right]^2}{\sum_{m=1}^{M}\sum_{n=1}^{N}\left[R(m,n)\right]^2}} \tag{4.2}$$

$$\text{PSNR} = 10\log_{10}\left(\frac{L^2}{\frac{1}{MN}\sum_{m=1}^{M}\sum_{n=1}^{N}\left[R(m,n) - I(m,n)\right]^2}\right) \tag{4.3}$$

$$\text{CORR} = \frac{2\sum_{m=1}^{M}\sum_{n=1}^{N}R(m,n)I(m,n)}{\sum_{m=1}^{M}\sum_{n=1}^{N}R(m,n)^2 + \sum_{m=1}^{M}\sum_{n=1}^{N}I(m,n)^2} \tag{4.4}$$

where L is the maximum pixel value. The NLSE and PSNR both measure the similarity between the images by examining the difference between corresponding pixel values. High mean-square error and low signal to noise ratio mean poor agreement between images. These algebraic measures are convenient because they are easy to compute, and provide a general comparison between images, but often do not match with human perception of image quality [21]. For example, images that are translated with respect to one another but identical in every other sense will yield very different RMSE values [22]. Despite this shortcoming, we can still glean something useful by comparing with these metrics.

4.3 Sample Preparation

Since we wish to compare the performance of our correction in simulation against real datasets, we need to prepare samples to image. We use two samples for our datasets, gold nanoparticles on a silicon substrate, and carbon nanotubes. Both specimens are rich in nanometer scale features that allow for high contrast images with 500 nm field of view. The gold nanoparticle samples were prepared by Mark Mondol at the Scanning Electron Beam Laboratory at MIT, and Jim Daley at the Nanostructures Laboratory at MIT. The carbon nanotubes were provided courtesy of Dr. Bin Ming of the Nanoscale Metrology Group at NIST.

There are several degradations that occur to real samples during imaging which we do not simulate in Artimagen. Real samples may be damaged through the scanning process through several mechanisms: heating, contamination, and milling. Additionally, sample charging is a non-destructive effect which degrades image contrast. We discuss each of these effects and what precautions were taken to prevent them during sample preparation.

A current passing through a resistor will dissipate power as heat. In the case of SEMs this current is focused down to a spot-size with diameter on the order of nanometers, resulting in very high current density passing through the sample near the surface. This current may cause damage to samples with high resistivity or fragile samples, such as organic specimen [23]. A similarly destructive effect is milling. Milling destroys sample features because of energy deposition from the electron beam. The effect is more pronounced when heavier charged particles are used as the imaging

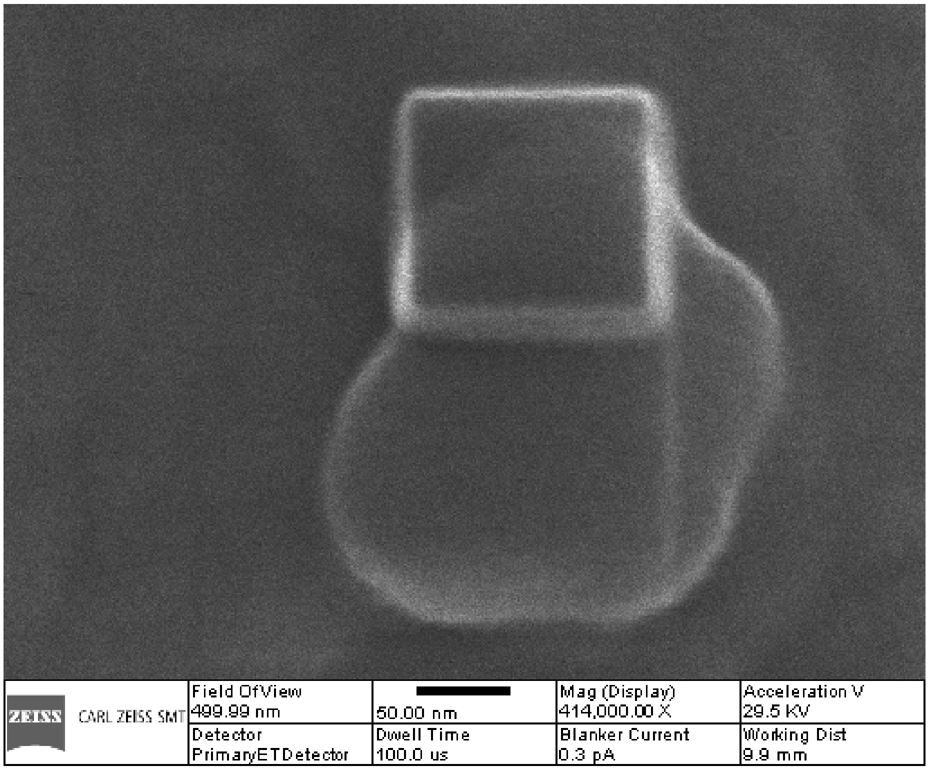
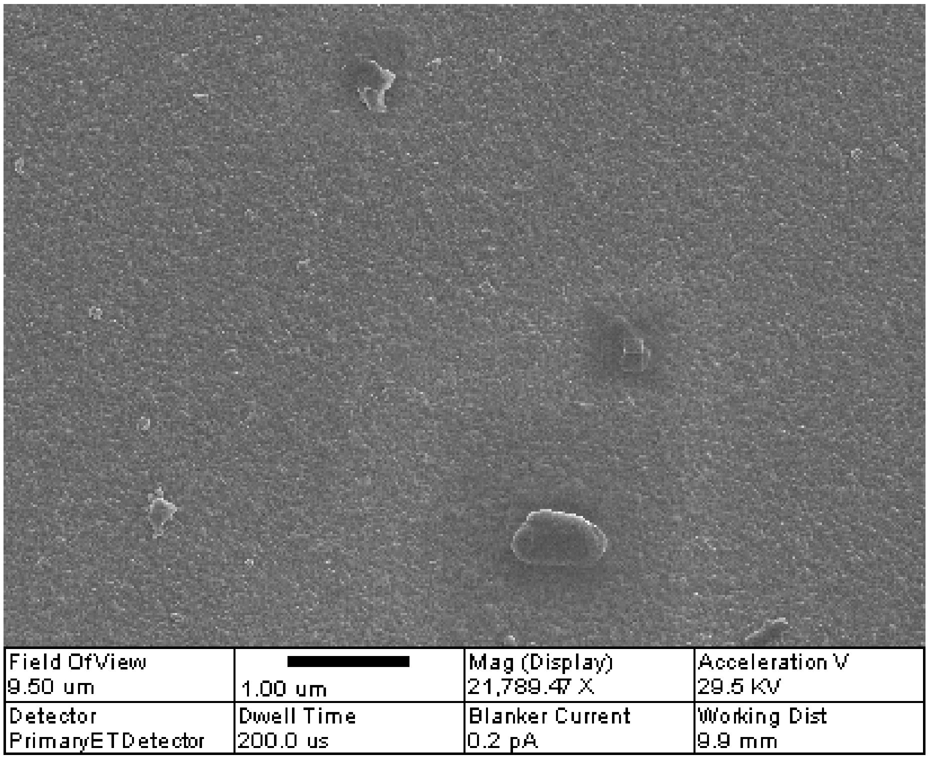


Figure 4-4: Milling is visible in the center of the image after performing focusing on a small patch of the sample. Features in the patch appear washed out and difficult to distinguish from the background.

mechanism, such as in the scanning helium-ion microscope (SHIM). The bright rectangular border in Figure 4-4 is an example of sample milling on a SHIM. Our choice of gold and carbon nanotubes are robust against both effects, and the decreased dwell times employed for our correction method minimize heat buildup in our sample.

Contamination occurs from interactions between the primary beam and gas particles in the sample chamber or on the sample surface. As the beam scans over contaminants near the surface they are vaporized and embedded into the surface causing a decrease in brightness and contrast at the contaminated region. The image in Figure 4-5 shows contamination at three regions after they were scanned at higher magnification. Good sample handling habits and regular plasma cleaning of the instrument sample chamber will minimize contamination.

Sample charging occurs when the primary beam electrons strike a non-conducting surface and begin to accumulate. The trapped charges deflect the primary beam



Field OfView 9.50 um	1.00 um	Mag (Display) 21,789.47 X	Acceleration V 29.5 KV
Detector PrimaryETDetector	Dwell Time 200.0 us	Blanker Current 0.2 pA	Working Dist 9.9 mm

Figure 4-5: Contamination of the sample surface after scanning at three locations (top-left, mid-center, bottom-center).

through coloumbic repulsion, resulting in a loss of focus for the beam. Charging is increased at higher magnifications due to the greater density of charge being deposited in the field of view. For high magnification imaging of nanometer-scale features, such as quantum dots, image contrast may degrade within seconds due to charging [2]. Firmly securing the sample to the stage mount provides a path to ground to prevent charging. Depending on the sample and magnification, it may be necessary to sputter a conductive metal film over the surface of the sample in order to increase conductivity to ground. We sputter coat our gold on silicon samples with two coats of gold-palladium in order to minimize charging and increase image contrast (see Table 2.1).

Chapter 5

Results

In this chapter we evaluate the performance of our linear drift correction method by using simulation data as well as real datasets. We first discuss the results on datasets generated using Artimagen with varying parameters of drift. Next we take datasets on three different microscope systems—the Zeiss 982 SEM at MIT's Nanostructures Laboratory, the Raith 150 lithography system at MIT's scanning-electron-beam lithography (SEBL) facility, and the ORION scanning helium-ion microscope located at the National Institute of Standards and Technology.

5.1 Comparison of Shear Correction vs. Translation Correction

The first question we wish to answer is whether or not the affine transformation used in our correction is appropriate for images distorted by drift of a constant velocity. To answer this question, we compare the results of alignment between pairs of drift distorted images simulated in Artimagen. In each pair, one image is generated with no drift distortion to serve as a reference, with a second image of the same area with drift distortion. Pairs of images are generated with drift velocities in both x and y directions between 1 px/frame to 34 px/frame. After distortion, both images are cropped to 512x512 and saved in 8-bit TIFF format. One such pair is shown

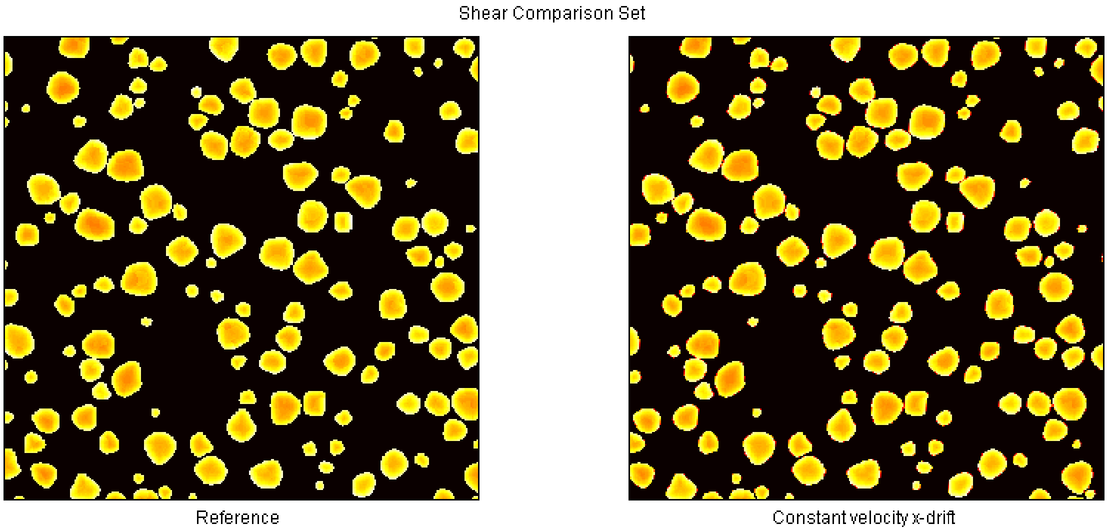


Figure 5-1: Set of images used to compare alignment methods. The effect of the linear drift can be seen by comparing both images along their right borders.

in Figure 5-1. Note the drift distortion is apparent when examining the lower right corner between the two images.

Each pair is processed by our alignment algorithm to calculate the shift vector, and the drift distorted image is corrected using one of three transformations: no correction—where no change is made to the pixel coordinates, translation correction—where a shift in x and y is applied, and affine correction—which includes both a translation and a shear component. Since the simulated images are not rotated with respect to one another we include a rotation estimate and correction.

After alignment we compute the normalized least-square error (4.2) and peak signal-to-noise ratio (4.3) between the pair, shown in Figure 5-2. Both measures confirm that our affine transformation is an appropriate compensator for constant drift velocity, and support our choice of image deformation model for such drift. One interesting trend is that the improvement provided by translation correction diminishes at increasing drift velocities. The performance of shear correction also diminishes but much more slowly. Because we knew the exact time information for the simulated pairs, this error comes from the shift vector used to estimate the drift. Errors in the magnitude of the shift vector are two-fold, causing a global mismatch in

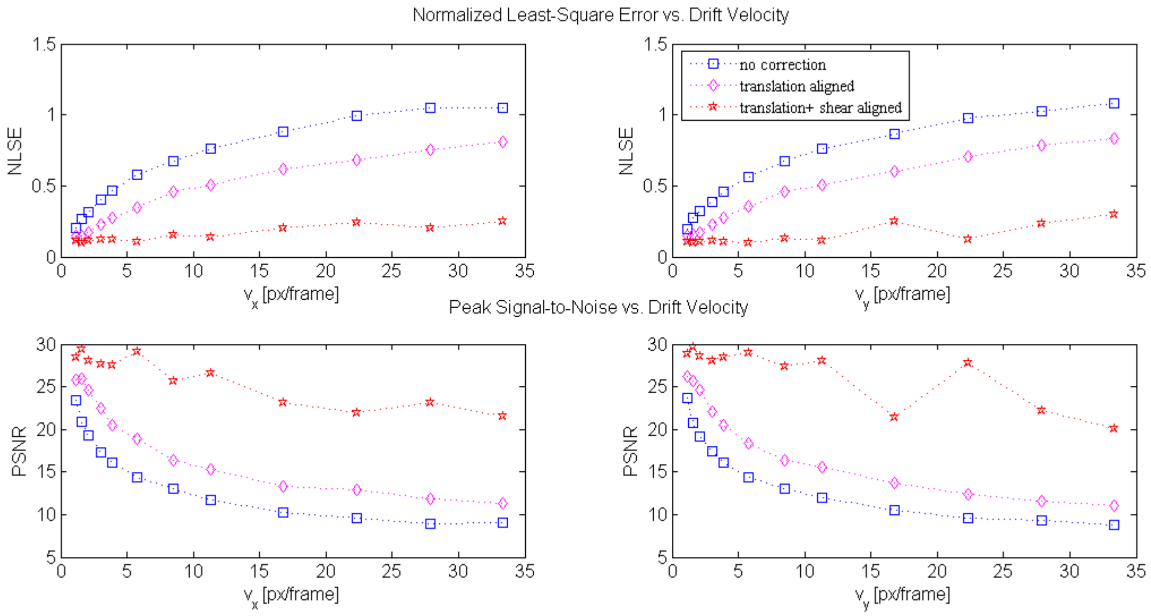


Figure 5-2: NLSE and PSNR between image pairs with increasing drift intensity for the three correction methods.

the aligned image from translation error, as well as increasingly inaccurate corrections away from the center of the image from the shear correction.

Figure 5-3 provides a visual interpretation of what is measured by our metrics. The difference images demonstrate that a significant amount of shearing is left over after translation correction, which is much reduced by including shear correction. The error in our estimate is also visible in the remaining difference near the border of the image.

5.2 Evaluating Correction under Various Drift Parameters

The next question we wish to ask is how sensitive our correction method is to different drift and image parameters. While we used image pairs above, we move to correcting a series of images for this section. The three parameters we investigate are the drift magnitude, the drift frequency, and dwell time t_{dwell}. Our drift magnitude measurements are similar to those in the previous section. By drift frequency, we mean the

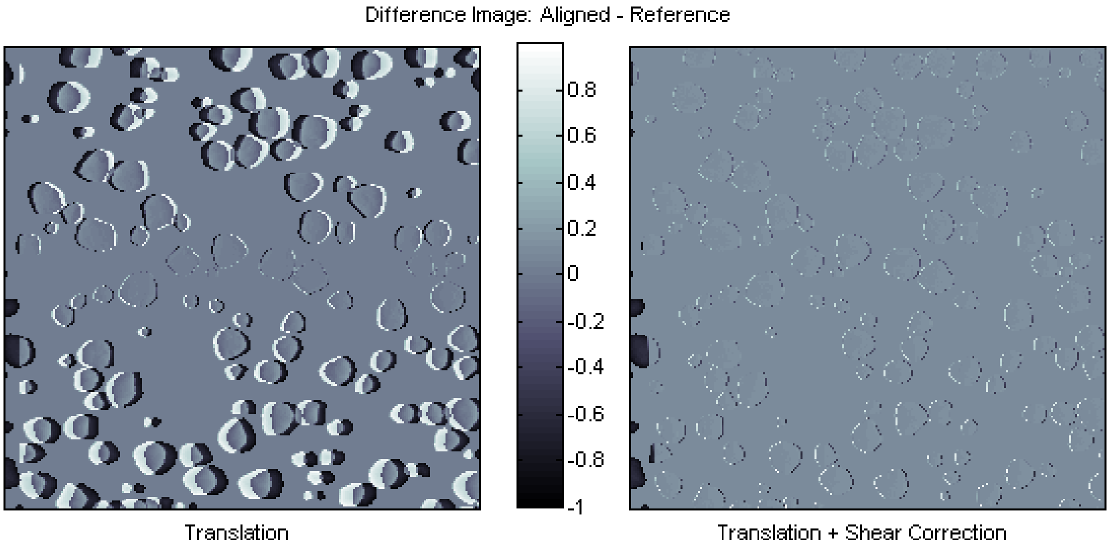


Figure 5-3: Difference Image between alignments and reference. Positive values occur where the aligned image has a feature not present in the reference, and vice versa. The negative section on the left is due to loss of image information when the aligned image is translated. The speckled edges in the shear corrected difference are due to drift vector estimation errors, caused by sub-pixel misalignment.

rate at which the drift velocity changes during the scan of the series. We use the dwell time as a measure of noisiness. As mentioned before, shot noise becomes a problem for images with short dwell times.

We generate datasets of 30 512x512 8-bit images with a fixed amount of drift and noise. Except for the parameter being varied, we hold constant all other parameters of drift, image noise, and scanning times. For scan times, t_{line} is chosen based on t_{dwell} and the number of pixels cropped when going to 512x512. Uncertainty is introduced into the image times by using a uniform distribution for total scan times, $t_{\text{frame,l}}$ t_{frame} $_{l}$ in order to introduce minor variations to the drift pattern.

We compare the composites produced by simply averaging the unaligned frames, as with the frame averaging feature available on commercial SEM software, correcting for translation only, and with shear correction and translation. We compute the NLSE and PSNR, using as a reference an undistorted version of the first frame in the series.

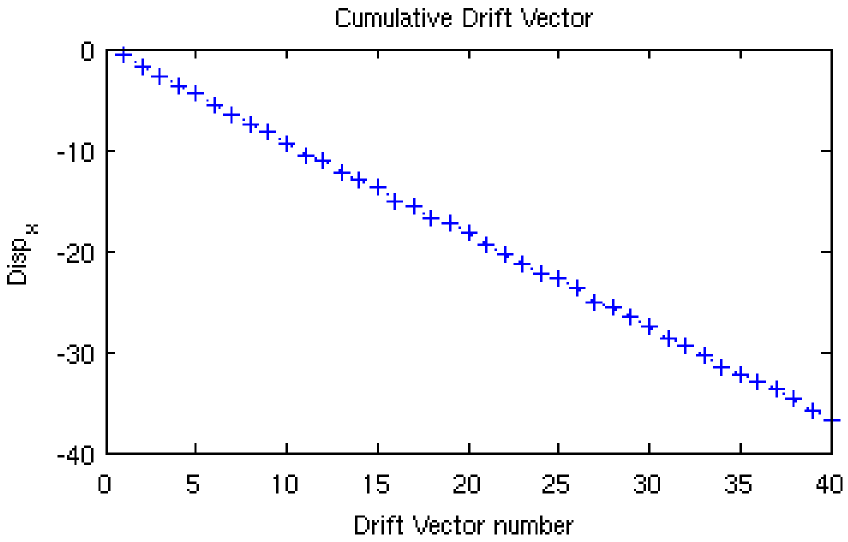


Figure 5-4: Shift vectors estimated for a simulation series of 40 frames with linear drift applied.

5.2.1 Drift Amplitude

For varying drift amplitude, images were acquired with drifts between 0.23 px/frame to 4.5 px/frame, corresponded to a motion of the field of view between 6.6 px to 130 px between the first and last image of the set. We chose this upper limit on drift to ensure sufficient overlap of corresponding areas between all images for our measurements. Figure 5-4 shows the linearity of the estimated shift vectors in a dataset.

Figure 5-5 shows the results for NLSE and PSNR as we vary the drift magnitude. For small drift velocities the shear caused by drift is minimal but becomes more significant as we increase the drift. In all cases we see that affine correction outperforms translation correction.

Figure 5-6 shows the composite images produced after correction and with no correction. In addition to clear feature boundaries, the averaging of the aligned frames removes some of the noisiness of images, further increasing contrast.

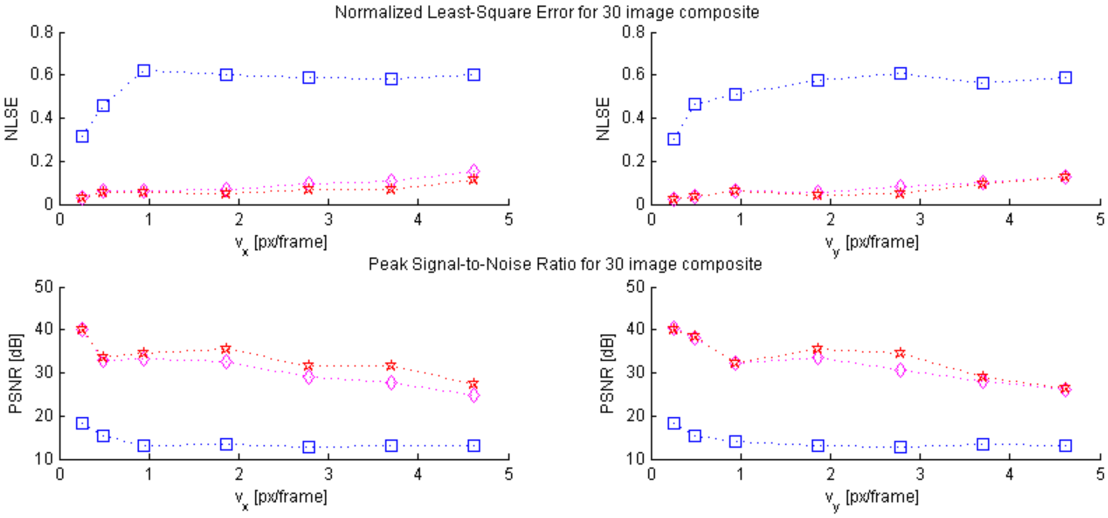
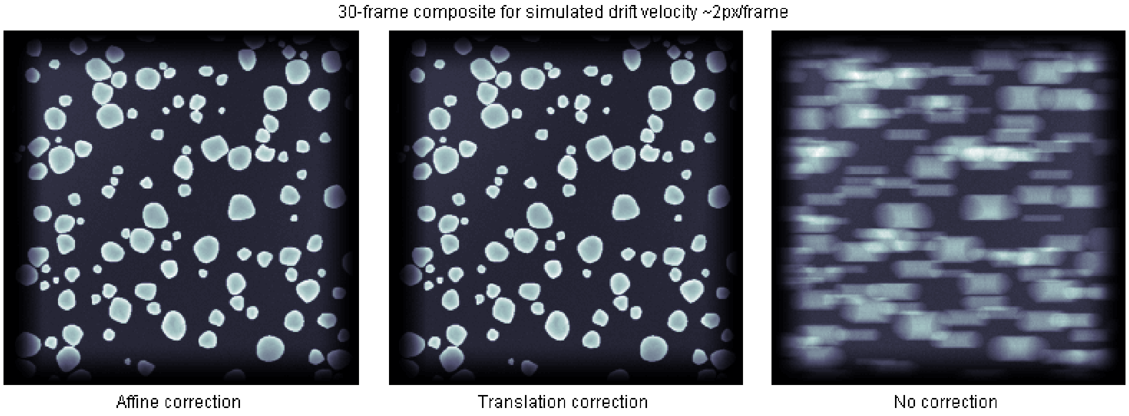


Figure 5-5: Mean-square error and SNR for composites of 30 frames created using the three correction methods under increasing linear drift.



Affine correction

Translation correction

No correction

Figure 5-6: Comparison of 30 frame composites with affine (left), translation (mid), and no correction (right) for 2 px/frame drift velocity.

5.2.2 Drift Velocity Rate of Change

Next we step away from the assumption of a linear drift, with a constant velocity, and explore how performance varies with increasingly nonlinear drifts. As mentioned above, Artimagen uses a sinusoidal form for the drift function. We previously simulated linear drift by choosing a sufficiently low frequency to approximate linearity over the time period of our image set. For $t_{\text{dwell}} = 100$ ns a frequency $f_{\text{drift}} = 0.02$ Hz was sufficiently linear over 30 images. The following datasets were simulated with f_{drift} between 0.04 Hz to 12 Hz with a maximum drift amplitude of 50 px. We maintain all other parameters as before, including noise, scan times, and image size.

As shown in Figure 5-7, efforts to correct frames above 6Hz do more harm than good. The cutoff above 6Hz marks the point at which our assumption of drift linearity begins to break down. Comparing the shift vectors for $f = 0.5$ Hz against $f = 12$ Hz in Figure 5-8, we see that at 0.5 Hz shifts may be approximated as linear with their nearest neighbors even at the cusp of the sinusoid, whereas for 12 Hz there is a much wider variation. This rapid variation in drift is most evident in images where the direction of the drift changes, as seen in Figure 5-9.

Under these conditions, our method for calculating the shift vector is insufficient. The cross-correlation is ideal for uniform translation, but for sufficiently large variations in drift image features may move relative to one another while the "center of mass" of the image remains unchanged. This weakness may be addressed by working with smaller portions of our image and estimating multiple shifts per image.

5.3 SEM Composites

5.3.1 Microscope Settings

Table 5.1 lists the settings used to acquire each of the three image series below. Images were saved in 8-bit tiff format. Frame averaging and line averaging were disabled for all of the datasets. Both the LEO and Raith instruments had in-lens secondary electron detectors, which improved image quality by decreasing spurious

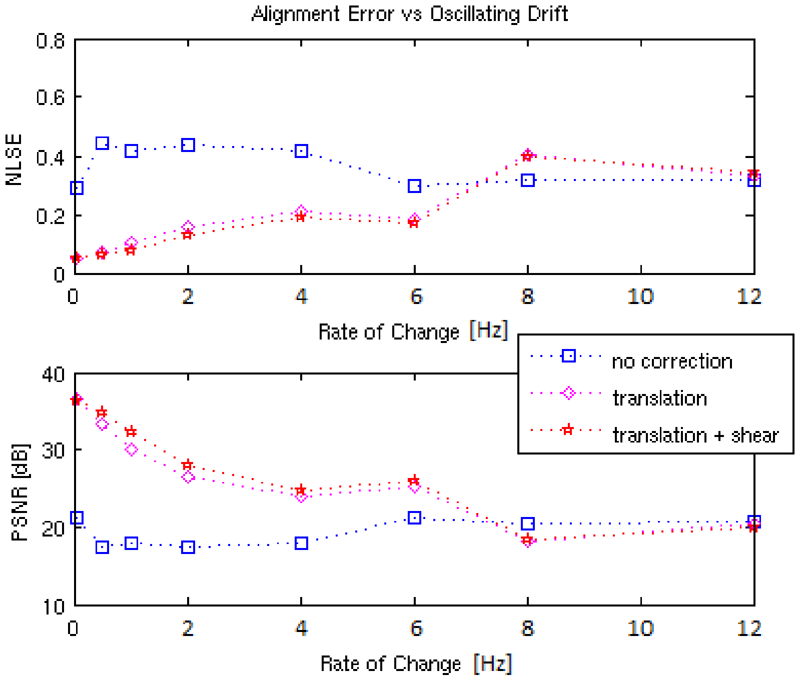


Figure 5-7: Mean-square error and SNR for 30 frame composites with increasing drift frequency.

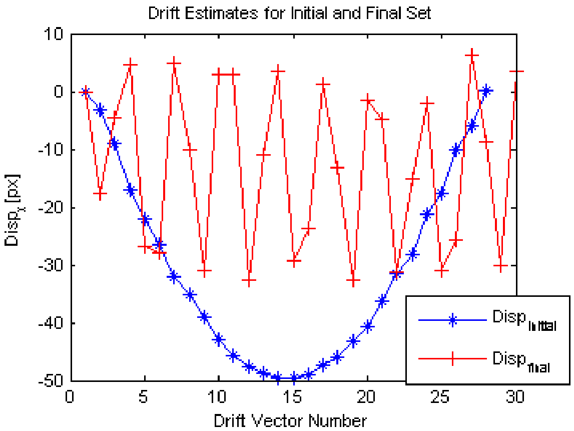


Figure 5-8: Comparison of estimated shift vectors with frequency $f = 0.5$ Hz and $f = 12$ Hz. The linearity between nearest neighbors is apparent at $f = 0.5$ Hz but breaks down as frequency of drift is increased.

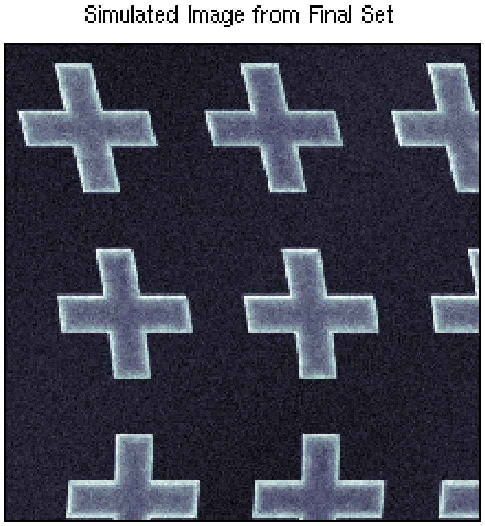


Figure 5-9: Simulated frame showing the nonlinear "bowing" caused by rapidly varying drift.

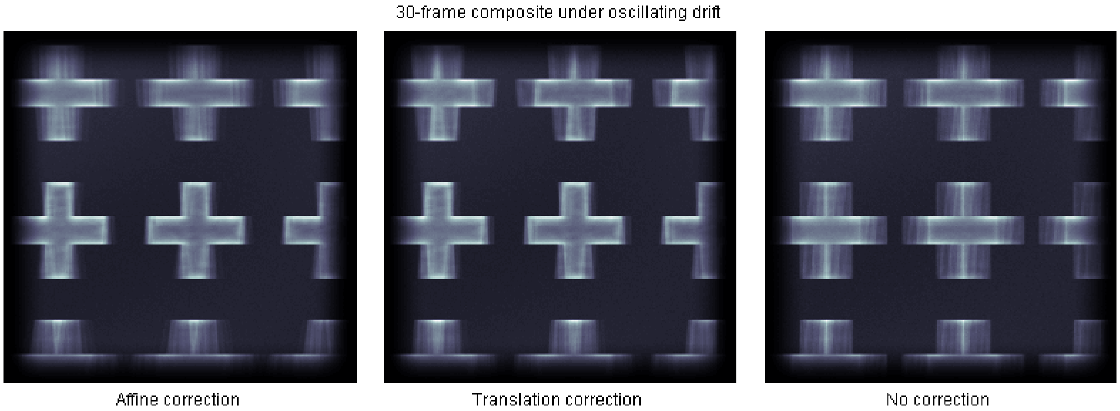


Figure 5-10: Comparison of 30 frame composites with affine (left), translation (mid), and no correction (right) with high frequency drift, f=12 Hz.

Imaging Properties	Zeiss 982 SEM	Raith 150 SEM	Orion SHIM
Accelerating Voltage [kV]	5	10	29.5
Working Distance [mm]	4	5.8	4.9
Aperture [μm]	-	30	5
Detector	In-Lens SE	In-Lens SE	Primary ET,SE
Field of View [nm]	2390	512	500
Image Size [px]	640x512	512x512	512x512
Dwell time [μs]	S1* ($\approx$5)	5	20
Object magnification	3.75 nm/px	1.0 nm/px	0.98 nm/px
Image Quantization [bits]	8	8	8

Table 5.1: Microscope Settings.

SE2 and SE3 signals (Section 2.2). The Orion system did not have this option and an in-chamber SE detector was used.

On the Raith and Orion system the image scans were automated to save and scan in sequence, with the times between images recorded in order to estimate t_{frame}. Automatically saving images on the Raith meant that for t_{dwell} =10 μs about 50% of the total time was spent scanning the image. On the Orion system with t_{dwell} =20 μs this automation allowed nearly all of the time to be scan time, with very little time spent waiting between images. The Leo SEM did not allow automatic image saving, resulting in a delay of about 40 s between images as the operator had to manually rename and save each file. Pixels on the Leo display were also scaled 4:3, requiring interpolation to create square pixels before saving the image, explaining the image

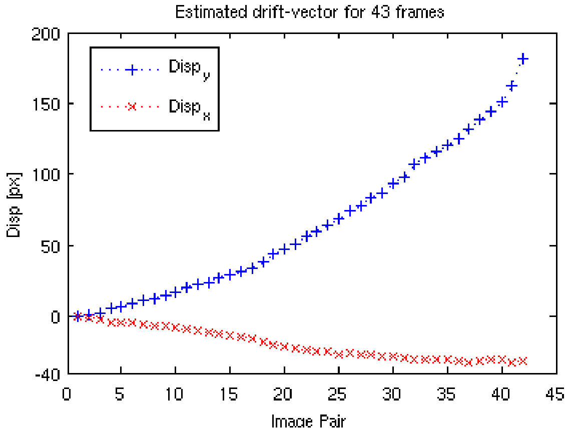


Figure 5-11: Drift vector estimates for gold on silicon dataset acquired on Zeiss 982 at NSL.

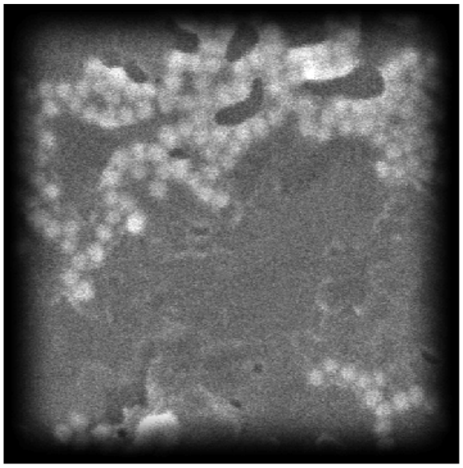


Figure 5-12: Single gold on silicon micrograph from Zeiss 982.

size.

After an initial period spent loading the sample, there was at least 15 minutes before images were taken, in order to allow for the instrument to settle after the stage was moved into position. The drift estimages and drift corrected composites for each dataset are listed below.

5.3.2 Zeiss 982

The drift estimates for our gold on silicon sample are shown in Figure 5-11. This first dataset was taken on the Zeiss 982 SEM at MIT's Nanoscale Laboratory. Because of a large initial drift, the image series was reversed and only the first 30 images were used for alignment. This allowed a larger corresponding area to be aligned between images, preserving more information in the composite. The time reversal had no effect on the shift vector estimates except to reverse their order. The smoothness of the drift affirms the choice of a linear drift model for our correction method.

Due to the large time spent between images—40 seconds compared to a 1 second scan time—the effects of shearing and scaling were minimal in this dataset. The drift correction in Figure 5-13 does include shear correction, but differences between it and translation correction alone were inconsequential. Comparing the corrected composite to a single frame in Figure 5-12 there is an improvement in image contrast,

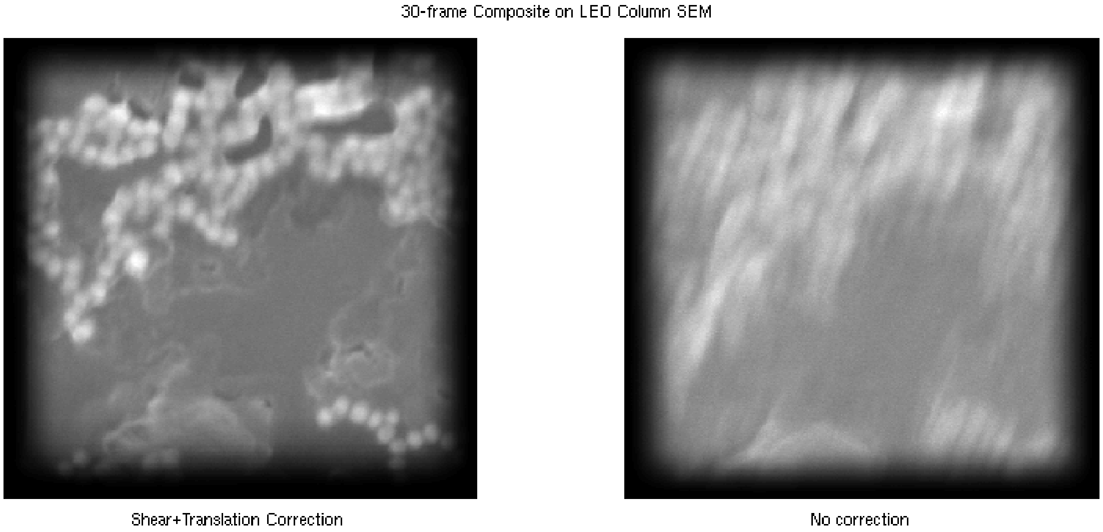

30-frame Composite on LEO Column SEM
Shear+Translation Correction
No correction

Figure 5-13: 30 frame composite of gold nanoparticles on silicon with drift correction (left) and no correction (right). Images acquired on a Zeiss 982 SEM at the Nanostructures Laboratory at MIT.

resulting in more pronounced features, especially in background regions. The blurring seen in the composite can be explained by loss of instrument focus over the 30 minutes required to acquire the dataset. Instrument focus was readjusted several times during the course of the session, which would cause misalignment of edges between frames.

5.3.3 Raith 150

The Raith 150 is an electron lithography system based on a modified LEO column. Because of the importance of maintaining the sample at a fixed position during extended lithographic exposures, the Raith is equipped with a laser interferometer which creates a feedback loop to maintain the position of the stage. Despite the compensation of the stage, Figure 5-14 shows a linear drift along the y-direction throughout the scans. The x-position of the stage appears to be maintained within 15 nm of the original position, with segments of linear drift separated by large displacements between adjacent frames. We scan another gold on silicon sample, Figure 5-15, with the Raith.

The automated image scanning feature on the Raith minimizes the wait time

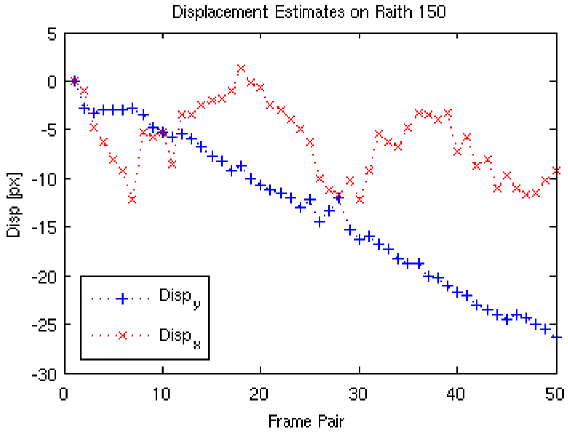


Figure 5-14: Drift Vector estimates for dataset acquired on Raith150 electron lithography system.

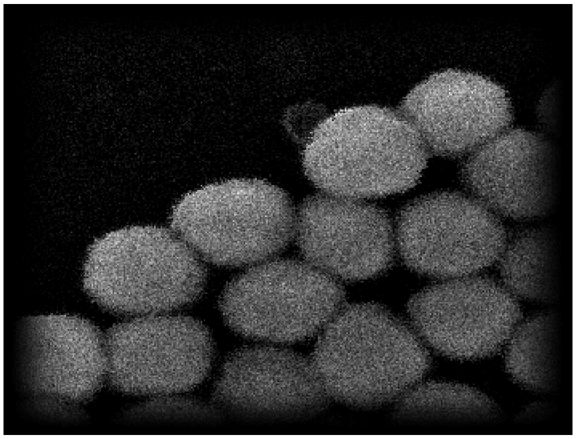


Figure 5-15: Gold on Silicon micrograph from Raith 150.

between scans and also saves the times between images, which gives us much more accurate timing information for drift correction. While we set t_{dwell} and measure t_{frame}, we vary t_{line} in order to optimize the PSNR between our 50 frame composite in Figure 5-16 and the reference frame in Figure 5-15. This maximum occurs for $t_{\text{line}} \sim t_{\text{row}}$.

Figure 5-17 compares the affine correction against only translation correction. What is shown is the difference image obtained by subtracting the affine corrected composite from the translation corrected composite. The two images differ in pixel intensity by as much as 38% along the edges of the gold nanoparticles. Since we don't have an undistorted reference to compare against, we filter the first frame and measure the PSNR difference for the two composites. We find that the affine transformation provides an improvement of 0.4 dB over translation correction.

5.3.4 Scanning Helium-Ion Microscopy on the Orion

While our correction method has been developed with correcting SEM images in mind, it is generally applicable to any scanning-beam microscope system. The first commercial scanning helium-ion microscope was the Orion system produced by Zeiss. The imaging mechanism of the SHIM are analagous to that of SEMs, except more massive helium ions are used to bombard the sample surface. The trajectory of the

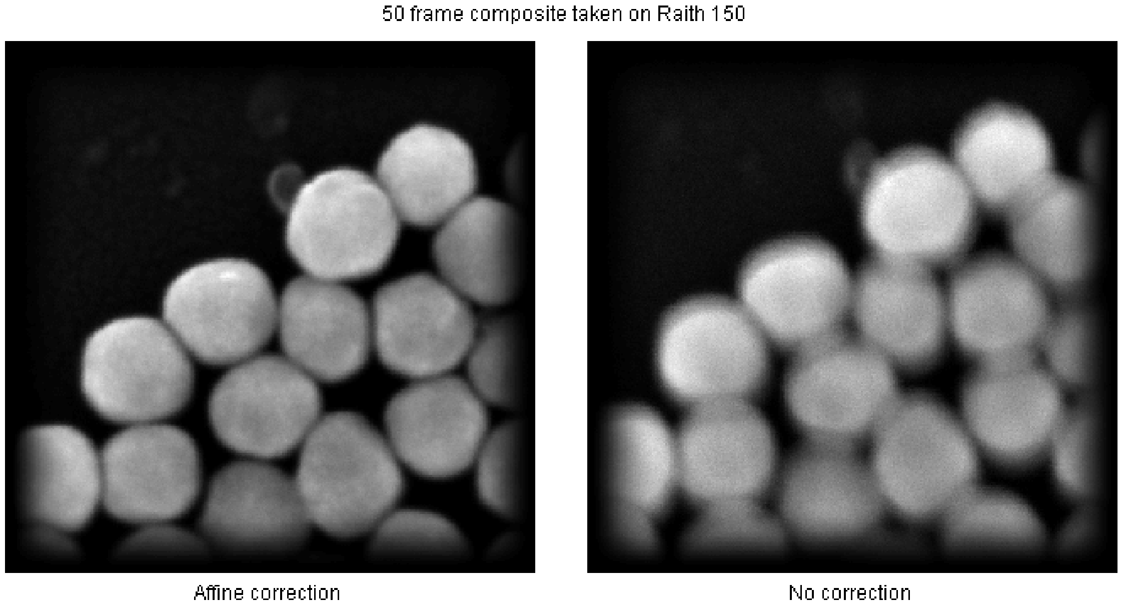


Figure 5-16: 50 frame composites of gold on silicon using Raith150 system at SEBL at MIT.

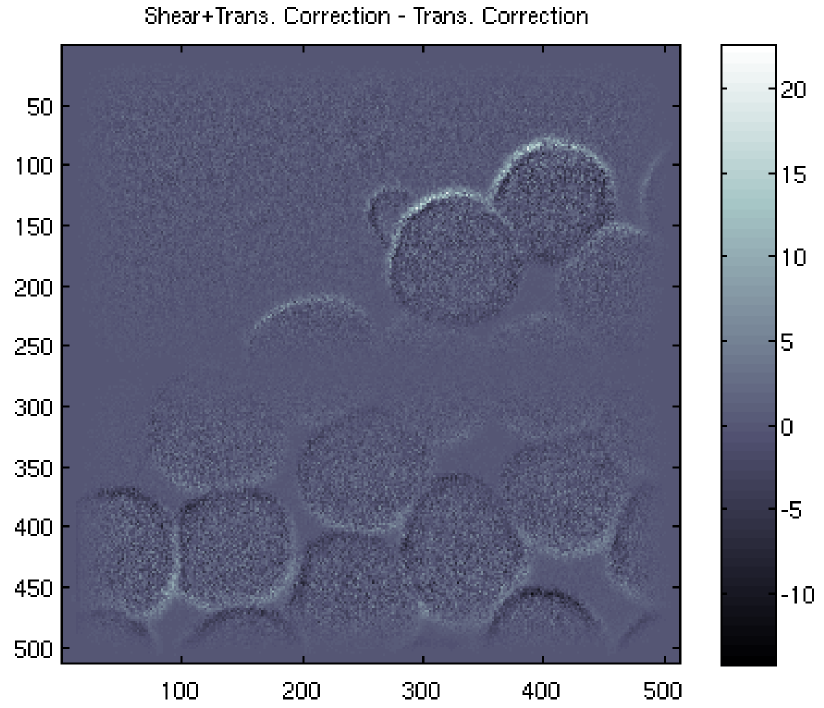


Figure 5-17: Difference Image comparing Shear correction against Translation correction on the Raith dataset. Color indicated difference at corresponding pixels.

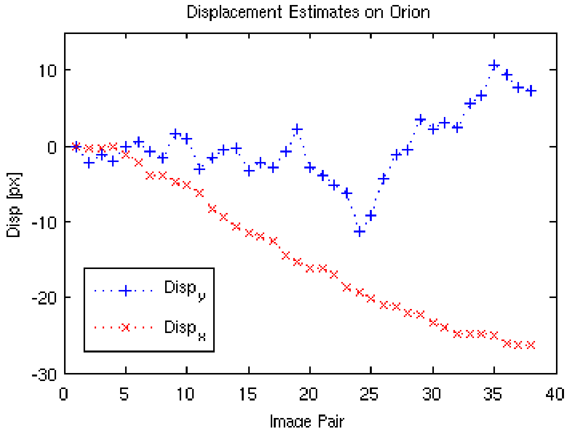


Figure 5-18: Drift Vector estimates for dataset acquired on Orion Scanning Helium Ion Microscope.

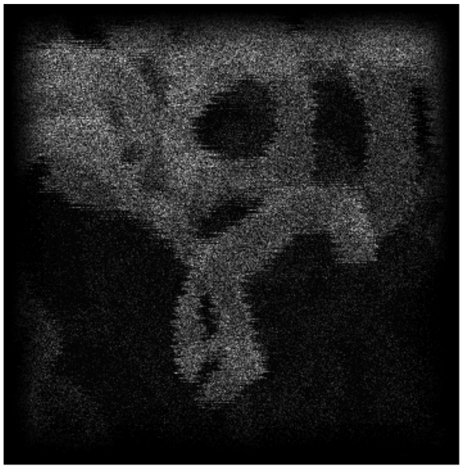


Figure 5-19: Gold on Silicon micrograph on Orion SHIM, scan speed $t_{\text{dwell}} = 20\mu$s.

massive helium ion is more difficult to deflect than that of the electron, resulting in a more confined volume from which SE are emitted [24]. Due to the greater potential for sample damage, a much lower beam current is used, which requires longer dwell times in order to achieve the same contrast as an SEM.

Carbon nanotubes were imaged on the Orion system due to their robustness against sample damage. Figure 5-18 shows a fairly smooth drift in the x-direction for the dataset. The jagged line edges in Figure 5-19 were caused by unfortunately scanning as the cryogenic pump used to cool the gun was activated.

The result of drift correction on the Orion system are shown in Figure 5-20. Our method is applicable to both types of scanning-beam microscope.

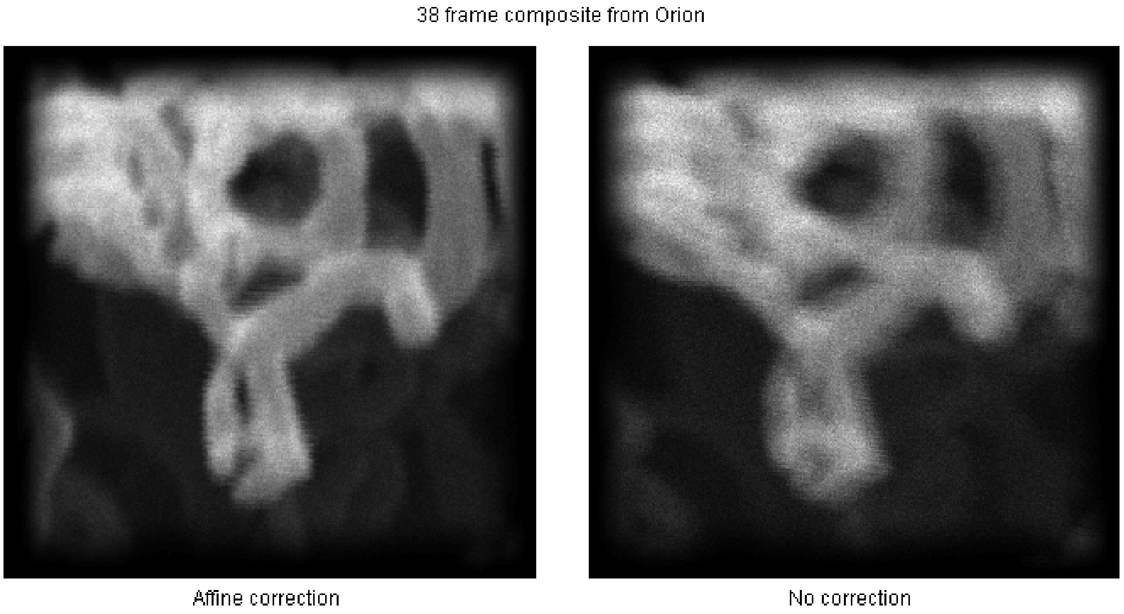


Figure 5-20: 38 frame composite of carbon nanotubes with drift correction (left) and without correction (right). Images taken on the Orion system at NIST.

Chapter 6

Conclusion

Now that we have evaluated our drift-corrected composites against commercially available averaging methods on the SEM and examined the effectiveness of affine correction compared to translation only, à la Cizmar [5, 11], we can draw some general conclusions. Most importantly, how do we fare in the face of the question we proposed to solve in Section 1.2?

6.1 Discussion

It's clear that our drift correction accomplishes the goal we set out: our composite images have gained the benefit of long scanning times, with greatly diminished drift-distortion. Our uncorrected composites expose the sad state of commercial SEM options in dealing with drift. Additionally, the performance of our affine correction against existing translation correction is promising for model-based distortion correction. We have demonstrated that linear drift was an appropriate assumption for all three of our datasets, and in general the diversity of instruments which were employed—a standard SEM, a high-end lithography system, and a modern SHIM—goes to expose the ubiquitous nature of such drift in scanning-beam microscopes.

While the improvement of our affine correction compared to translation only is marginal on the real datasets, this can be explained by the circumstances of the measurement. Both PSNR and mean-square error rely on pixel-to-pixel correspondence to determine how good a match there is between test and reference images. Since these are global measures, all of these comparisons have equal weight, which is not appropriate when most of the useful information we wish to compare is contained in a small section of each image around feature edges. A more sophisticated comparison could be made with a feature selective metric, or perhaps some form of edge comparison through edge-detection. However, the increase in image quality across the full range of our measurements suggest that affine correction does serve an advantage.

The advantage of any drift correction technique is the use of additional information encoded in the series of images to compensate individual images. In our case, this additional information is used explicitly through the model-based transformations applied to each image. By including additional timing information for our images, our correction method is robust in the sense that it is applicable over a wide range of imaging conditions. As long as it is possible to record the timing information t_{dwell}, t_{line}, and t_{frame}, the algorithm will perform without regard to the scan parameters.

6.2 Further Work

There are several areas which could yield improved performance for our correction method. The first proposed investigation is replacing the global shift vector with multiple shift vectors. The advantage of multiple shift vectors is localization in time. By breaking the image down into multiple horizontal slices we take advantage of the raster pattern to cluster together segments of the image which are related in time. Such slices will allow for a finer time granularity in the drift estimates and will provide additional information for changes in drift within each frame. Secondly, an alternative peak estimator for fitting the cross-correlation using wavelet transforms was proposed by Chang [25]. The Gaussian form of the cross-correlation peak (see Figure 3-4) matches well with the wavelet form, and it would be worthwhile to investigate the effectiveness of such a switch. Finally, while not an element of the correction method, employing a more appropriate metric for evaluation, such as phase congruence [21, 22] might yield a more accurate measure of algorithm performance.

The results of the simulated image pairs in Section 5.1 suggest that by further increasing t_{dwell}, there would be more need for affine correction. Instead of a series of images, an alternative image sequence would be a single, quickly scanned frame followed by a long scanned frame. This would correspond more closely with our simulation, and would rely solely on correcting features within the frame rather than creating a composite.

6.3 Concluding Remarks

The goal of this project has been to establish accurate, repeatable imaging and data acquisition on the SEM by correcting drift. An improvement in the area of drift compensation may mark a watershed moment in SEM metrology. If drift can be taken off the table, a clean reference sample can provide information about the beam profile of the SEM scanning it. This information can be used for further quality improving corrections within the instrument. This type of correction can also work in reverse, with a well-understood beam profile yielding more accurate information about the measured sample.

If we can eliminate the influence of drift, the hope is that we can better understand how to improve the information gathering capabilities of the SEM as we continue to explore more sophisticated sample structures on the nanometer scale. This author hopes that this work is a small, fruitful step in the right direction.

Appendix A

MATLAB Code

Included below are the overall script and MATLAB functions written for the algorithm.

alignment.m

Alignment.m is a script which calls each operation of the algorithm and stores the final composite.

```
% SEM image alignment algorithm written in MATLAB
% (c)Michael Snella
% ver 0.3 8/12/10

%Dimension of output image in pixels, dxd.  Must be square.
d=512;
ovsample=2; % Increase image x/y by 2^ovsample
kernel_size=5; %NxN pixel kernel
%=======IMAGE LIST TO PROCESS=======================================
clear image_list;

name_string1=’name’;
name_string2=’.tif’;
for k=1:50
    image_list{k}=[name_string1 num2str(k) name_string2];
end

%=======INITIATE FIRST IMAGE (REFERENCE)=============================
disp(’imalign>>WINDOWING INITIAL IMAGE...’)
```

```
%cropped, windowed reference
im_ref=imwindow(imtrim(imread(image_list{1}),d));

im_aligned=im_ref; %Initial composite image
im_aligned_nsc=im_ref; %Initial composite image no shear corr
im_aligned_no=im_ref; %simple average

disp(['imalign>>PERFORMING ' num2str(2^(ovsample)) 'x EXPANSION...'])

im_ref_os=interp2(im_ref,ovsample,'*linear'); %oversampled image

%=======INITIATE SHIFT VECTORS=========================================

shift_adjacent=[0 0]; % shifts between a pair of images, nx2 vector.
shift_cumulative=[0 0];

%=======TIME DATA FOR SHEAR CORRECTION=================================

    t_pixel = 100;  %Dwell time for each pixel in ns
    t_linereset = 0.3*512*t_pixel; %Time to reset between image rows in ns
    t_frame = 100*10^6; %Time from start to start between frames in ns
    t_info=[t_pixel, t_linereset, t_frame];

% =======FOR LOWPASS FILTER ============

lpf_s=filter2d(kernel_size);

ovsample_size=(d-1)*2^ovsample+1;%size after interpolation
lpf_f=fft2(lpf_s,ovsample_size,ovsample_size);%fft of filter kernel

%NB:zero padding of kernel is necessary for your image.

lpf_ff=lpf_f.*conj(lpf_f);

%=======LOOP FOR EACH IMAGE=======================================
for num=2:length(image_list)

    %=========CROP=AND=WINDOW========================================
    im_i=imtrim(imread(image_list{num}),d);   %Read image & Crop
    disp('imalign>>WINDOWING IMAGE...')
    im_i_windowed = imwindow(im_i); % trimmed, window frame


    %======SPACIAL=CROSS-CORR========================================

    disp(['imalign>>PERFORMING ' num2str(2^(ovsample)) 'x EXPANSION...'])
    im_i_os=interp2(im_i_windowed,ovsample,’*linear’);

    % --- corr between two adjacent images -----

    disp(’imalign>>PERFORMING CROSS-CORRELATION...’)

    f_corr=fcorr(im_i_os,im_ref_os);
    f_corr_filtered=f_corr.*lpf_ff;
    s_corr=fftshift(ifft2(f_corr_filtered)); %Center is origin

    %======FIND=SHIFT-VECTORS========================================
    disp(’imalign>>EXTRACTING SHIFT VECTOR...’)

    %Shift vectors for adjacent images
    shift_adjacent(num,:)=shiftv(s_corr)./(2^ovsample);

    %cumulative shift vector
    shift_cumulative(num,:)=shift_cumulative(num-1,:)+...
     shift_adjacent(num,:);

    disp([’shift’ num2str(num) ’=’]), disp(shift_cumulative(num,:))

    % ---- Update adjacent images ----

    disp(’imalign>>UPDATING ADJACENT IMAGE MARKER’)
    im_ref_os=im_i_os; %update the reference image for next iteration.

    % ====== ALIGN IMAGE TO REFERENCE ==============

    im_al=imwindow(affshift(im_i,-2*shift_adjacent(num,:),...
     -shift_cumulative(num,:),[t_pixel, t_linereset,tframe]));

    im_al_nsc=imwindow(affshift(im_i,[0,0],-shift_cumulative(num,:),...
     [t_pixel, t_linereset, tframe]));

    im_aligned=im_al+im_aligned;
    im_aligned_nsc=im_al_nsc+im_aligned_nsc;
    im_aligned_no=im_i_windowed+im_aligned_no;
end

im_aligned=im_aligned/num;

%no shear correction, only translation
im_aligned_nsc=im_aligned_nsc/num;

%no correction, simple average
im_aligned_no=im_aligned_no/num;
```

```
disp('imalign>>DONE')
```

affshift.m

Affine transformation function. Calculates and applies the correction discussed in Section 3.5.

```
function [im_out,A] = affshift(im_in,shift,disp,times)

% affshift - 2D affine transformation to reverse shearing/scaling from
% drift.
% v0.2 5/13/10
%
% [im_shifted] = affshift(image_in,shift,times)
%
% Inputs: (4) image_in - image to process
%             shift - 1x2 vector of shift [x y] btw adjacent images.
%             disp - 1x2 vector with translation wrt reference [x y].
%             times - 1x3 vector with time parameters from image scan
%                            [t_pixel,t_linereset,t_frame]
% Outputs: (1) im_shifted - affine transformed output image
%
% Description:
% Input shift vector must be 1x2, not checked.  Shift vector values are the
% x and y components of the spatial shift.  These shifts can be positive or
% negative.  Time vector contains dwell-time per-pixel, dead-time between
% lines, and start-to-start time between frames (time "pitch" for the
% series from which the frame is taken).
%
% NB: In Matlab, A(x,y) is indexed w.r.t X - ROW (vertical index)
%                                        Y - COLUMN (horizontal index)
%

ux = shift(1);
uy = shift(2);

Wx = size(im_in,1);
Wy = size(im_in,2);

%break out the three time components
t_pixel = times(1);
t_linereset = times(2);
t_frame = times(3);
```

```
%additional spacing created by drift during scan
x_space = (t_pixel*Wy+t_linereset)*ux/t_frame;
y_space = t_pixel*uy/t_frame;

%image warping along x/y direction
%NB: this is different from the shear factor for each index, here x and y
%are just the directions in which the shearing factor acts.
y_shear = (t_pixel*Wy+t_linereset)*uy/t_frame;
x_shear = t_pixel*ux/t_frame;


%correction factors for shearing and scaling
% y'=y+shy(x)
% x'=x+shx(y)
shx = x_shear;
shy = y_shear;

kx = (1+x_space);
ky = (1+y_space);

%Displacement of center of frame due to shearing w.r.t. the upper-left
%corner.  I.E.-MATLAB treats upper-left pixel as origin for each frame.
d_x = 0.5*(x_space*Wx+x_shear*Wy);
d_y = 0.5*(y_space*Wy+y_shear*Wx);

%correction for displacement of center due to shear removal.

tx=disp(1)-d_x;
ty=disp(2)-d_y;



A = [ky shx 0; shy kx 0; ty tx 1];
tform = maketform('affine',A);
[im_out] = imtransform(im_in,tform,'bilinear',...
    'FillValues',0,...
    'XData',[1 Wx],...
    'YData',[1 Wy]);
```

fcorr.m

Due to calculation speed, a frequency domain cross-correlation is used instead of a spatial cross-correlation.

```
function [f_corr]=fcorr(image,reference)
% 2D cross-correlation of two matrices
% v0.41 7/29/10
%
% fcorr takes two matrices and performs a cross-correlation in the
% frequency domain, which can then be filtered or transformed back to
% spatial domain.
%
% [f_corr]=fcorr(image,reference,oversampling)
%
% Inputs: (2) image - matrix compared w.r.t reference matrix,
%             reference - reference matrix
% Outputs: (1) f_corr - FFT of spatial cross-correlation
%
% Description:
% The cross-correlation can be represented as
% IFFT( complexconj( FFT(reference)) * FFT(image) )
% Take note that the complex conjugation of the reference FFT means that
% the peak position in the cross-correlation gives the relative offset of
% image with respect to reference, in pixels.

if ~(size(image,1)&&size(image,2)&&size(reference,1)&&size(reference,2))
    error('fcorr>>Images not same size or square.  Please run imtrim.')
end

d = size(image,1); %If images are same size, get the dimension.

f_image=fft2(image,2*d-1); %Perform the 2D FFT,
f_ref=fft2(reference,2*d-1);
%Make sure to zero pad each image so there is no aliasing in the FFT.

fftcorr = f_image.*conj(f_ref);
% Generate the crosscorrelation by FT trickery. Pay attention to the conj
% applied to ref, meaning the x-corr is of the image w.r.t the reference.

f_corr=fftcorr;  %correlation in frequency domain.
```

imtrim.m

Crop and change the type of image data to double. Although not fully implemented, imtrim would allow cropping rectangular blocks. This would allow for multiple shift vectors to be estimated per image pair, as proposed in the final chapter.

```
function image_trimmed=imtrim(image,varargin)
```

```
% Trims an image down to d-by-d pixels, and makes it type double.  Image
% implicitly needs to be a 2-D matrix.
% v0.2 7/30/10
%
% [trimmed_image]=imtrim(image,d) or imtrim(image,top,bot,left,right)
%
% Inputs: (2/5) image - input matrix to trim;
%               d - width/height in pixels
%               top,bot,left,right - pixel row/column borders for cropping
% Outputs: (1) trimmed_image - square, type double image
%
% Description:
% Need to make sure you supply a value for either d or the corners.  Using
% d selects upper-left corner as 1,1 mark.
% lt,rt,lb,rb correspond to px positions used for cropping.
% It's your problem if you feed in a non-2D input.

if (size(varargin,2)==1)
    disp(sprintf('imtrim>>CROPPING IMAGE TO %dx%d',varargin{1},varargin{1}))
    image_trimmed=image(1:varargin{1},1:varargin{1});

elseif (size(varargin,2)==4)
    disp(sprintf('imtrim>>CROPPING IMAGE TO %dx%d',...
     (varargin{2}-varargin{1}+1),(varargin{4}-varargin{3}+1)))
    image_trimmed=image(varargin{1}:varargin{2},varargin{3}:varargin{4});

else
    error('imtrim>>incorrect number of inputs')
end

if ~(isa(image_trimmed, 'double'))
    image_trimmed = double(image_trimmed);
end

disp('imtrim>>IMAGE CROPPED')
end
```

imwindow.m

Apply a 2D tukey window in the image domain.

```
function [image_iw]=imwindow(image_i,R)
% Windowing sub-function for image alignment setup
% v0.41 7/30/10
```

```
%
% image_w=imwindow(image,[R])
%
% Inputs: (1-2) image - image to window, [R - optional rolloff parameter.]
% Outputs: (1) image_w - windowed image.
%
% Description:
% 2-D tukey window to eliminate boundary effects from the image.  Accepts
% an optional parameter R for window taper: 0 - rectangle to 1 - hann.
% imwindow assumes a class double input image.

if (nargin==1)
    tukey_R=0.25; %Default Rolloff
elseif (nargin==2)
    tukey_R=R;
else
    error('imwindow>>WRONG NUMBER OF INPUTS')
end

if ~strcmp(class(image_i),'double')
    error('imwindow>>IMAGE MUST BE CLASS DOUBLE')
end

image_x = size(image_i,1); %Return height.
image_y = size(image_i,2); %width.

window_horiz = tukeywin(image_y, tukey_R)'; %2-D window setup
window_vert = tukeywin(image_x, tukey_R);

window_2D = window_vert*window_horiz; % 2-D window

image_iw = image_i.*window_2D;  % Windowed image

end
```

shiftv.m

Shift vector estimation function. Shiftv.m detects the peak of the cross-correlation between two images and returns it as a vector $[\delta_m \delta_n]^T$ in units of pixels.

```
function v_shift = shiftv(corr)

% Produces the x and y shifts from the peak of a crosscorrelation.  Based
% off of peak.m.
% v0.2 4/26/10
%
% [shiftx,shifty]=shiftv(xcorr,scale)
%
% Inputs: (2) xcorr: spacial crosscorrelation
% Outputs: (2) x: x-index of shift, y: y-index of shift
%
% Description:
% Output vector adjusts for indexing on 1, so [1,2] means a
% displacement of +1 in x and +2 in y, or that peak is at position [2,3].
%
% (Sub-pixel accuracy is achieved by the scaling factor, which is passed
% from the cross-correlation generation. Not implemented)

[max_val,max_index]=max(abs(corr(:)));
[xpeak,ypeak]=ind2sub(size(corr),max_index); %Convert the index to an x,y.

d=size(corr,1);

if mod(d,2)
    center = [(d+1)/2,(d+1)/2]; %center for odd image
else
    center = [d/2+1,d/2+1]; %center for even.
end

v_shift = [xpeak-center(1),ypeak-center(2)];


%Check if the peak x,y are in the rhp of image.  If they are, consider it a
%negative shift.
%{
if (xpeak > floor(size(corr,1)/2));
    x = xpeak - size(corr,1) - 1;
else
    x = xpeak - 1;
end
if (ypeak > floor(size(corr,2)/2));
    y = ypeak - size(corr,2) - 1;
else
    y = ypeak - 1;
end

v_shift = [x , y];
%}
```

filter2d.m

Creates a 2D filter kernel in the image domain using the window method [16]. Hamming and Blackman functions are included, but any windowing function can be used.

```
function [kernel]=filter2d(size)
% 2D image filter kernel, defined in space
% v0.1 7/28/2010
%
% filter2d generates a 2D FIR filter from a 1D window.
% The size of the filter kernel (n-by-n) is square.  Blackman filter.
%
% [kernel]=filter2d(size)
%
% Input: (1) size - the size of the filter in pixels - (size x size)
% Output: (1) kernel - spacial kernel for selected filter
%
% Description:
% The 2D filter is generated using the windowing method.  A 1D window is
% generated from a continuous equation for the filter.  The curve is
% rotated along its axis of symmetry to extrude a 2D surface -
% w(t)|t=sqrt(x.^2+y.^2).
% The equation is adjustable, the blackman window is currently used.

[f1,f2]=freqspace(size,'meshgrid');
%create a grid of discrete frequencies in x and y
%(from -1 to 1, not -0.5 to 0.5)


%hamming = @(f1,f2) (0.54-0.46*cos(2*pi*sqrt(f1.^2+f2.^2)/2-pi));
%blackman= @(f1,f2) (0.42-0.5*cos(2*pi*sqrt(f1.^2+f2.^2)/2-pi)+...
0.08*cos(4*pi*sqrt(f1.^2+f2.^2)/2-2*pi))

ffilter = blackman(f1,f2);
%The equation needs to be adjusted to center around 0 and stretched to
%expand from -1 to 1.

mask=((f1.^2+f2.^2)<=1);%logical mask to cut off the filter at radius=1.

kernel=ffilter.*mask;

end
```

imrotdetect.m

Rotation detection function based off [17].

```
function angle_sum=imrotdetect(image)
% Sums the frequency domain information of an image along rays from -180 to
% 180 degrees.
% v0.1 8/01/10
%
% [angle_sum]=imrotdetect(image)
%
% Inputs: (1) image - input image;
% Outputs: (1) angle_sum - 1d function.
%
% Description:
% This function is used to detect the rotation angle between two images.
% Similar to proposed method by Vandewalle.

M=size(image,1);
N=size(image,2);
f_image=fftshift(fft2(image));

[fx,fy]=meshgrid(linspace(-1,1,M),linspace(-1,1,N));

theta_map=atan(fx./fy);
%vertical over horizontal, x and y reversed in matrix column row notation.
theta_map(1:256,1:256)=theta_map(1:256,1:256)-pi;
theta_map(257:512,1:256)=theta_map(257:512,1:256)+pi;
r_min_mask=(0.01<=fx.^2+fy.^2);
r_max_mask=(fx.^2+fy.^2<=1);
r_mask=r_min_mask.*r_max_mask;

anglestep=0.1;
delta_angle=0.05;

for t=0:(360/anglestep)
    theta_mask=and(((t*anglestep-180-delta_angle)*pi/180<=theta_map),...
        (theta_map<=((t*anglestep-180+delta_angle)*pi/180)));
    mask=and(theta_mask,r_mask);
    angle_sum(t+1)=sum(abs(f_image(mask)));
end

end
```

Appendix B

Artimagen code

The Artimagen library allows datasets to be generated from a C/C++ program. The source code used to generate the datasets in this thesis is included in this appendix.

```
/*
Template for generating series of drift-distorted .tiff images
based off of P Cizmar’s examples included with Artimagen.
*/


#define SIZEX 612
#define SIZEY 612

#include <iostream>
#include <cmath>
#include <ctime>
#include <cstdlib>
#include <artimagen_i.h>
#include <vector>

using namespace std;
using namespace artimagen;


//====SAMPLE PARAMETERS=====
void generate_gold_on_carbon_image(CSample **sam){
   t_gc_definition gcdef;

   gcdef.sizex=SIZEX;
   gcdef.sizey=SIZEY;
   gcdef.ee_coefficient=0.2;
   gcdef.ee_top_above_base=0.3;
   gcdef.base_level=0.5;
   gcdef.base_level_variation=0.1;
   gcdef.grain_min_size=5;
   gcdef.grain_max_size=20;
   gcdef.number_of_grains=SIZEX*SIZEY/2000;
   gcdef.fs_density = 7e-3;
   gcdef.fs_min_r = 6;
   gcdef.fs_max_r = 10;
   gcdef.fs_min_coe = 0.95;
   gcdef.fs_max_coe = 1.05;
   *sam = new CGoldOnCarbonSample(&gcdef);
}

void generate_corner_structure_image(CSample **sam){
   t_cor_definition cordef;

   cordef.sizex=SIZEX;
   cordef.sizey=SIZEY;
   cordef.ee_coefficient=0.01;
   cordef.ee_top_above_base=0.1;
   cordef.base_level=0.5;
   cordef.base_level_variation=0.2;
   cordef.lsize=50;
   cordef.tsize=150;
   cordef.distance=200;
   cordef.fs_density = 7e-4;
   cordef.fs_min_r = 6;
   cordef.fs_max_r = 10;
   cordef.fs_min_coe = 0.95;
   cordef.fs_max_coe = 1.05;
   *sam = new CPeriodicCornerSample(&cordef);
}

void generate_cross_structure_image(CSample **sam){
   t_crs_definition crsdef;

   crsdef.sizex=SIZEX;
   crsdef.sizey=SIZEY;
   crsdef.ee_coefficient=0.2;
   crsdef.ee_top_above_base=0.5;
   crsdef.base_level=0.3;
   crsdef.base_level_variation=0.2;
   crsdef.lsize=70;
   crsdef.tsize=20;
   crsdef.distance=200;
   crsdef.rotation=0;
   crsdef.fs_density = 2e-2;
   crsdef.fs_min_r = 2;
   crsdef.fs_max_r = 6;
   crsdef.fs_min_coe = 0.9;
   crsdef.fs_max_coe = 1.1;
   *sam = new CPeriodicCrossSample(&crsdef);
   try {
      CSample *sample;
      sample = new CPeriodicCrossSample(&crsdef);
      sample->move_by(CVector(70,70));
      *sam = sample;
   }
   catch (int exc){
      if (exc == AIG_EX_FEATURE_OVERLAP)
 cout << "Bad configuration, features overlap" << endl;
   }
}

void generate_rectangle_structure_image(CSample **sam){
   t_rct_definition rctdef;

   rctdef.sizex=SIZEX;
   rctdef.sizey=SIZEY;
   rctdef.ee_coefficient=0.2;
   rctdef.ee_top_above_base=0.5;
   rctdef.base_level=0.3;
   rctdef.base_level_variation=0.2;
   rctdef.lsize=300;
   rctdef.tsize=200;
   rctdef.rotation=30*3.1416/180;
   rctdef.fs_density = 2e-2;
   rctdef.fs_min_r = 2;
   rctdef.fs_max_r = 6;
   rctdef.fs_min_coe = 0.9;
   rctdef.fs_max_coe = 1.1;
   *sam = new CSingleRectangleSample(&rctdef);
}


void generate_snake_structure_image(CSample **sam){
   t_rct_definition rctdef;

   rctdef.sizex=SIZEX;
   rctdef.sizey=SIZEY;
   rctdef.ee_coefficient=0.2;
   rctdef.ee_top_above_base=0.5;
   rctdef.base_level=0.3;
   rctdef.base_level_variation=0.02;
   rctdef.lsize=1200;
   rctdef.tsize=50;
   rctdef.rotation=0*3.1415927/180;
   rctdef.fs_density = 2e-2;
   rctdef.fs_min_r = 2;
   rctdef.fs_max_r = 6;
   rctdef.fs_min_coe = 0.9;
   rctdef.fs_max_coe = 1.1;
   *sam = new CSnakeSample(&rctdef);
}
//====END SAMPLE PARAMETERS=====

void message_call_back(t_message *msg){
   // this is the call-back function that will print messages
   cout << msg->sender_id << ": " << msg->message << " and comments:
   " << msg->comment << endl;
}


int main(){
   srandom(time(0));

   CApp app; // initialization of the generator
   app.set_message_call_back(message_call_back);
   // set the call-back function
   CSample *sam = NULL;

   //generate_snake_structure_image(&sam);
   //generate_rectangle_structure_image(&sam);
   //generate_corner_structure_image(&sam);
   generate_cross_structure_image(&sam);
   //generate_gold_on_carbon_image(&sam);
   CImage im(SIZEX,SIZEY);


   CWavyBackgroud back(0.05,0.25,5,5);
   // pars: min-bg-level, max-bg-level, x-density, y-density
   back.apply(&im); // paint the background pattern

   sam->paint(&im);
   delete sam;

//===============DEFINE TIME INFO ============================

   t_vib_definition vibdef;
   vibdef.pixel_dwell_time = 1;
   // all that matters here is the dwell time,
   // the rest is junked once we define drift vector
   vibdef.min_frequency = 0.0;
   vibdef.max_frequency = 0.0;
   vibdef.max_amplitude = 0;
   vibdef.number_of_frequencies = 0;
   vibdef.pixel_dead_time = 0;
   vibdef.line_dead_time = 0;


//================DRIFT VECTOR PARAMETERS =====================
vector<float*> drift;
float d1[4]={50.0,0.0,1200.0,0.0}; // {Ampx Ampy freq x-y-phase}

//float d2[4]={0.2,0.49,150.0,0.572}; in case I need a 2nd vector

drift.push_back(d1);
//drift.push_back(d2);
//tack on the second set of drift parameters into a new vector slot.

//========= VIBRATION FUNCTION =================================

   CVibration vib(&vibdef, drift);

//=========== BLURRING PSF ================

   CGaussianPsf psf(SIZEX,SIZEY,0.7,1,10);
   // pars: sizex, sizey, sigma, astig. ratio, astig. angle.
   psf.apply(&im); //applied to image template
//=========== GAUSSIAN NOISE ================

   CGaussianNoise gn(0.03); // pars: sigma

//=========== APPLY THE VIBRATIONS TO EACH IMAGE================

#define DWELL 1
#define CROP 50

/*
   psf.apply(&im); // apply the blur
   gn.apply(&im); // apply the gaussian noise
   im.crop(CROP,CROP,SIZEX-CROP,SIZEY-CROP);
   // pars: left-crop-margin, top-crop margin, right.., bottom.
   im.tiff_write("im0.tiff","Generated by ArtImaGen", BI_8B);
    // pars: file name, file comment, BI_8B for 8-bit image, BI_16B
// for 16-bit-image output.
```

```
*/

   CImage im3(&im);
   im3.crop(CROP,CROP,SIZEX-CROP,SIZEY-CROP);
   // pars: left-crop-margin, top-crop margin, right.., bottom.
   im3.tiff_write("im30.tiff","Generated by ArtImaGen", BI_8B);
    // pars: file name, file comment, BI_8B for 8-bit image, BI_16B
// for 16-bit-image output.
   gn.apply(&im3);
   im3.tiff_write("im0.tiff","Generated by ArtImaGen", BI_8B);

for (int i=0; i<29; i++)
   {
      CImage im2(&im);
      double FRAMERAND = 0.5*(double)rand()/RAND_MAX;
      vib.apply(&im2,(i+FRAMERAND)*SIZEX*SIZEY*DWELL);
      im2.crop(CROP,CROP,SIZEX-CROP,SIZEY-CROP);
      //CPoissonNoise pn(50.3);
      //pn.apply(&im2);
      //CGaussianNoise gn(0.01);
      gn.apply(&im2);

      char fn[200];  //start writing the image with unique name
      sprintf(fn, "im%d.tiff", i+1);
      im2.tiff_write(fn,"Generated by ArtImaGen", BI_8B);
   }
}
```

Bibliography

[1] J. S. Villarrubia, A. E. Vladár, and M. T. Postek, "Scanning electron microscope dimensional metrology using a model-based library," *Surface and Interface Analysis*, vol. 37, p. 951958, Nov. 2005.

[2] J. C. Leu, "Templated self-assembly of sub-10 nm quantum dots," master's thesis, MIT, May 2008.

[3] K. Berggren and V. Goyal, "Eliminating vibration from scanning microscopy by signal processing," tech. rep., MIT, Quantum Nanostructures and Nanofabrication Group, Oct. 2008.

[4] M. A. Sutton *et al.*, "Metrology in a scanning electron microscope: theoretical developments and experimental validation," *Measurement Science and Technology*, vol. 17, pp. 2613–2622, Oct. 2006.

[5] P. Cizmar, A. E. Vladár, and M. T. Postek, "Real-time scanning charged-particle microscope image composition with correction of drift." `http://arxiv.org/abs/arXiv:0910.0213v2`, Oct. 2009.

[6] B. Zitová and J. Flusser, "Image registration methods: a survey," *Image and Vision Computing*, vol. 21, pp. 977 – 1000, Oct. 2003.

[7] T. C. Chu, W. F. Ranson, and M. A. Sutton, "Application of digital-image-correlation techniques to experimental mechanics," *Experimental Mechanics*, vol. 25, pp. 232–244, Sept. 1985.

[8] Z. Xu *et al.*, "Drift and spatial distortion elimination in atomic force microscopy images by the digital image correlation technique," *The Journal of Strain Analysis for Engineering Design*, vol. 43, no. 8, pp. 729–743, 2008.

[9] M. A. Sutton *et al.*, "Scanning electron microscopy for quantitative small and large deformation measurements part I: SEM imaging at magnifications from 200 to 10,000," *Experimental Mechanics*, vol. 47, pp. 775–787, Dec. 2007.

[10] M. A. Sutton *et al.*, "Scanning electron microscopy for quantitative small and large deformation measurements part II: Experimental validation for magnifications from 200 to 10,000," *Experimental Mechanics*, vol. 47, pp. 789–804, Dec. 2007.

[11] P. Cizmar, A. E. Vladár, and M. T. Postek, "Advances in modeling of scanning charged-particle-microscopy images." `http://arxiv.org/abs/1006.0171v1`, May 2010.

[12] H. Smith and K. Berggren, "Nanostructure fabrication." Draft from book manuscript, Feb. 2010.

[13] J. Goldstein *et al.*, *Scanning Electron Microscopy and X-ray Microanalysis.* Springer, third ed., 2003.

[14] L. Reimer, *Scanning Electron Microscopy: Physics of Image Formation and Microanalysis*, vol. 45 of *Optical Sciences.* Springer, second ed., 1998.

[15] P. R. Bevington and D. K. Robinson, *Data Reduction and Error Analysis for the Physical Sciences.* McGraw-Hill, third ed., 2003.

[16] J. S. Lim, *Two-Dimensional Signal and Image Processing.* Signal Processing, Prentice Hall, 1989.

[17] P. Vandewalle, S. Süsstrunk, and M. Vetterli, "A frequency domain approach to registration of aliased images with application to super-resolution," *EURASIP Journal on Applied Signal Processing*, pp. 233–239, Jan. 2006.

[18] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing.* Signal Processing, Prentice Hall, second ed., 1999.

[19] P. Cizmar *et al.*, "Simulated SEM images for resolution measurement," *Scanning*, vol. 30, pp. 381–391, Sept. 2008.

[20] P. Cizmar, A. E. Vladár, and M. T. Postek, "Optimization of accurate SEM imaging by use of artificial images," in *Scanning Microscopy 2009*, vol. 7378 of *Proceedings of SPIE*, May 2009. 737815.

[21] Z. Liua and R. Laganière, "Phase congruence measurement for image similarity assessment," *Pattern Recognition Letters*, vol. 28, pp. 166–172, Jan. 2007.

[22] G. P. Lorenzetto and P. Kovesi, "A phase based image comparison technique," in *DICTA*, pp. 52–56, The Australian Pattern Recognition Society, Dec. 1999.

[23] J. USA, *A Guide to Scanning Microscope Observation.* JEOL, May 2006. `http://www.jeolusa.com/tabid/320/DMXModule/692/EntryId/1/Default.aspx`.

[24] Nano Technology Systems Division, *Instruction Manual: Orion Plus^{TM}Scanning Ion Microscope.* Carl Zeiss SMT Inc., Peabody, MA, Sept. 2008.

[25] S. Chang *et al.*, "Nanoscale in-plane displacement evaluation by AFM scanning and digital image correlation processing," *Nanotechnology*, vol. 16, pp. 344 – 349, Apr. 2005.