Surface Reconstruction From AFM and SEM Images

by
Adam Seeger

A dissertation submitted to the faculty of the University of North Carolina at Chapel Hill in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Computer Science.

Chapel Hill
2004

ABSTRACT

Adam Seeger
Surface Reconstruction From AFM and SEM Images
(Under the direction of Russell M. Taylor II)

Current methods for surface reconstruction from AFM images do not enable one to incorporate constraints from other types of data. Current methods for surface reconstruction from SEM images are either unsuitable for nanometer scale shapes or limited to shapes described by a small number of parameters.

I have developed a new approach to surface reconstruction from combination AFM/SEM images that overcomes these limitations. A dilation model is used to model AFM image formation and a filter bank model is used to model SEM image formation. I construct noise models for both AFM and SEM images from real data. The image formation models including the noise descriptions are used to construct an objective function expressing the probability of observed images given a hypothetical surface reconstruction. The surface is modeled as a sum of Gaussian basis functions and I derive a formula to estimate the gradient of the objective function in the surface parameter space. The conjugate gradient method is used to optimize the surface parameters.

My thesis is that this algorithm is more general and accurate than existing methods and that the gradient-based optimization based on my formula enables one to compute a reconstruction is a few hours. This thesis is demonstrated by applying the algorithm to real and synthetic examples.

ACKNOWLEDGEMENTS

I would like to thank the following people for their help with this project:

- My advisor Russell M. Taylor II for inviting me to join the nanoManipulator project and giving me the opportunity to work on so many fun projects, supporting and guiding my research, teaching me about how to build tools for science, and for helping me with so many aspects of the PhD process
- My committee members Stephen Pizer, Richard Superfine, Joe Costello, and Sarang Joshi for sharing their wisdom and for inspiring discussions
- Jon Tolle for discussions about optimization methods
- The Nanometer-Scale Metrology Group at NIST, John Dagata, Mike Postek, John Villarrubia and Andras Vladar, for helpful discussions and for providing me with the MONSEL source code and samples of silicon structures used in testing my algorithm.
- Members of the Laboratory for Photochemistry & Spectroscopy in the chemistry department at KU Leuven, Frans De Schryver, and in particular Philippe Foubert and Michel Martin for giving me interesting problems to work on and for helping me learn to use an AFM
- My colleagues and fellow students in the science part of the Nanoscale Science Research Group for helpful discussions, advice and collaboration, and in particular Aarish Patel, Phillip Williams, Stergios Papadakis, Onejae Sul, Mike Falvo, Garrett Mathews, Atsuko Negishi, Michael Staderman, Tim Meehan, and Sean Washburn for sharing time on their microscopes, help with acquiring AFM and SEM images and testing my software, and Adam Hall for giving me one of his carbon nanotube AFM tips
- My colleagues and fellow students in the computer science part of the Nanoscale Science Research Group for helpful discussions, advice and collaboration, and in particular Aron Helser, Alexandra Bokinsky, Chris Weigle, Tom Hudson, Charalampos (Haris) Fretzagias, Chris Dwyer and Jun Chen
- My teachers in the computer science department and in particular Guido Gerig, Gary Bishop, Greg Welch, and Anselmo Lastra for teaching me and encouraging me to learn more about computer vision and graphics and techniques that helped me to complete this project
- My supervisor at Intel, Horst Haussecker, for encouraging me to develop a new approach to SEM image simulation

- My parents, Judith and Joshua Seeger, for their encouragement and support of my education
- My wife, Sukyung, for her love and support

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS AND SYMBOLS

AFM = atomic force microscope

SEM = scanning electron microscope

SE = secondary electron

BSE = backscattered electron

V = volts

eV = electron-volts

keV = kilo electron-volts

$\otimes$ =convolution operator

$\oplus$ = grayscale dilation operator

$\ominus$ = grayscale erosion operator

$\circ$ = opening operator ($A \circ B = (A \ominus B) \oplus B$)

$\bullet$ = closing operator ($A \bullet B = (A \oplus B) \ominus B$)

$S(x,y)$ = SEM image defined for x=1..N_x, y=1..N_y

$A(x,y)$ = AFM image defined for x=1..$N_{x,AFM}$, y=1..$N_{y,\mathrm{AFM}}$

$H(x,y)$ = true height of the specimen surface in SEM coordinates defined for x=1..N_x, y=1..N_y

$T(x,y)$ = true height of the tip reflected about the origin ($T(x,y) \rightarrow -T(-x,-y)$)

$\hat{T}(x,y)$ = estimate of $T(x,y)$

$H_k(x,y)$ = estimate for $H(x,y)$ at scale σ_k in SEM coordinates

$h_k(x,y)$ = incremental surface at scale σ_k in SEM coordinates

$H_k'(x,y)$ = estimate for $H(x,y)$ at scale σ_k in AFM coordinates

$H_A'(x,y)$ = height found by eroding AFM image ($H_A' = A \ominus \hat{T}$)

$H_A\ (x,y)$ = height found by eroding AFM image and resampling onto the SEM image

$G_n\left(x, y\right)$ = gradient magnitude image for neighborhood size n

$\kappa_{+,n}\left(x, y\right)$ = maximum principal curvature image for neighborhood size n

$\kappa_{-,n}\left(x, y\right)$ = minimum principal curvature image for neighborhood size n

$K^F_{m,n}$ = convolution kernel used to compute cubic facet model coefficient *m* for a neighborhood size *n*

FB_{BSE} = BSE yield estimated by the filter bank method

FB_{SE} = SE yield estimated by the filter bank method

MC_{BSE} = BSE yield estimated by the Monte Carlo method

MC_{SE} = SE yield estimated by the Monte Carlo method

$S_{\mathrm{MC}}(x,y)$ = simulated SEM image by Monte Carlo method for a height field H

$S_{\mathrm{FB}}(x,y)$ = simulated SEM image by filter bank method for a height field H

$S_A\ (x,y)$ = SEM image simulated from the eroded and resampled AFM image

$A_k(x,y)$ = simulated AFM image by dilation operator for a height field H_k. ($A_k = H_k \oplus \hat{T}$)

$C_k(x,y)$ = parameters of surface reconstruction (Gaussian function coefficients) at scale σ_k defined for x=1..N_x, y=1..N_y

G^F_k = Gaussian function with standard deviation σ_k

$E_{\mathrm{AFM}}(x,y)$ = residual image for AFM using dilation model = $A(x,y)$- $A_k(x,y)$

$E_{\mathrm{SEM}}(x,y)$ = residual image for SEM using filter bank model = $S(x,y)$- $S_{\mathrm{FB}}(x,y)$

$\mathbf{T}_{S\to A}$ = 2D-2D transformation that transforms points in the SEM coordinate system into the corresponding points in the AFM coordinate system

1 Introduction

The atomic force microscope (AFM) and scanning electron microscope (SEM) are commonly used to study the structure of specimens at the nanometer scale. The two types of images are complementary because the AFM is good at measuring height in some places but is poor at measuring sharp edges while the SEM cannot measure height but is very good at detecting sharp changes in height or slope of the surface [Russell01]. Typically, very little processing is applied to raw AFM and SEM images, and humans must interpret the meaning and perform any integration of information from multiple sources. As the scale of the specimen decreases, the relation between the raw data and the specimen shape becomes increasingly complex and more difficult for a human to interpret accurately. As the scale of specimens continues to decrease, new imaging technologies may be developed that can resolve smaller structures but there will always be a need to push the small-scale limit at which quantitative shape information can be extracted from image data. The purpose of this project is to show how physically-based models of image formation can be used to derive more accurate quantitative shape information from images. It aims specifically at the case where there is more than one type of image available for the same specimen.

1.1 Assumptions and Approach

Determining a model of a physical specimen from physical measurements of that specimen is termed an "inverse" problem. The associated "direct" problem is to predict measurements from the model. An inverse problem is considered well-posed if a solution exists, is unique, and is stable (small changes in measurements do not lead to unbounded changes in the model) [Sabatier00]. In my problem (determining a specimen surface with nearly maximum likelihood given AFM and SEM data) uniqueness is not required but stability is important. I seek one of many reasonably likely surfaces given the available data and the reconstruction should be expected to vary from the true surface inasmuch as the available data does not constrain the solution. Stability is aided in my method using the scale-space reconstruction approach discussed later in this section and described in [Jones94].

The direct problem of computing the expected AFM image and SEM image given a model for a specimen is reasonably well understood although existing methods were not fast enough for the purpose of this dissertation. Much less is known about how to solve the inverse problem (automatically computing the best-fit model given one or more images) or what additional constraints would be required to make the solution unique and stable. I use physically-based models for image formation together with optimization to solve this inverse problem, computing an accurate specimen model from the combination of a single AFM image and a single SEM image. The relation between the two images and the shape of the surface is illustrated in simplified form in Figure 1-1.

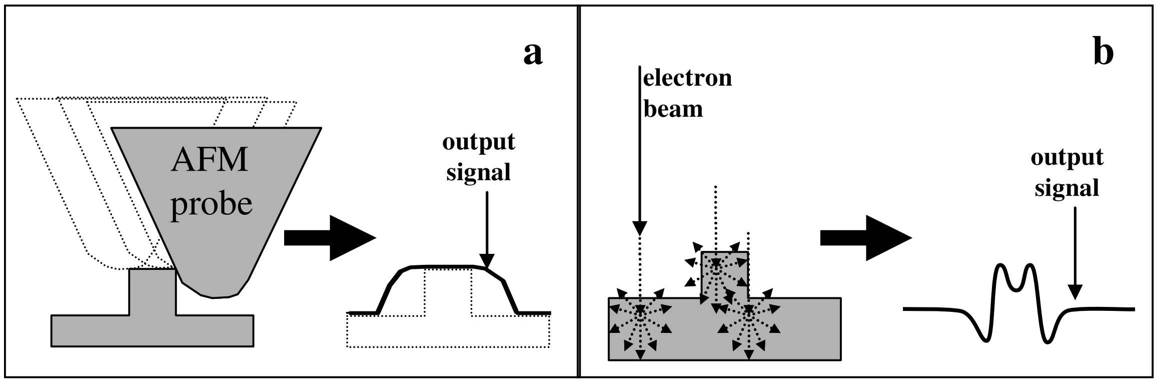


Figure 1-1: Simplified view of imaging mechanisms. (a) The AFM acquires a distorted topographical image of the surface. (b) The SEM projects a beam of electrons into the surface and measures how many come back out after scattering within the solid material.

In my approach, the accuracy of a specimen model is judged by an objective function based on likelihood (the probability of the observed images given the specimen model). The calculation of likelihood takes into consideration estimates of the noise in both images and, by estimating the sensitivity of the direct solution to surface shape, automatically determines how much influence each image should have in determining the shape of the surface at each location.

The method presented here assumes that both images are taken from directly overhead. It also assumes that the specimen surface can be described by a height function, $z=H(x,y)$.

This restriction excludes surfaces with undercuts, whose shape cannot be uniquely determined by overhead images. It also assumes that the specimen is composed of a single known material because it was easiest to implement a fast algorithm for simulating SEM images in this case. Particularly in the case of manufactured specimens, it is reasonable to assume such knowledge of the material. The model can be extended to the more general case of multiple materials at the expense of computation speed.

The algorithm models the specimen using a linear combination of hierarchical Gaussian basis functions of decreasing widths arranged on square grids. It searches for optimal coefficients of the basis functions at each width separately from the widest to the narrowest. Once the optimal coefficients for one width are found, it proceeds to the next smaller width. At each stage, the candidate surface is the sum of all weighted basis functions from the current stage and those from the previous stages. This approach, called scale-space reconstruction, was described in [Jones94]. The surface parameters (basis function coefficients) are adjusted using a deterministic iterative optimization procedure.

1.2 Thesis Statement and Contributions

My thesis is that

> *A more general and accurate method for maximum likelihood scale-space reconstruction combining both AFM and SEM images is presented. Unlike existing methods, it provides an integrated approach, combining AFM and SEM images into a single reconstruction objective function that weights the significance of each image through consideration of noise and the ambiguities inherent in each imaging process. Completion of a reconstruction within a few hours is enabled by an optimization approach described in [Jones94] and calculations of likelihood gradient based on an existing model (grayscale dilation) for AFM image formation and a new model (filter bank synthesis) for SEM image formation.*

I demonstrate this thesis using both synthetic and real microscope images.

For the test involving real images, I reconstruct the specimen surface using a real SEM image and a real AFM image taken with a relatively dull tip. The accuracy of this reconstructed surface is then judged by the sum of squared differences in height between that surface and a reconstructed surface from an AFM image taken with a much sharper tip.

I also reconstruct the surface from the dull-tip AFM image alone and compare this with the sharp-tip AFM-based reconstruction.

For the synthetic data test I simulate AFM and SEM images from a synthetic surface and attempt to reconstruct that surface from both images and from the AFM image alone. Results based on synthetic data can be compared to a known true sample, which is not possible in the case of real specimens.

One important sub-problem is the initial registration of AFM and SEM images. Because the AFM image gives an approximation to the specimen topography, knowledge of how the images are related can be incorporated by simulating an SEM image for the specimen surface estimated from the AFM image and directly comparing intensities in the simulated SEM image with the actual SEM image as described in section 7.2.

In completing this project I have made the following contributions:

- novel reconstruction objective function combining AFM and SEM image likelihood enabling gradient-based optimization of a surface
- software library for Monte Carlo SEM simulation that includes an optimized parallel algorithm that achieves up to 1500x speedup on a 32 processor machine over the previous sequential implementations
- novel automatic registration method for AFM and SEM images
- filter bank technique for constructing generative image models for surfaces represented by height fields
    - application of this technique to the simulation of SEM images
- software application with numerous features useful for AFM/SEM analysis:
    - blind tip-reconstruction from AFM images and manually-driven tip modeling utility
    - spatially-referenced comparison of AFM and SEM images and cross-sections of those images
    - SEM simulation using Monte Carlo and shape-based methods
    - faster filter-bank SEM simulation
    - graphics-hardware-accelerated discrete grayscale dilation and erosion operations useful for reconstruction from AFM and simulation of AFM images
    - manual and automatic image registration with overlaid display
    - visualization of surfaces using 3D computer graphics

The specific problem of reconstruction from AFM and SEM images is a particularly challenging member of the class of inverse problems with multimodal image data. By demonstrating that detailed models of image formation can be used to solve this specific inverse problem I have provided an example that can motivate the application of this approach to reconstruction from other types of images.

Existing methods for surface reconstruction from SEM are based on physically-based simulation of SEM images. As I describe in section 2.2.1, because this simulation is very slow it is only practical for very small and highly constrained problems. I have shown how a phenomenological model for image formation can be fit to the physically-based simulation to enable the solution of much larger and less constrained problems. This is a general approach that could be used to model other types of images and enable solution of other inverse problems.

1.3 Algorithm Summary

My approach uses the maximum likelihood method [Hoel71] with new AFM and SEM image estimation functions tuned for accuracy and speed. These simulating functions are combined into a single likelihood model that estimates the probability that the observed AFM and SEM images are noisy versions of simulated images from a hypothetical specimen and imaging conditions. The observed images are fixed and parameters describing the specimen surface and imaging conditions are varied. In the maximum likelihood method, one estimates the true specimen by maximizing this likelihood function. Because there is no closed-form solution for the maximum, it is necessary to use an iterative optimization algorithm. I use an optimization method that iteratively updates a model of the surface based on the gradient of the log-likelihood function (logarithm of the likelihood function). Given a hypothetical specimen surface, I estimate the log-likelihood and its gradient using models of image formation and noise distributions that will be described in the following chapters.

Imaging conditions, including the shape of the AFM probe, the coordinate transformation between the AFM and SEM images, and SEM detector parameters, are estimated before optimizing the likelihood. Alternatively, one might include these typically unknown parameters with the specimen surface parameters as part of a single optimization. To simplify the optimization I chose to find the probe shape and coordinate transformation independently.

1.4 Introduction to the AFM

The Atomic Force Microscope (AFM) was invented in 1986 by Binnig, Quate, and Gerber at the IBM Zurich Research Laboratory [Binnig86]. Its mechanism is very similar to that of the Scanning Tunneling Microscope (STM) invented by Binnig and Rohrer in 1982 [Binnig82] and earlier stylus profiler devices. An AFM probes a specimen surface with a sharp tip mounted on the end of a cantilever (Figure 1-2). The flexible cantilever bends in response to force between the tip and specimen. A beam of light from a laser is reflected off the cantilever and onto a quadrant photodiode. As the force between the specimen and tip changes, the cantilever bends. The bending cantilever changes the position of light hitting the photodiode, thereby causing a change in the electrical current produced by the photodiode. The photodiode current thus provides a very sensitive measurement of the force between the tip and specimen. If the cantilever is lowered (z direction) past the point where the tip touches the surface, the cantilever will bend upwards and the photodiode will measure the normal force between the tip and surface. If the cantilever is then translated in x or y, the cantilever may twist and the resulting force signal includes some component due to lateral force between the tip and surface. By taking different combinations of the 4 parts of the photodiode one can separate out the lateral and normal force components for a particular scan direction.

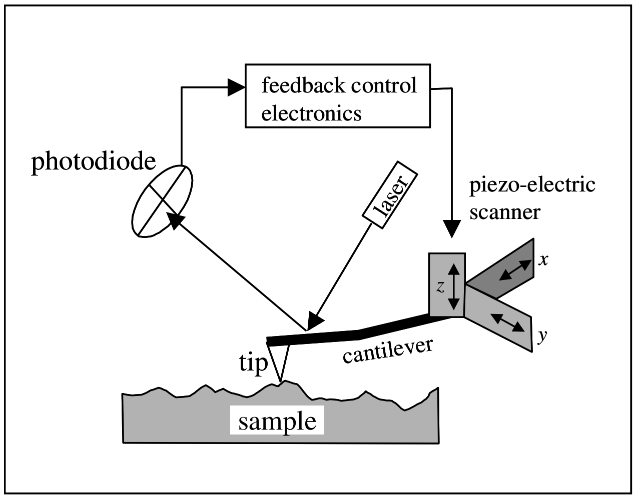


Figure 1-2: Parts of an AFM

The simplest way to measure a surface with an AFM is to record the variations in force between the tip and surface as the tip is scanned in x and y. If the surface is not very flat or level, this can result in large forces that will damage either the tip or specimen. A more practical approach is to scan the surface in x and y while maintaining a constant normal force. Constant force (or *contact mode*) imaging uses negative feedback to adjust tip height by controlling the voltage on the z-piezo positioning element as the force on the tip deviates from a user-determined set point. The voltage applied to the z-piezo is directly related to the height of the tip (modulo hysteresis) so it is possible to convert the z-piezo signal into a measure of the height of the surface.

Another method called *non-contact* mode involves oscillating the cantilever at its resonant frequency and measuring the variation in the amplitude of oscillation as the tip is scanned over the surface. The feedback control system maintains a constant decrease in the amplitude of oscillation relative to the amplitude when the tip is far from the surface. This method has the advantage of only intermittently contacting the surface and thereby avoids much of the specimen and tip damage that can occur with contact mode imaging. Lateral force is much lower in non-contact or tapping mode than in contact mode. True non-contact imaging occurs when there is a purely attractive force between the tip and specimen and typically requires operating the AFM in ultra-high vacuum. When the tip is oscillated above the specimen in air, a water layer on the surface usually causes the tip to jump into intermittent contact. At the range of feedback values used in this study (40-50% amplitude reduction), the damping is caused by the tip striking the sample at each downstroke during its oscillation.

For the AFM images used in this project I used non-contact mode because it is less damaging to the AFM tip and therefore makes more valid the assumption that the tip has a constant shape for each image. It also reduces complicated artifacts that might be caused by lateral forces. After taking a set of AFM images with a relatively sharp tip I used the AFM in contact mode with a large force to intentionally dull the tip. The dulled tip was then used to produce the dull tip test images mentioned above for demonstrating my thesis.

The largest artifacts in AFM images are caused by tip shape and imperfections in the feedback control system. The ideal AFM would have an infinitely sharp tip to reach as much of the surface as possible and an infinitely sharp impulse response in its feedback system to instantly adjust the height of the tip as it is scanned over the surface. In reality, the tip has a pyramidal or conical shape with some finite end radius so it is durable enough to withstand the surface interaction forces. Also, a real feedback system requires finite time to adjust the height of the cantilever in response to changes in topography. Limitations due to imperfect feedback can be eliminated by sufficiently slowing down the speed with which the *x*-*y* piezos scan the surface – effectively reducing the rate at which the height of the surface below the tip changes. However, this may introduce more severe lateral distortions of the image as the position of the sample relative to the tip gradually changes over time due to thermal expansion or contraction of parts of the microscope or specimen. The effects of the tip shape cannot be avoided and these result in characteristic tip-dilation artifacts. A more detailed description of models for forces between the tip and specimen is provided in Appendix A. Additional background material on mathematical morphology concepts useful for understanding tip-specimen interactions is provided in Appendix A.

1.5 Introduction to the SEM

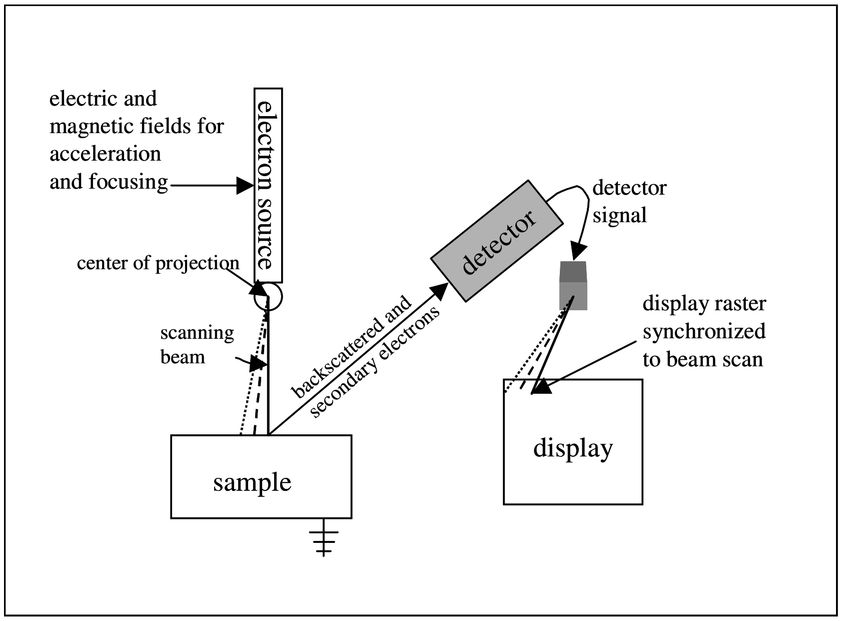

electric and magnetic fields for acceleration and focusing
electron source
center of projection
scanning beam
backscattered and secondary electrons
detector
detector signal
display raster synchronized to beam scan
display
sample

Figure 1-3: Parts of an SEM

The first Scanning Electron Microscope (SEM) was built by Max Knoll and Ernst Ruska in 1931. An SEM bombards a specimen with a sharply focused beam of electrons and detects the resulting electrons that scatter out of the specimen. An SEM uses magnetic lenses to focus the electron beam. Unlike the lenses in an optical microscope, the magnetic lenses serve to demagnify (i.e. make smaller) the image of the electron gun rather than to magnify the image of the specimen. Initially, the electron beam is on the order of tens of microns. The beam is focused to a very small spot on the order of several nanometers in diameter through a demagnification of about 1/5000 so that it can sample the specimen surface as finely as possible. Electromagnetic deflection coils raster this spot over the surface to create an image just like the beam in a cathode ray tube (CRT) is scanned over a phosphor-coated screen. In a CRT, a flat surface hit by the electrons displays an image; in a SEM the response of the surface hit by the electrons is measured and displayed as an image. The signal produced by electrons scattered out of the specimen is displayed on a CRT screen or acquired by a digital frame grabber that is scanned in synchrony with the scanning of the SEM beam over the surface. In this way, the projection for the image is determined by the electron beam direction. Typically, the beam will be deflected at some point about a centimeter above the specimen. The deflection point is the effective center of projection for the image. Because the scanned region on the surface may be several orders of magnitude smaller than the distance to the center of projection, the direction of the beam is nearly constant as it is scanned and the resulting projection is nearly orthographic. For a typical 10-micron scan range and 10 mm working distance, the change in angle is only 0.06 degrees over the entire scan.

The signal that determines the observed intensity in the SEM comes from the fraction of scattered radiation that can be sensed by a detector. One of the most popular types of detectors is the Everhart-Thornley (ET) detector because of its low cost and high sensitivity. The ET detector consists of a scintillator (material that converts energy from electrons to photons) and a photomultiplier tube (device for converting the energy from photons back into electrons and amplifying the electrical signal) and is sensitive enough to detect individual electrons. Changing the bias voltage on a metal screen in front of the detector can control the fraction of electrons that hit the ET detector. When the screen is biased to a very negative voltage (relative to the voltage at the specimen surface), it deflects slow moving (low energy) electrons but fast moving (high energy) electrons can still get through to the detector. When the screen is biased to a very positive voltage (relative to the voltage at the specimen surface), it deflects slow moving electrons into the detector, thereby collecting nearly all slow moving electrons coming out of the specimen in any direction. To understand what these different signals mean it is necessary to understand what happens below the surface to produce the emitted electrons.

Electrons emitted by the electron gun are accelerated through an electric field and the change in voltage as they pass from the gun to the specimen determines the energy of the electrons at the point that they start to interact with the specimen. The SEM images used in this work were taken with an accelerating voltage ranging from about 1.0kV to 3.0kV. When an electron enters the specimen, it can interact in two different ways with the atoms in the specimen. If it comes close enough to an electron that is already in the specimen (usually as part of one of the atoms in the specimen), then it can transfer some of its energy to that electron in what is called an inelastic interaction (as when one billiard ball hits another billiard ball). If an electron comes close enough to the relatively massive nucleus of an atom in the specimen then it will be deflected with almost no loss in energy in what is called an elastic interaction (as when a billiard ball hits a bowling ball). Typically, a high-energy electron moving through the specimen is gradually slowed down through numerous inelastic interactions, very few of which change its direction significantly. Elastic interactions occur much less often but are responsible for most of the change in direction. If an electron from the electron gun is deflected and comes out of the specimen, it is known as a back-scattered electron (BSE). Electrons that are part of the specimen but are kicked out through inelastic interactions are known as secondary electrons (SE). Because the origin of a particular scattered electron is not directly measured, scattered electrons are most commonly classified by their energy, which can be measured. The energy is typically expressed in units of electron volts (eV), the kinetic energy acquired an electron as it traverses a change in potential of 1 volt. If the electron gun is at a potential of -1000 volts relative to ground and the specimen is grounded, each electron in the beam will hit the specimen with 1000 electron volts or 1 kilo electron volts (keV). Electrons escaping the specimen typically have energy somewhat lower than the energy of the incident electrons and a single incident electron may generate multiple secondary electrons at much lower energy such that the number of electrons ejected is more than the number that were incident. Secondary electrons normally have a very low energy (below 50eV). Because they are so low in energy, secondary electrons cannot travel very far before they are stopped by atoms in the specimen. Consequently, the only secondary electrons that can be detected are those that are produced within a thin layer several nanometers thick near the surface. The most likely energy for SE is about 3-5 eV and the most likely energy for BSE is about 0.8-0.9 times the incident electron energy [Goldstein92]. By convention, electrons are classified as SE if they are below 50 eV and BSE if they are above 50 eV. Such a classification, given typical SE and BSE energy distributions, attempts to minimize the probability that an electron will be misclassified as SE or BSE.

There are actually three different types of secondary electrons, categorized according to where they are generated along the primary electron trajectory as seen in Figure 1-4. At the detector these different secondary electrons are indistinguishable but their origin affects the resolution of the measured signal. To a first approximation, the trajectory of an incident electron can be divided into two phases, before and after the first large-angle elastic scattering event. In the first phase, the only secondary electrons that are detected are those generated near the point of entry because others are too deep in the specimen to escape. It is unlikely that an incident electron is scattered by a large angle before penetrating too deeply in the specimen to produce detectable secondary electrons. In the second phase, the electron diffuses in a random walk and possibly escapes the specimen. If it escapes, additional detectable secondary electrons are generated at the point of escape. The escaping secondary electrons produced in the first phase (SE type I) are highly localized near the entry point of the primary beam electrons and therefore provide the highest resolution information. Escaping secondary electrons produced in the second phase (SE type II) are emitted over a region much larger than the beam diameter and therefore carry only low resolution information. When the beam is sharply focused, the resolution of SE type II electrons is typically much lower than that of SE type I electrons so the highest spatial frequency intensity variations in an image represent variations in the SE type I signal. At sufficiently high magnifications, the SE type II signal may not vary significantly across the entire image and will mainly contribute only an offset in the intensity. Secondary electrons are also produced when escaping electrons from the specimen hit the specimen chamber (SE type III). The SE type III signal is mainly due to BSE that do not travel directly to the detector but instead are detected indirectly by the secondary electrons they generate. Because of the SE type III mechanism, a secondary electron detector, even one that is biased to preferentially detect low energy secondary electrons, may detect the high energy BSE indirectly.

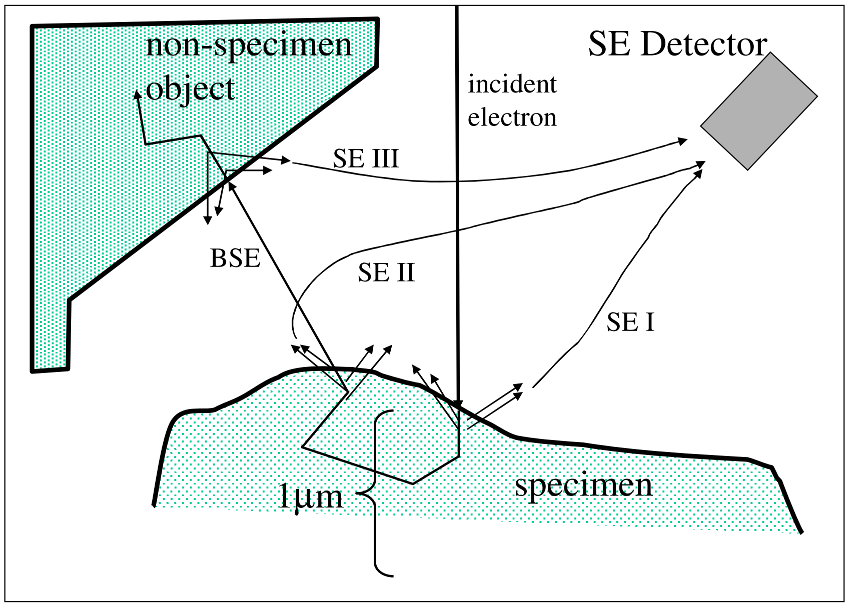


Figure 1-4: SEM beam-specimen interactions

Quantitative estimates of the beam-specimen interactions can be obtained by simulating many electron interactions in a Monte-Carlo simulation. Such a simulation takes the initial beam energy, specimen composition and specimen shape as input and produces estimates of the fraction of back-scattered and secondary electrons which would escape from the surface of the specimen (Figure 1-5).

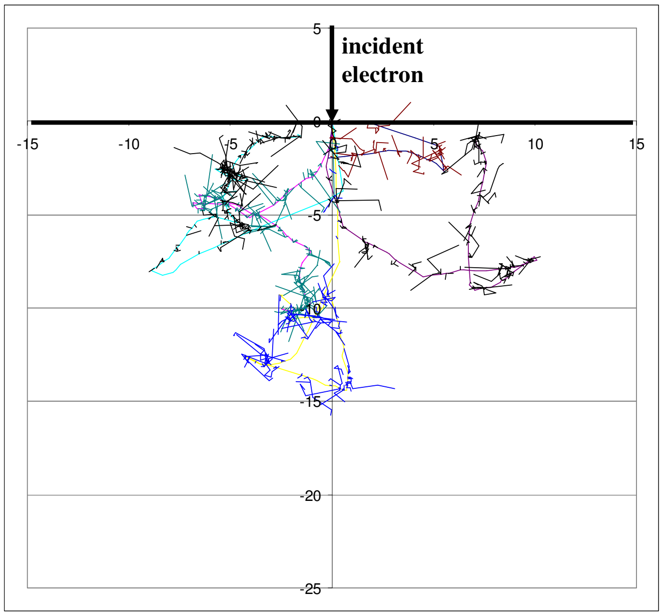


Figure 1-5: Graph of trajectories computed by Monte Carlo simulation of electrons striking a flat silicon specimen at the point (x=0,z=0)

As stated earlier, the projection for an SEM image is determined by the beam direction as the beam is scanned over the specimen. The SEM detector determines the observed intensity in the image but the location of the detector inside the SEM has no effect on the perspective for the image. Despite the strangeness of the SEM image formation mechanism, the images can be understood intuitively by analogy with images taken with a light camera (Figure 1-6). The proper analogy is that the scanning motion of the beam is like the optics of the camera and the detector is like the light source. The paths of the electrons in the SEM are like the paths of photons for the light camera except that the electrons travel in the opposite direction.

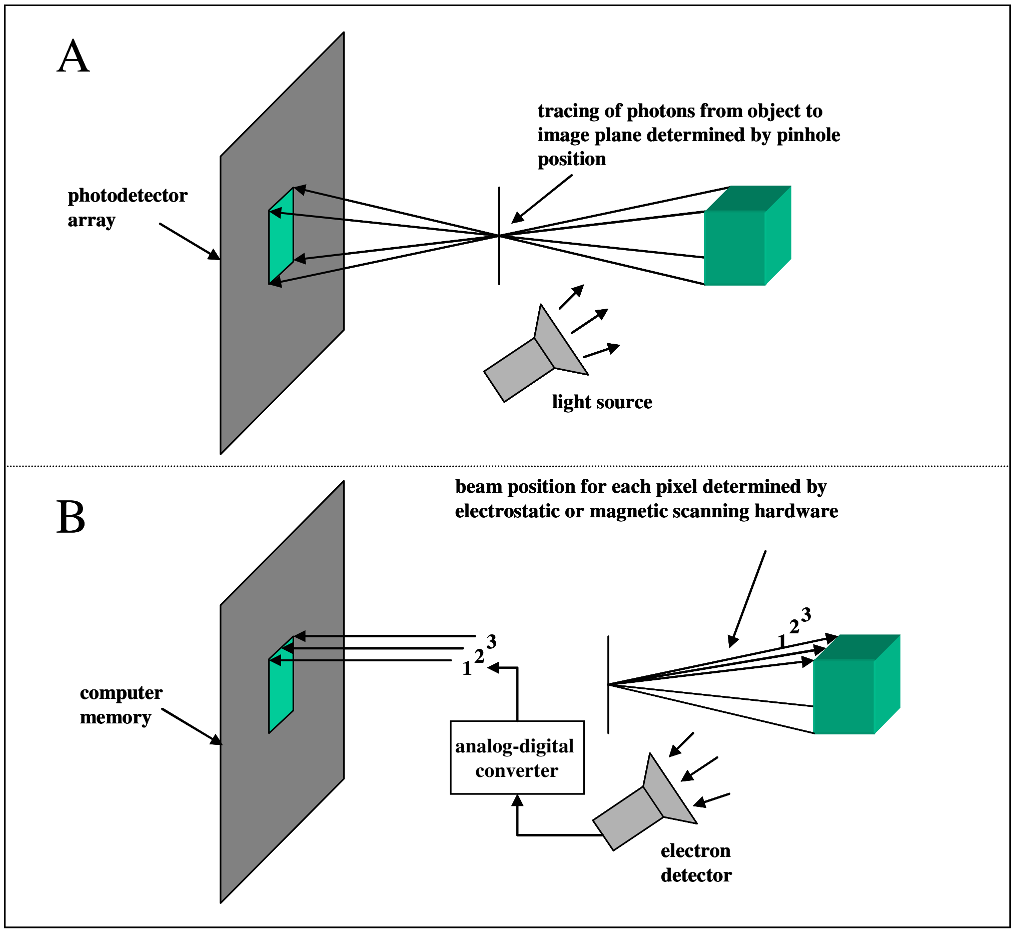

A
tracing of photons from object to image plane determined by pinhole position
photodetector array
light source
B
beam position for each pixel determined by electrostatic or magnetic scanning hardware
1
2
3
1
2
3
computer memory
analog-digital converter
electron detector

Figure 1-6: Analogy between a pinhole camera (A) and an SEM (B) in terms of viewing projection and contrast

Because electrons in the SEM travel below the specimen surface, they tell us something about volumetric qualities of the specimen. In the case of a surface covered by a thin film, it is possible for the SEM to "see through" the film to observe details of the covered surface. Another consequence of the subsurface scattering is that rough surfaces appear brighter than smooth surfaces. This change in contrast with roughness can be understood intuitively in terms of surface area. As the surface area increases, the volume of the thin layer from which secondary electrons can escape increases, providing more opportunities for secondary electrons to escape. Even though the topographic variations in the surface characterized by roughness may be too small to be resolved in an SEM image, they can still affect the overall intensity of the image. A related phenomenon occurs near sharp edges or small bumps where electrons scattering within the specimen effectively have more opportunities to escape the specimen and reach the detector. The result is characteristic brightening near the edges of raised topographic features (so-called edge effects) (shown in Figure 1-7 and Figure 1-8).

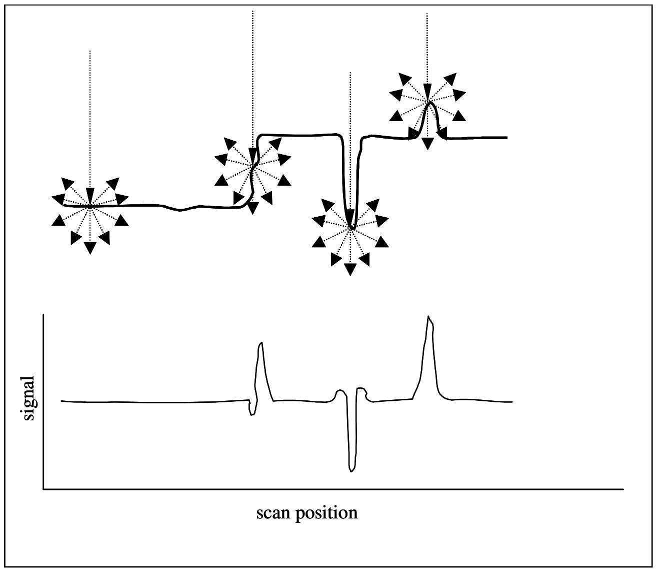


Figure 1-7: The distribution of material surrounding the point at which electrons enter the surface determines how easy it is for scattered electrons to escape the specimen. Edge effects are one of the observable manifestations of this relationship.

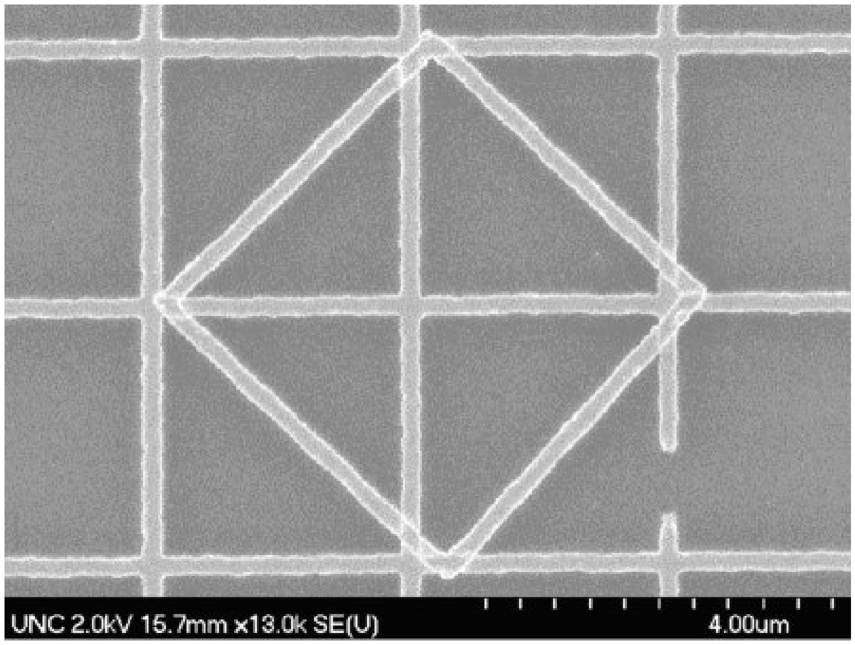


Figure 1-8: Example of SEM image showing edge effects

1.6 Summary of Later Chapters

Chapter 2 describes previous work that inspired and motivated this technique including methods for AFM-based surface reconstruction using concepts from mathematical morphology, methods for SEM-based surface reconstruction using Monte Carlo simulations and shape-from-shading. Chapter 3 describes some additional related inverse problems. Generative models for AFM and SEM are important tools on which many parts of my approach depend so I describe these models next in chapters 4, 5 and 6. I use a traditional approach to modeling the AFM image but extend the model to allow it to be more efficiently inverted in a maximum likelihood framework. I initially considered using Monte Carlo simulation to compute the SEM component of likelihood but determined that this was either too slow or too noisy to be practical. Instead I have developed a new deterministic model that is significantly more accurate than slope-based models and is also easily differentiated to enable calculation of likelihood gradient. Chapter 4 describes the model for AFM based on concepts from mathematical morphology. Chapter 5 describes a model for SEM based on Monte Carlo simulation of electron trajectories. Chapter 6 describes my new model for SEM based on a filter bank of convolution operators. Chapter 7 describes how the generative models are used to help with alignment of AFM and SEM images. Chapter 8 describes the noise models incorporated into my objective function where they determine how much each image influences the surface reconstruction. Chapter 9 describes my optimization method including specimen surface parameterization, and calculations of log-likelihood and its derivatives with respect to surface parameters. Chapter 10 presents results for surface reconstruction that are critical to demonstrating my thesis statement. Chapter 11 summarizes contributions, describes limitations of the reconstruction algorithm, and discusses areas for future work.

2 Previous Work

In this section I describe previous work on surface reconstruction from SEM and AFM images and how it relates to my approach.

2.1 Application of Morphological Operations to AFM Image Analysis

Earlier approaches to surface reconstruction based only on AFM data attempted to directly remove artifacts caused by the tip shape. These approaches require estimation of the tip shape either from the same AFM image used for the reconstruction or an independent measurement. The tip shape estimated from an AFM image is very sensitive to sharp noise spikes in the image but the effect of noise can be reduced by an ad hoc approach. Noise also has a less serious but somewhat odd effect on the surface reconstruction. The method of surface reconstruction using morphological operations is similar to replacing each pixel in an image by the minimum value over the local neighborhood. This tends to favor downward noise spikes, while upward noise spikes tend to be ignored. In this case the effect of noise is mainly to introduce a global offset of the reconstructed surface downward which is no problem because the AFM only measures relative height for the surface. There will also be some variation in the offset introduced by the noise that I expect to be somewhat reduced in my reconstruction because of the more principled approach to dealing with noise. I do not claim any significant practical contributions in this area but do contribute a more principled approach to surface reconstruction from noisy AFM images that more fairly considers all the data in an AFM image. My main contribution is the algorithm that enables use of additional information about shape such as that provided by an SEM image.

2.1.1 Dilation and Erosion

An AFM image is the result of scanning a mechanical probe over a specimen surface and the image contains features of both the specimen and probe shape. Grayscale dilation is a morphological operation that approximates the transformation from specimen surface to AFM image [Pingali92]. Dilation of a specimen surface by an AFM tip is illustrated in the top part of Figure 2-1. The dilation (dotted line in top figure) is the surface traced by a point on the tip as it is scanned horizontally and positioned vertically to where it just touches the specimen. To the extent that the AFM tip touches all parts of the specimen surface and the AFM image is approximated by a dilation operation (ignoring noise, surface deformation, and feedback artifacts), the specimen surface can be reconstructed using a morphological operation called grayscale erosion using an estimate of the tip shape. Erosion of an AFM image using an estimate of the tip shape is illustrated in the middle part of Figure 2-1. The erosion (dotted line in middle figure) is the surface carved out by the tip as it is translated to every point on the AFM image. Where the tip has not touched the surface, the eroded AFM image only provides an upper bound on the true specimen surface.

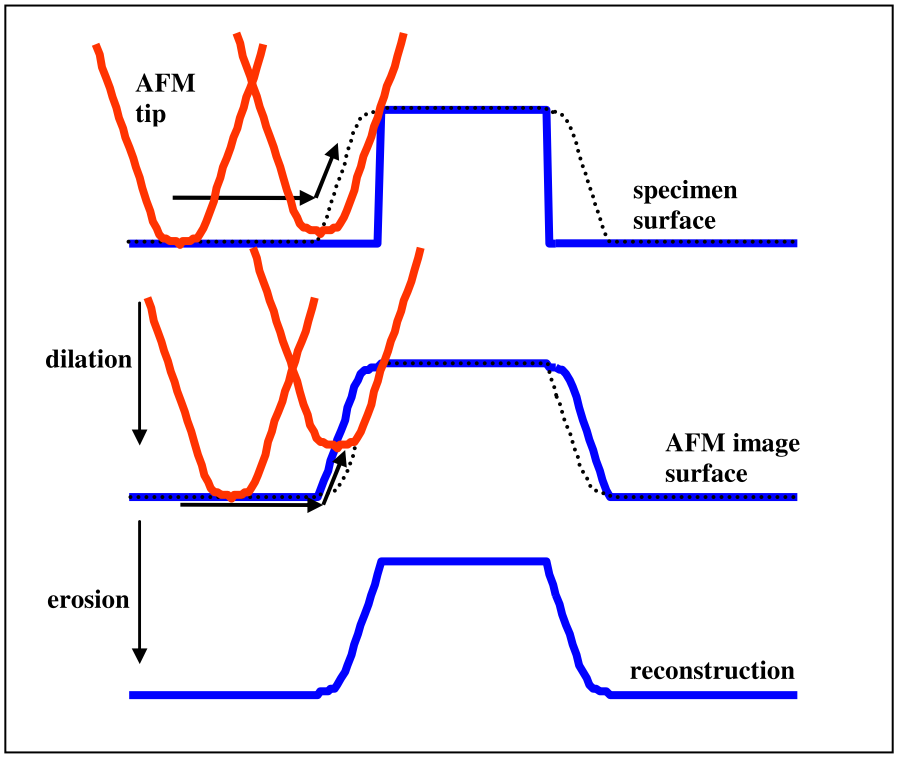


Figure 2-1: AFM image formation can be modeled as a dilation operation. An erosion operation can be used to partially reconstruct the specimen surface.

2.1.2 Estimating Tip Shape

One way to estimate the tip shape is to scan the tip over a known structure and erode the resulting image with the known shape to reconstruct the tip. This approach is somewhat impractical because it requires the time-consuming scan of a calibration structure and because parts of the tip can break off, changing its shape within or between scans (this is particularly common when the tip first engages the specimen surface). One is not guaranteed that the estimated tip shape is a good estimate of the shape of the tip during previous or subsequent scans of different specimens. Another approach reconstructs the tip from the image of the unknown specimen surface under study using a so-called "blind" tip reconstruction algorithm. Such an approach is enabled by constraints in the dilation model for the AFM image along with some assumptions about the maximum width of the projection on the image plane of the parts of the tip that touched the surface. In this section I first describe a version of the blind tip reconstruction algorithm that assumes that the AFM image is a pure dilation. Then I describe a modification to the algorithm to enable it to work on noisy images.

2.1.2.1 Blind Tip Reconstruction for Noiseless Images

A blind reconstruction algorithm for recovering an estimate of tip shape was described in [Villarrubia94] and [Williams96]. The advantage of this approach is that it enables recovery of the tip shape from an AFM image of an uncharacterized specimen. Informally, the basis for the algorithm is that any structure in the AFM image must have been scanned by a tip that is at least as sharp as that structure. The algorithm starts with an overly blunt tip shape and iteratively cuts away parts of the tip until it is consistent with the image.

Blind tip reconstruction is based on the fact that the dilation of a surface can opened (eroded and then dilated again) without any change if the probe shape for the opening operation has the same shape as the probe shape for the original dilation (Equation C-6). The blind tip reconstruction algorithm attempts to find the broadest probe that satisfies this property. While the idempotency of the opening operation is a necessary condition for the tip estimate to match the true tip, it is not sufficient and therefore it is possible for the estimate to only be an outer bound on the true tip shape. The estimate will not be sharper than the true tip shape because the algorithm finds the broadest tip that satisfies this property and so the actual tip shape will be preferred over any that are sharper.

One of the inputs to the algorithm is the size of the image representing the tip. Ideally this image should be just large enough to describe all parts of the tip that touched the specimen surface. If one were to choose the tip image size to be the same size as the AFM image, the final tip estimate would be the same as the AFM image (corresponding to the specimen being a sharp spike). The tip image size essentially limits the parts of the tip that can touch the surface so reducing the size forces the contact between tip and specimen to be more localized and this has an important effect on how the AFM image constrains the possible tip shapes. As the tip image size is reduced, the number of points in the AFM image affecting a pixel in the tip model will increase and provide a better estimate of the tip shape but making the tip image too small will prevent some parts of the tip farthest from the apex from being reconstructed.

2.1.2.2 Blind Tip Reconstruction for Noisy Images

Unfortunately, noise in the AFM image tends to create sharp spikes and if a pure dilation is assumed, these spikes imply that the tip shape must be very sharp to observe such features [Villarrubia97]. Some noise may be filtered from AFM images by using line-by-line flattening algorithms (removing constant offsets between lines) or a median filter (replacing each pixel value by the median value in the neighborhood of that pixel) [Villarrubia97]. Even after such filtering, noise can cause the tip estimate to be overly sharp. To fix this problem, the tip estimation procedure was modified by introducing a threshold parameter [Villarrubia97]. This parameter causes the tip to only be updated if the change in tip height at a point is greater than the threshold parameter. Furthermore, when updated, the change is reduced by the threshold. It is necessary to iterate over the points in the AFM image until no points result in an update to the tip. Experiments with synthetic noisy AFM images show that if the threshold is chosen to be about 3σ, where σ is the standard deviation for the noise, then the reconstruction will most closely match the true tip shape and will bound the true shape from the outside. Below this size, the reconstruction will be too sharp and above this size the reconstruction will broaden. Also, the transition from being too broad to being too sharp occurs very suddenly as the threshold value is reduced [Villarrubia97]. In [Dongmo00] and [Todd01], the threshold value is chosen slightly greater than the value at which the derivative of the tip volume with respect to the threshold value is maximized.

Another algorithm for blind tip reconstruction in the presence of noise is described in [Williams98]. In this algorithm, a positive and negative offset is added to the AFM image to produce two bounding surfaces representing the region of uncertainty for the ideal AFM image without noise. The algorithm is similar to the algorithm assuming no noise described in [Villarrubia94] and [Williams96] but the lower bounding surface is used to position the current AFM tip model at each pixel of the AFM image while the upper bounding surface is used to erode the tip model.

2.1.3 Surface Reconstruction from a Noisy AFM Image

Once a tip shape is found either by using a special tip characterizer or one of the blind reconstruction algorithms, the specimen surface can be reconstructed by grayscale erosion of the AFM image with the tip shape. While the erosion operator has an inherent smoothing effect on the input image that filters out some of the noise, it introduces artifacts and ignores some information in reconstructing the specimen topography (Figure 2-2).

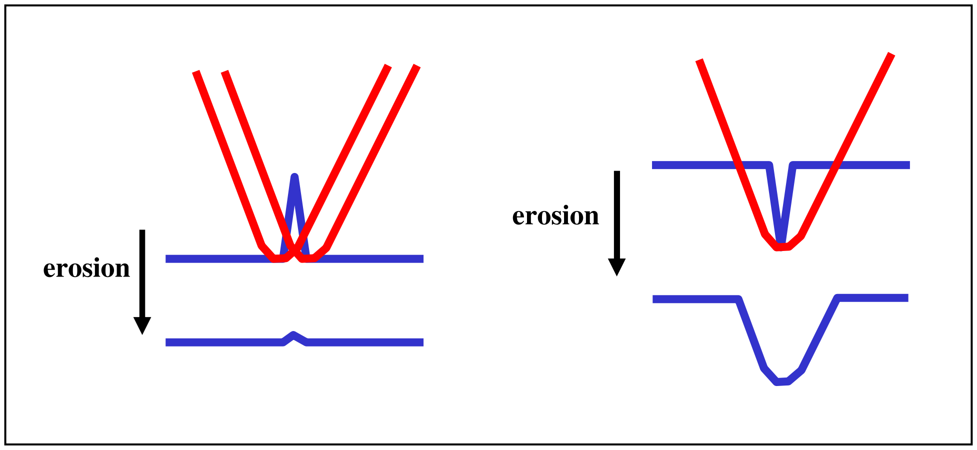


Figure 2-2: Effect of noise on erosion. Upward spikes in the AFM image surface (left) are removed or reduced in the eroded surface. Downward spikes (right) have a relatively large effect, creating large pits in the eroded surface and neighboring pixels may have no effect on the reconstruction.

Even if the exact tip shape were known, erosion of a noisy AFM image with this tip would produce only an approximation to the maximum likelihood reconstruction because of how the erosion operation to some extent ignores the value of certain data points (such as upward noise spikes or points neighboring downward noise spikes). My approach addresses this issue by seeking the surface that explicitly maximizes the likelihood of the entire AFM image assuming a particular tip shape. Instead of using the erosion operation to reconstruct the surface, I seek a surface with a corresponding dilation that deviates in the most likely way from the AFM image assuming a model for correlated noise in the AFM image. Starting from an initial estimate of the surface (that is optionally computed using the erosion operation), I adjust the surface in a way that leads to the greatest increase in likelihood of the difference between the dilated surface and the AFM image using the gradient of the likelihood with respect to the surface parameters. While previous work used ad hoc methods for dealing with noise in AFM images without any consideration of correlation of the noise, I construct a model for correlated noise in the AFM image and use this model in a principled way to find an optimal surface.

2.2 Models for SEM Image Intensity and Applications to Surface Reconstruction

When using SEM images for surface reconstruction, one must choose an appropriate SEM intensity model, which produces an intensity estimate at each location given a description of the specimen. This model is important because it determines the applicable optimization methods. Previous work from the computer vision field used a model based on surface slope that enables various shape-from-shading approaches as discussed in section 2.2.4.2. The Monte Carlo model described in chapter 5 is a more accurate approximation than models based solely on slope but is not as easily inverted and the direct problem of simulating images using the Monte Carlo model can be extremely time-consuming.

Modeling the SEM signal as a function of slope works in some special cases of surfaces that are very smooth at the scale of the electron probe. More precisely, the slope-based approximation works if the surface can be well approximated by a plane over the region of the surface from which electrons escape (for the electron beam focused at a single point on the surface). The slope-based approximation breaks down when a significant number of electrons contributing to the signal can escape through the surface where the slope is significantly different from the slope where the beam enters the surface. The electron escape zone ranges from tens of nanometers to several microns across depending on the material and initial electron energy. The size of the escape zone is a complicated function of topography so it is more useful to speak in terms of the straight-line distance electrons can travel in the material before they run out of energy. This electron range is a bound on the radius of the escape zone for a flat specimen. For the material (silicon) and energy (1-3 keV) considered in this project, the electron range varies from about 30nm (at 1keV) to almost 200nm (at 3keV). For surfaces with sufficiently sharp corners, small projections, or pits the SEM signal is more accurately predicted by a Monte Carlo simulation.

My SEM model has much of the computational convenience of the slope-based models, with a closed form solution that is relatively quick to compute, but is also capable of mimicking the Monte Carlo simulation more closely than slope-based models. Here I review existing SEM models and methods for surface reconstruction using some of those models.

2.2.1 Accelerated or Precomputed Monte Carlo Simulation

An example that typifies all previous methods that use the Monte Carlo model to compute shape from SEM images is the "inverse scattering" approach [Davidson99][Villarrubia04]. Because of the long time required to calculate an SEM image using the Monte Carlo method, the approach in [Davidson99] makes use of precomputed images and assumes a very simple model for the surface shown in Figure 2-3.

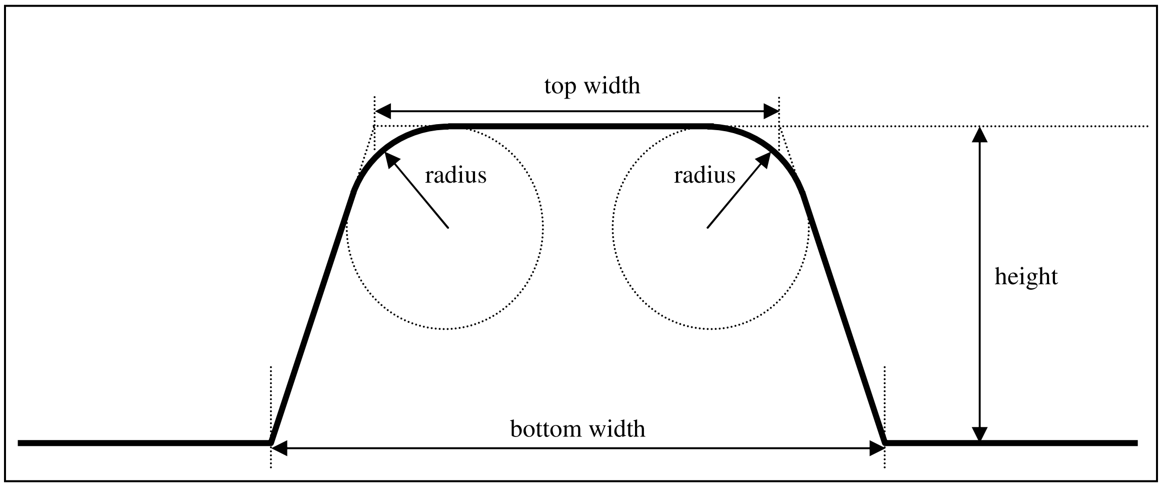


Figure 2-3: A simple model of surface shape (from [Davidson99])

The problem is to determine the cross section shape (a curve in 2D) of a surface whose shape is an extrusion of that cross section described by 4 parameters: top width, bottom width, height, and radius of curvature (see Figure 2-3). The 4-dimensional parameter space is sampled and corresponding SEM scan profiles are generated offline by Monte-Carlo electron scattering simulation to create a library of scan profiles. The offline calculation requires several days on a 330MHz Pentium machine. Actual data (line scans extracted from real SEM images) is compared to all of the approximately 1000 simulated line scans stored in the library to find the best match. To compensate for an unknown and possibly spatially varying amplification of the signal from the microscope, a slowly spatially varying gain parameter is optimized to minimize the least squares error between the simulated and acquired SEM intensity before testing each possible match [Davidson99]. This sort of brute force approach is only applicable for highly constrained specimen models with a small number of parameters but it does demonstrate the value of the Monte Carlo model for real applications at scales where slope-based models do not provide the required accuracy.

For the less constrained problems that I am targeting it is not practical to do the simulation in a preprocess step. Instead, images must be simulated on the fly as they are needed by a surface optimization algorithm. At each step of an optimization algorithm, an image would be simulated, compared with the real SEM image, and an update to the surface generated in accordance with an optimization search strategy, minimizing the difference between the simulated and real SEM images. For such an optimization approach, it is reasonable to assume that a simulated SEM image would need to be computed at least once for each surface parameter. For a surface represented by a 64x64 grid of basis functions there are 4096 parameters. Computing a 64x64 pixel SEM image on a single processor with 1000 electrons per pixel (the same number of electrons used in [Davidson99]) requires about 820 seconds. This figure is based on the Monte Carlo SEM simulator being limited to rate of about 5,000 electrons per second per processor when secondary electrons are included [Seeger03]. Computing 4096 such images would require 38 days on a single processor. This is far from practical and the only way it might work would be to drastically reduce the number of electrons simulated and take advantage of independence of the effects of different parts of the surface on the objective function to optimize the surface more efficiently. However, as the number of electrons is reduced the noise in the objective function evaluations would increase and tend to reduce the rate of convergence so it is not obvious that this approach would improve the speed of the optimization overall. If each basis function was assumed to be independent from those outside of a 7x7 neighborhood then the gradient might be estimated by simulating just 50 images but this would still require almost a whole day on a single processor. At least in the near future, any work using this approach would probably be limited to matching a 1D line scan as in [Davidson99] rather than a 2D image.

Even if one could overcome the speed issues, Monte Carlo simulation doesn't provide a quantitative prediction of an SEM image without some additional tweaks. While the Monte Carlo model is useful for understanding what happens within the specimen, there are many interactions occurring after electrons leave the specimen that are not typically simulated. These interactions include the production of secondary electrons from parts of the specimen chamber and detector sensitivity characteristics. Even if one could assume fixed geometry of the specimen chamber and detector, instrument parameters such as amplifier gain and offset are not typically recorded and make it difficult to quantitatively analyze SEM images. Unknown factors affecting image contrast (such as contamination with vacuum pump oil) may vary from one SEM to the next and from day to day for the same SEM. If specimen charging occurs, taking this into account can increase the simulation time by several orders of magnitude. In matching a Monte Carlo simulation to real data, various calibration factors must either be estimated through trial and error, labor-intensive measurement of detector characteristics [Drouin99], optimized along with the unknown specimen parameters within a numerical optimization, or some combination of these approaches [Davidson99]. It would be better to have a model that could be easily calibrated to any SEM and operating conditions, perhaps by using a calibration object of known shape.

2.2.2 Diffusion Matrix

A diffusion matrix [Desai90] [Czyzewski91] represents electron scattering in an infinite volume by sampling the electron flux through hypothetical surface patches with different orientations at regularly-spaced points in a 2D or 3D voxel grid. The matrix is precomputed from a Monte Carlo simulation in an infinite volume of material. To compute the SEM image from a height image, one scans the precomputed matrix over a piecewise linear representation of the surface and at each point, sums the contributions from the flux through each facet of the surface. This algorithm has only been implemented for surfaces defined by 2D cross sections to generate 1D SEM scans. While in theory the diffusion matrix could work in 3D it may not be fast enough. Another drawback of this approach is that as with Monte Carlo simulation it is limited to modeling the emission of electrons from the specimen. Like Monte Carlo simulation, the diffusion matrix would require additional fitting parameters to compare its output with real SEM images.

2.2.3 Stopping Point Distribution Model

This approach has been described in [Aristov91] and [Firsova91] and is based on the distribution of points at which electrons stop within a hypothetical infinite volume of material relative to the point at which incident electrons enter the specimen. In [Aristov91], the relation between the BSE signal and the topography is approximated by the integral of stopping-point density that lies above the surface:

$$\frac{\eta(x_0, y_0)}{1+\eta(x_0, y_0)} = \iiint\limits_{\substack{-\infty<x,y<+\infty \\ z>f(x,y)}} F\left[x-x_0, y-y_0, z-f(x_0, y_0)\right] dxdydz$$

where $\eta(x_0, y_0)$ is the ratio of BSE to incident electrons when the electron beam hits the surface at (x_0, y_0), $f(x,y)$ is the height of the surface, and $F(x,y,z)$ is the probability that (x,y,z) is the stopping point of an electron trajectory. This stopping point distribution is approximated by a Gaussian function with center at some depth directly below the beam entry point. In [Firsova91], the width and center parameters of the Gaussian distribution function were estimated both from experimental data and from a Monte Carlo simulation. An iterative algorithm is used to estimate $f(x,y)$ from an experimentally measured BSE signal $\eta_{\exp}(x, y)$. Given an experimentally measured BSE signal and the surface estimate $f^{(n)}(x,y)$ at iteration n of the algorithm the surface is updated at every point (x_0, y_0) using the recursion

$$f^{(n+1)}(x_0, y_0) = f^{(n)}(x_0, y_0) + s\left(\frac{\eta_{\exp}(x_0, y_0)}{1+\eta_{\exp}(x_0, y_0)} - \iiint\limits_{\substack{-\infty<x,y<+\infty \\ z>f^{(n)}(x,y)}} F\left[x-x_0, y-y_0, z-f^{(n)}(x_0, y_0)\right] dxdydz\right)$$

where s is a constant factor that determines the rate of convergence.

One of the weaknesses of this model is that it cannot accurately account for shadowing effects caused by electrons escaping and re-entering the surface. Also, this model has only been demonstrated for the BSE signal which is not typically used for nanometer-scale metrology because of its low resolution and low signal to noise ratio. Another limitation of this approach is that it assumes that the variation in height is smaller than the distance that backscattered electrons can travel within the specimen (electron range). The electron range can be less than 100nm and in many cases surface height variation is larger than this distance.

2.2.4 Function of Incident Angle and Shape-from-Shading

2.2.4.1 Slope-based Models for Intensity

In [Ikeuchi81], real secondary electron SEM data was compared with several approximations to the image brightness including $e^{\alpha(1-\cos(\theta))}$, $\sec(k\theta)$, and $(1-s)+s\cdot\sec(\theta)$ where θ is the incident angle (equal to 0 when the electron beam is perpendicular to the surface). The best fits to experimental data were found with $\alpha \approx 1$, $k \approx 0.8$, and $s \approx 0.5$. To avoid the non-physical unbounded increase of the third model at angles of incidence near 90°, an ad hoc modification was applied above 70° to limit the increase.

The models described in [Ikeuchi81] are plotted in Figure 2-4 along with the output of a Monte Carlo simulation and the simpler formula, sec(θ), used in [Jones94]. It may seem obvious from this graph that a better fit between the Monte Carlo simulation and those in [Ikeuchi81] could be achieved by simply scaling the various approximations. This is the first comparison between such slope-based models and Monte Carlo simulation that I am aware of and so the thought of such an adjustment was not likely to have occurred to anyone in the past. In [Davidson99] and in this work, a scaling factor is applied to the output of Monte Carlo simulation to match experimental data (see section 6.7.1 for more details). Though the Monte Carlo simulation is certainly a more accurate model of the underlying process it may not be the most accurate phenomenological model of the measured SE yield. Also, there are such large variations in experimental data that it is difficult to say which model best matches the data [Joy94].

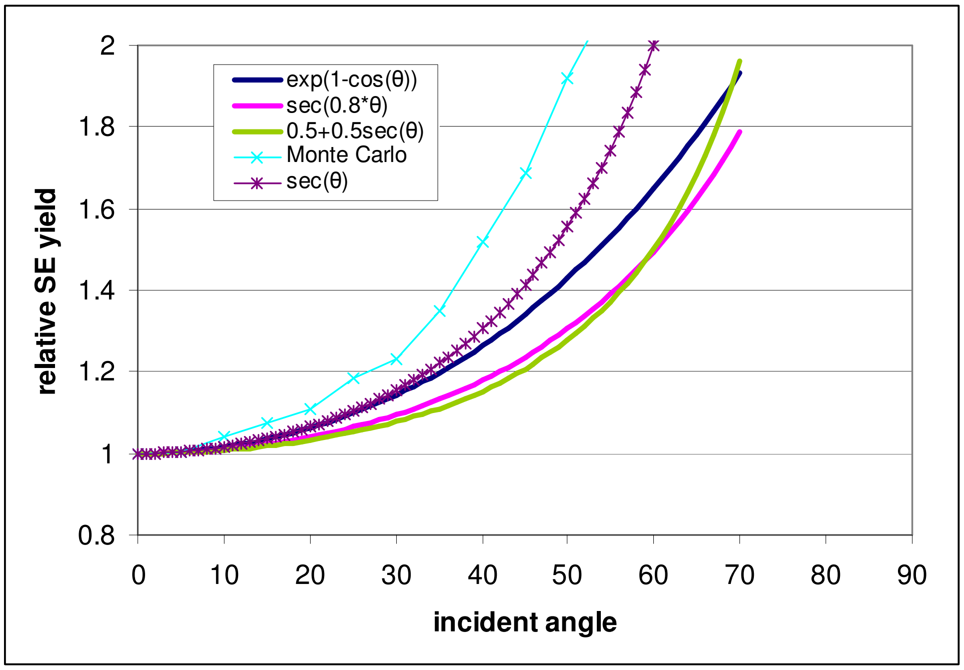


Figure 2-4: Comparison of previously used reflectance functions with the output of a Monte Carlo simulation. The y-axis is brightness relative to the brightness for a surface perpendicular to the electron beam

Approximations to the SE type I signal based on the secant function come from the assumptions that secondary electrons can only escape a specimen if they are generated within a thin layer near the surface, that secondary electrons are generated at a constant rate over the part of the incident electron's trajectory within this thin layer, and that the chance of escape for all secondary electrons generated within the layer is the same. With these assumptions, the number of SE type I is proportional to the length of an incident electron trajectory that is contained in the near surface layer. This distance is proportional to a secant function of the angle between the incident beam and the surface normal as shown in Figure 2-5.

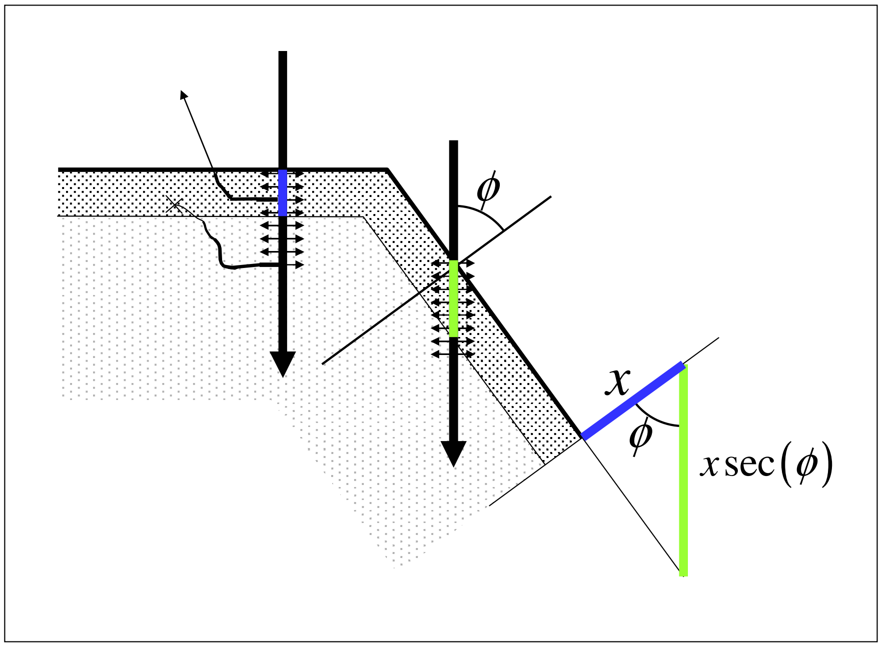


Figure 2-5: Secondary electron yield due to primary beam electrons as they first enter the specimen (SE type I) depends strongly on angle of incidence for the beam because this determines the distance over which the incoming electrons remain near the surface.

Although a shading model (where intensity is assumed to depend only on slope) may not accurately model SEM intensity, some of the approaches that assume such a model are still relevant because they may be modified to use a more accurate model.

2.2.4.2 Surface Reconstruction Using Slope-based Models

The goal of *shape from shading* is to recover the height field describing a surface from a 2D shaded image of the surface. The intensity *I* at a point *(x,y)* in the shaded image is assumed to be given by the image irradiance equation

$$I(x,y) = R(\vec{\mathbf{n}}(x,y))$$

Equation 2-1

where *R*, the reflectance function, is a function only of the surface normal $\vec{\mathbf{n}}$ at the point on the surface that projects to *(x,y)* in the shading image. Lighting and viewing parameters are considered part of the reflectance function. A common choice for the reflectance function is the Lambertian shading model defined as

$$R(\vec{\mathbf{n}}(x,y)) = \eta\vec{\mathbf{n}}(x,y)\cdot\vec{\mathbf{l}}$$

where η is a scalar representing the product of the light intensity and intrinsic reflectance (albedo) of the surface and $\vec{\mathbf{l}} = (l_x, l_y, l_z)$, the light source vector. Locally, the intensity in the shaded image does not fully constrain the surface normal because the normal is only the dot product of the normal with the light vector is fixed. The possible normals projected onto a plane perpendicular to the light vector form a circle representing a constant angle with the light vector. Additional constraints such as surface smoothness, boundary conditions and integrability must be introduced to find a unique solution. Computing shape from SEM images has been treated as an instance of the shape from shading problem by several researchers [Horn70, Jones94]. For shape from shading the SEM intensity is commonly assumed to be the SE type I signal with the form $\delta = \delta_0 \sec(\phi)$ where δ_0 is the signal from a flat surface perpendicular to the incident electron beam and ϕ is the angle between the surface normal and the incident electron beam. In the reflectance function notation the function is defined as

$$R(\vec{\mathbf{n}}) = \frac{\delta_0}{\vec{\mathbf{n}} \cdot \vec{\mathbf{l}}}$$

As in the Lambertian model, the surface normal only appears in a dot product with the light vector appears in this reflectance function so the normal is similarly unconstrained. Consequently, the model produces an estimate of the magnitude of the height gradient but not the direction of the height gradient. For cases where image intensity depends only on height gradient magnitude one can use Horn's method of characteristic strips to integrate the height along special curves that follow the intensity gradient in the shading image [Horn70]. With this method, small errors due to noise are accumulated by the integration. Other approaches to shape from shading attempt to avoid problems with noise by imposing smoothness constraints or by using a multi-resolution method. Dupuis and Oliensis developed a method based on Horn's characteristic strips and proved that knowledge of local minima and maxima for the surface height function enabled their method to converge to an optimal solution [Dupuis94]. It is not obvious how to apply such a method to shape from SEM shading when the intensity cannot be assumed to be a function only of slope.

The work of Leclerc and Bobick [Leclerc91] is representative of methods that incorporate an explicit smoothness constraint and multi-resolution representation for the input image. In [Leclerc91], the surface is represented as an array of height values $\mathbf{z} = \{z_{ij}\}$ and the objective function for the optimal reconstruction is defined as

$$E = \sum_{i,j} \left[(1-\lambda)\left(R\left(p_{ij}, q_{ij}\right) - I_{ij}\right)^2 + \lambda\left(u_{ij}^2 + v_{ij}^2\right) \right]$$

Equation 2-2: Example of an objective function for shape from shading incorporating a smoothing term that serves the purpose of sufficiently constraining (or *regularizing*) the solution (from [Leclerc91])

where approximations to first and second derivatives of the height are computed as

$$p_{ij} = \frac{1}{2}\left(z_{i+1,j} - z_{i-1,j}\right) \text{ (slope in x)}$$
$$q_{ij} = \frac{1}{2}\left(z_{i,j+1} - z_{i,j-1}\right) \text{ (slope in y)}$$
$$u_{ij} = z_{i+1,j} - 2z_{i,j} + z_{i-1,j} \text{ (curvature in x)}$$
$$v_{ij} = z_{i,j+1} - 2z_{i,j} + z_{i,j-1} \text{ (curvature in y)}$$

and *R*(*p,q*) is the reflectance function in terms of the slope of the surface rather than the surface normal as in Equation 2-1. The first term inside the summation is the intensity error and the second term is a smoothness constraint. The scalar, λ, controls the relative weighting of the smoothing term and the intensity error term.

The objective function is optimized using a multiresolution approach in which an image pyramid is formed by blurring the input shaded image and subsampling by a factor of 2 from an initial resolution of 64x64 to 32x32, 16x16, and 8x8. Starting with the 8x8 image, the corresponding subsampled array of height values is iteratively optimized to minimize *E* and then bilinearly interpolated to the next higher resolution and the optimization started again for the new resolution. This process is repeated up to the highest resolution. At the lowest resolution, the factor λ is initially set to 1.0 and is decreased by a factor of $\frac{1}{\sqrt{2}}$ in a geometric sequence down to 1/16. Thus there are actually two nested sequences that incrementally increase the level of detail: one for the resolution of the image and one for weighting factor controlling the smoothness (λ). For each resolution and value of λ, the partial derivatives of *E* with respect to each of the components of the light vector $\vec{\mathbf{l}} = (l_x, l_y, l_z)$ and the height array $\mathbf{z} = \{z_{ij}\}$ are computed and the conjugate gradient algorithm is used to minimize the objective function. When a minimum is reached for a given value of λ, the value is decreased to the next smaller value and the conjugate gradient algorithm is restarted. Each time the resolution is increased, the minimum and maximum values for λ are both divided by 2. Thus, at the lowest resolution, the weight for the smoothing term (λ) is stepped in a geometric sequence from 1.0 to 1/16. At the next higher resolution, the weight for the smoothing term (λ) is stepped in a geometric sequence from 1/2 to 1/32 and so on.

Peleg and Ron [Peleg90] observed that such a reconstruction based on a blurred input image can produce a qualitatively incorrect solution if the relationship between the input data and the solution is not linear and suggested a way to partially reduce this problem using a non-linear blurring operation. A complementary method developed by Jones and Taylor [Jones94] called scale-space reconstruction uses a multi-resolution representation of the surface but does not require any blurring or subsampling of the input image.

In [Jones94] the surface height is represented as the sum of Gaussian basis functions arranged on a square grid. Because the surface model is composed of smooth functions, an explicit smoothness constraint is not needed in the objective function. As in [Leclerc91], the conjugate gradient algorithm is used to optimize the coefficients of the Gaussian basis functions but where the gradient of the objective function becomes 0 a genetic algorithm is used to search a small neighborhood around the current solution to help escape local maxima or saddle points. The standard deviation of the Gaussian basis functions is gradually reduced from 60 pixels down to 0.39 pixels for a 64x64 shaded image and at each scale the surface is optimized starting from the optimal surface at the previous larger scale. The resolution of the grid of basis functions (number of functions along one edge of the surface) is reduced for larger Gaussian widths according to

$$M(\sigma) = \begin{cases} 6M_{\max}/\sigma & \sigma > 6 \\ M_{\max} & \sigma <= 6 \end{cases}$$

where $M_{\max}$ is the largest grid width equal to the number of pixels in the shaded image ($M_{\max} = 64$ for 64x64 shaded image) and σ is the standard deviation of the Gaussian basis functions in units of shaded image pixels. This subsampling of the basis function grid significantly reduces the number of parameters in the surface model at large scale. The original shaded image is also blurred using Peleg and Ron's non-linear blurring method and subsampled to the same resolution as the grid of basis functions.

In both [Jones94] and [Leclerc91], the numbers of parameters are reduced at low resolution but at the highest resolution, the number of parameters is equal to the number of pixels in the input image and therefore quite large for an optimization problem even for a small image with 64x64 pixels. A variation on Jones and Taylor's approach is described by Wei and Hirzinger [Wei97]. They also use Gaussian basis functions but only place functions where the intensity error is above a threshold. Initially, all basis functions on a grid similar to that used by Jones and Taylor are considered. The average intensity error for a receptive field corresponding to each basis function is computed and tested against a threshold. If the average error is above the threshold, the corresponding basis function is introduced into the surface model. After this initial selection, the surface model is optimized with the centers and widths of the basis functions as additional free parameters. Also, instead of the conjugate gradient method, the stochastic gradient method is used. The stochastic gradient method consists of iterating over all pixels in the image and for each pixel, updating the model by considering the error gradient for the single pixel. For the *i*th pixel each model parameter *w* is updated according to

$$\Delta w^{(n)} = -\beta \frac{\partial E_i}{\partial w} + \alpha \Delta w^{(n-1)}$$

where E_i is the error for the pixel and $\Delta w^{(n)}$ is the change in *w* for the *n*th pass through all the pixels. β is a learning rate and $\alpha \Delta w^{(n-1)}$ is a momentum term. Wei and Hirzinger state that values of β=0.3 and α=0.6 work reasonably well in their application. The authors claim improvements in performance using the stochastic gradient method over conjugate gradient. The number of basis functions used is typically half that used by Jones and Taylor's approach but the total number of free parameters may actually be greater because each basis function has 4 adjustable parameters (height, width, x position, y position) instead of just a height parameter. I do not use this method because it is not obvious how to efficiently maximize a match to both an SEM image and an AFM image. It may be difficult to incorporate other constraints (besides those from the SEM image) in the stochastic gradient method.

I adopt the scale-space reconstruction framework described in [Jones94] but do not use any blurring of the input AFM and SEM images. I use the same specimen representation, coarse-to-fine optimization strategy and conjugate gradient method for optimization at each scale. Unlike [Jones94] I do not use a genetic algorithm to help prevent the conjugate gradient method from getting stuck. I extend this method by incorporating constraints from the AFM image data along with those from the SEM image data by optimizing combined likelihood, using data in each image to resolve ambiguities and reduce errors due to noise that would occur using either image alone. Also, the model for SEM intensity that I use, described in chapter 6, is significantly more complex than those used by existing shape from shading methods. While this model for SEM intensity can be computed nearly as quickly as those used by shape from shading algorithms, it captures behavior of the Monte Carlo model that is missing from existing shape from shading methods.

3 Other Related Inverse Problems

Chapter 2 reviewed previous work directly related to surface reconstruction from AFM and SEM. In this chapter I describe some additional inverse problems that are not so directly related but that are similar in some important aspects to my problem. For example, the non-linear nature of the SEM gives it an effectively variable point spread function over an image because the interaction volume has a different shape depending on the shape of the topography. Although it has a more subtle effect, the finite depth of focus in SEM also contributes to this variability but I do not model this. In reconstruction of a scene from synthetic aperture rader (SAR) the point spread function also varies across an image although unlike the SEM, the point spread function depends on image location in a known way.

In [Robini97], the simulated annealing algorithm (to be described shortly) was used to reconstruct a 2D grayscale image acquired by synthetic aperture radar (SAR). SAR works by scanning a wide-bandwidth radio emitter along a straight path. At sample points along the path, echoes from objects that scatter the radio signal back are acquired. The 2D image encodes this backscattered signal such that one dimension of the image represents the time that it took the echo to come back and the other dimension represents distance along the emission path. The temporal dimension is also proportional to the distance of the scattering object from the radar device. Scatterers that are farther away are effectively measured with a larger aperture because of the conical volume from which the receiving antenna collects the backscattered signal. As a result, the formation of the SAR image may be accurately modeled as a convolution of a 2D scene with a spatially varying blurring operation in which the blur kernel depends on distance from the acquisition device. The 2D scene describes the density of scatterers as a function of distance from the acquisition path and distance along that path. Robini et al construct an objective function that consists of a smoothing term (as in Equation 2-2) and an error term consisting of the sum of squared differences between measured data and the data predicted by the scene model and a theoretical point spread function. In this case, the model is also a 2D image representing the sampled density of scatterers.

The simulated annealing algorithm iterates over the parameters (pixels) of the model and for each pixel considers a new randomly generated grayscale value. If the objective function value for the resulting candidate image is less than the old objective function value, the new candidate is accepted; otherwise it is accepted with some probability that depends on a scalar temperature parameter. The role of the temperature parameter is to allow changes that reduce fitness with high probability initially and over the course of the optimization the temperature is gradually reduced making acceptance of such changes less likely. This approach was inspired by the simulation of a liquid as it cools to form a crystal (a type of annealing process). As it cools, the material finds the lowest energy state. In [Robini97] the new grayscale values are sampled from a subset of the range of intensities in the image based on the intensities in the local neighborhood of that pixel following an approach suggested in [Yang93]. This restriction increases efficiency by reducing the frequency of rejected candidates. Another method used to speed up the annealing process involved a distortion of the objective function that automatically made it more and more difficult for the algorithm to escape from minima as the objective function value decreased [Robini97]. Even with such acceleration techniques, the number of objective evaluations and computing time required for convergence with simulated annealing can be very large compared with algorithms that are less general and include restrictions that are specially fit to the particular optimization problem. Yang remarks that the computational cost of simulating annealing is especially high when the parameters are strongly correlated (e.g. when the point spread function in the image reconstruction problem is large)[Yang93].

In my specimen representation, the parameters are very strongly correlated in terms of their effect on simulated AFM and SEM images, particularly at the nanometer scale. This correlation is the result of the wide effective point spread function of both the AFM and the SEM relative to the size of the topographic features I wish to reconstruct.

Computing exposure patterns for electron beam lithography has some interesting parallels to shape reconstruction from SEM images because both cases involve 2D images resulting from the same electron scattering process. In the electron beam lithography problem, the "measurement" is actually a target surface that one wishes to create through the electron beam lithography process. The "model" to be recovered consists of the parameters of the manufacturing process: an exposure pattern that describes the dwell time for the SEM electron beam at each point in a set of (x,y) scan positions on a regular grid. An important difference between this problem and the surface reconstruction problem is that for electron beam lithography one wishes to find a 2D image (the exposure pattern describing density of incident electrons) given a surface, while in surface reconstruction one wishes to find a surface given a 2D image (describing signal from electrons that are scattered out of the surface). The direct problem, mapping from exposure pattern to predicted surface, may be solved using a Monte Carlo simulation of electron scattering for an initial surface (as in surface reconstruction from an SEM image), keeping track of energy deposited by the electrons in the volume of material, and by simulating the etching process that removes exposed parts of the surface to form the final surface. The target surface may not be physically realizable but the goal is to create a surface that is as close to the target as possible. Deviation between the target surface and the nearest physically realizable surface is analogous to measurement noise in the other reconstruction problems described here. The process of electron irradiation and etching is analogous to the process of image formation.

Robin et al used a genetic algorithm and a simplex-downhill method to solve a problem in electron beam lithography [Robin03]. In electron beam lithography, the electron beam is scanned over a flat surface of resist material. The surface is then etched such that the places that were exposed to the beam are removed. Over a fixed amount of time, the speed and hence depth to which the resist is etched depends on the amount of exposure to the electron beam. By varying the exposure across the surface, 3D structures can be created in the resist. Because the electron scattering and subsequent etching processes are non-linear, in general it is difficult to determine the appropriate exposure pattern given a desired 3D shape.

Robin et al determine an exposure for a shape represented by a 2D cross-section. Given a string of (x,z) pairs representing the cross-sectional shape of a surface (shape was invariant for translations in y), the problem was to determine what exposure pattern to use in creating that shape. They used a pattern composed of 100 dwell points spaced along a line at 16 nm intervals. Since the electron beam penetrated significantly deeper in the sample than the depth of the target pattern, the density of scattering electrons was assumed not to depend on z. This assumption combined with the fact that the initial surface (at the time of exposure) is flat simplified the direct problem so a 1D convolution could be used in place of a Monte Carlo simulation (electron scattering was modeled as a linear process). The convolution kernel was a sum of a narrow Gaussian function (40 nm full-width-half-maximum (FWHM)) representing the spatial distribution for exposure due to incident electrons before their first elastic scattering and a much wider Gaussian function (1780 nm FWHM) representing the spatial distribution for exposure from backscattered electrons. The etching process was simulated using the "string" model that iteratively steps the points in the profile perpendicular to the surface, using empirically-fit models of the rate of propagation as a function of developing time. This profile calculation was found to dominate the computation time for optimization. The objective function was computed as the Euclidean distance between the points in the target profile and the points in the computed profile.

The simplex-downhill method (or Nelder-Mead method) successively transforms a simplex (a set of (n+1) points in the n-dimensional search space) to seek the optimal point. In [Robin03] the search space has 100 dimensions. Each coordinate direction corresponds to the dwell time for one of the 100 dwell points. To initialize the algorithm, Robin et al initialized a single vertex (representing a particular exposure pattern consisting of dwell times for each of the 100 points) of the simplex randomly and set the other n points to that point plus constant offsets in each of 100 coordinate directions (perturbations of a single dwell time in the exposure pattern). Depending on the value of the objective function at each of the vertices in the simplex, one of four transformations is applied to the simplex so that it moves towards a local minimum. All transformations are essentially a scaling about either a point or a face (n points in a face) of the simplex and each transformation usually requires just a single vertex to be updated. In their experiments, Robin et al found that the simplex-downhill method converged after only a few hundred transformations and was apparently easily trapped by local minima far from the global minimum. An objective for surface reconstruction from SEM images is likely to have a similar abundance in local minima due to the strong correlation between neighboring pixels in the exposure pattern or SEM image (from overlapping interaction volumes in the specimen). Robin et al concluded that the problem was too difficult for the simplex downhill method and that a global optimization method such as a genetic algorithm was required.

For the genetic algorithm, a population of 100 individuals was used (same as the number of model parameters). The first generation was initialized by randomly picking all dwell times from a uniform distribution and then forcing random small groups of consecutive times within an individual to have the same value. In each iteration, two parent individuals were selected from the previous generation, recombined to create two new offspring by splicing at 2 randomly selected points the arrays of dwell times for the two parents. A mutation operation was then applied to the offspring that randomly perturbed one of the dwell times. Then the objective function was computed for the offspring and they were reinserted into the population. Three methods of selecting the two parents were considered. Roulette-wheel selection sets the probability of an individual being selected proportional to its fitness. Rank-based selection sets the probability of selection proportional to an individual's rank in the population. Tournament selection selects each individual by randomly selecting a subset of the population and choosing the fittest individual from that subset to be added to the next generation. In each case, individuals were selected without replacement to avoid duplication. No significant differences were found in the performance using these different selection methods. Significant benefit was found in using a rejuvenation operator once every 5000 cycles which replaced the 75% worst individuals by new randomly generated individuals. The genetic algorithm was found to provide a useful solution (transistors were successfully manufactured using the output dwell times).

Both [Robini97] and [Robin03] found it necessary to use global optimization methods: simulated annealing and a genetic algorithm. Such methods may be necessary when there are multiple locally optimal solutions as is typical for such non-linear inverse problems based on complicated physical simulations. The problem of determining shape from combination AFM and SEM images is also likely to have multiple locally optimal solutions. However, given the large number of objective function evaluations required by global optimization algorithms, I did not consider such strategies to be practical for this problem because of the long time required to simulate AFM and SEM images and compute the objective function. Instead, I use a scale-space reconstruction framework [Jones94] that constrains the solution to one that can be tracked by local optimization from large scale to small scale. Though this solution may not be globally optimal, I assume that it is significantly closer to the globally optimal solution than solutions provided by existing methods. Local optimization is much more efficient and greatly reduces the number of objective functions required. Scale-space reconstruction also provides the added benefit of an implicit smoothness constraint that helps to find a relatively simple surface that explains the observed images when there are many more complicated surfaces that would explain the images just as well.

4 AFM Image Analysis and Modeling

This chapter describes preprocessing of the AFM image in my technique that is independent of the SEM image. The first step is to flatten the AFM image, transforming it into a standard coordinate system for registration to the SEM image. Next, the tip shape is estimated using Villarrubia's blind tip reconstruction algorithm. An initial estimate for the surface is found using the erosion operation and the reconstructed tip shape.

Many of the new contributions related to AFM analysis are covered elsewhere: A model for correlated noise in AFM images is described in section 8.1. An objective function for the surface reconstruction based on a model for the AFM image as a dilation plus additive correlated noise is described in section 9.2.1. The derivative of this objective function is described in section 9.3.1. Much of the background on mathematical morphology including a discussion of the differences between discrete morphological operations (on which my method is based) and continuous morphological operations is provided in Appendix A. Acceleration of mathematical morphology operations using graphics hardware is described in appendix section Appendix E.

4.1 Flattening

To bring the AFM data into a common coordinate system (up to a 2D transformation) with the SEM image plane that is assumed to be perpendicular to the specimen substrate, I apply a flattening procedure that transforms the data by subtracting a plane fit to the AFM image. This is equivalent to a applying a shear operation to the data set. Because of the small tilt angles involved, this procedure is for practical purposes equivalent to a 3D rotation. For the example shown in Figure 4-3 the height of the substrate varied by about

80nm over a distance of 7000nm, equivalent to a tilt of about 0.65°. Subtracting a plane fit to the substrate compared with rotating the surface to align the substrate with the x-y plane results in a scaling by cos(0.65°)=0.999935 for points at the same height relative to the substrate. There is also a translation of points that are on raised features relative to the substrate. This translation is equal to the height of the raised features times sin(0.65°). The tallest features are about 100nm which corresponds to a translation of (100nm)*sin(0.65°) = 1.13nm of the peaks relative to their base. Such distortion is much smaller than the expected error given that the pixels in the AFM image are 23.33nm wide. The scaling is also quite insignificant, resulting in an image that is about 0.5nm smaller than the rotated surface (7000nm instead of 7000.5nm). These errors are illustrated in Figure 4-1.

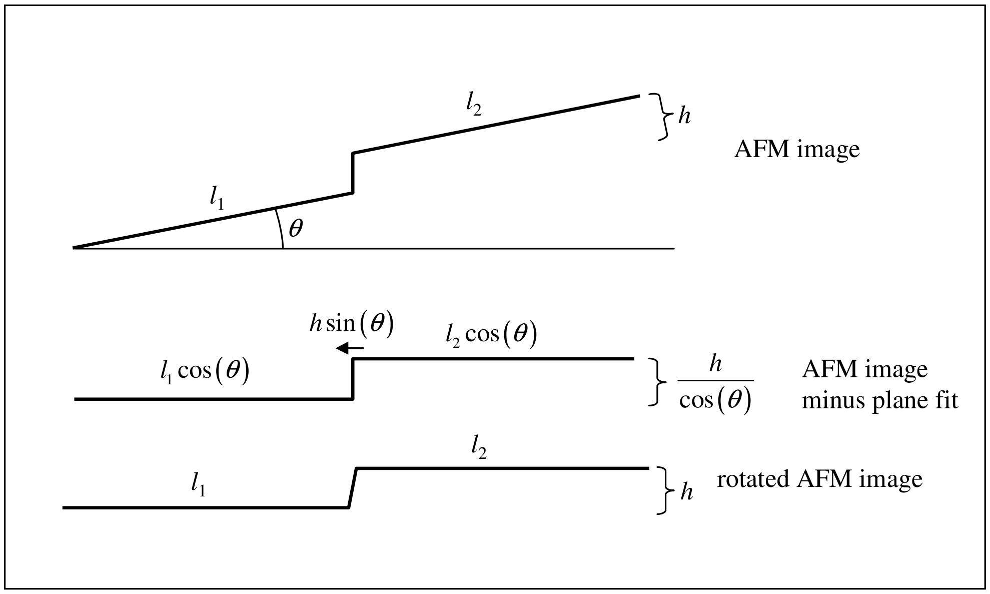


Figure 4-1: Errors introduced by using a shearing operation for flattening rather than a 3D rotation. All distances measured parallel to the plane fit are scaled by cos(*θ*). All heights measured perpendicular to the plane fit are scaled by 1/cos(*θ*). Points at a height *h* above a substrate are translated by *hsin(*θ*) relative to the substrate.**

I developed a new approach to AFM image flattening that is appropriate for specimens where most of the surface is a flat substrate or where a significant fraction of the surface is well approximated by the plane to be subtracted as part of the flattening operation. The plane fit is found by first using a 3D Hough transform to identify the plane that matches within some tolerance the largest number of pixels on the surface.

The Hough transform, introduced in 1962 by Paul Hough transforms shapes such as lines or circles in image space into points in an accumulator space [Hough62]. The problem of identifying salient examples of such shapes in an image is converted to one of finding peaks in the accumulator space. One common application is to find straight lines in a 2D image [Gerig87][Illingworth88]. For this application, a commonly used accumulator space is the space of (r,θ) where the equation for a line is $r = x\cos(\theta) + y\sin(\theta)$ as shown in Figure 4-2.

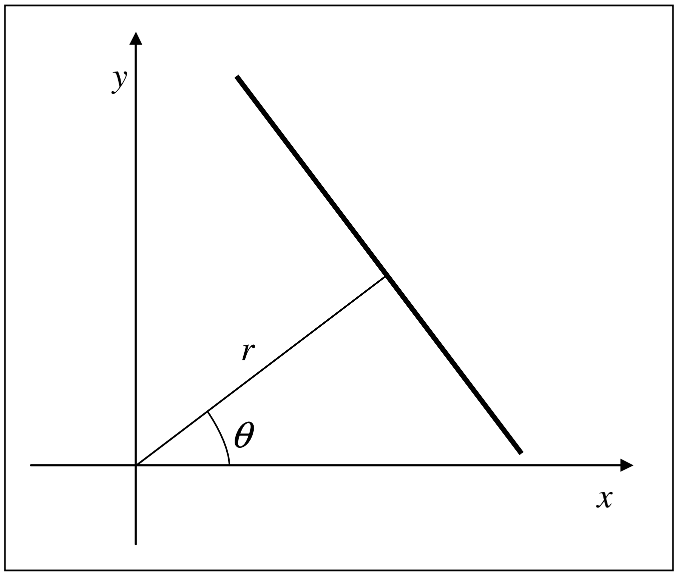


Figure 4-2: Parameterization of a line typically used for the straight line Hough transform

An input image is converted through edge-detection or other means into a binary image where each pixel has value 1 where an edge is detected and 0 otherwise. The algorithm for identifying significant lines maps each pixel with value 1 to a curve in the (r,θ) accumulator space. This curve represents all possible lines passing through that point. The accumulator space is discretized to a 2D grid and an accumulator consisting of one counter for each grid point is initialized to 0. Then, for each pixel with value 1 and each value along the θ axis, the radius is computed using $r = x\cos(\theta) + y\sin(\theta)$ where (x,y) is the position of the pixel. The radius value is rounded to the nearest discrete radius value in the grid and the counter corresponding to the (r,θ) point in the curve is incremented. After all pixels have been counted, the (r,θ) point with the greatest value in the accumulator gives an approximation to the most salient line in the original image.

As the straight line Hough transform identifies a line from a set of points in 2D, a 3D Hough transform can be used to find a plane given a set of points in 3D. A plane is represented by the equation $r = x\cos(\theta)\cos(\phi) + y\sin(\theta)\cos(\phi) + z\sin(\phi)$ and the accumulator space is over (r,θ,ϕ). For each data point at position *(x,y,z)* in the AFM image, the algorithm considers each discrete orientation from the grid of (θ,ϕ) values and computes the radius for a plane using $r = x\cos(\theta)\cos(\phi) + y\sin(\theta)\cos(\phi) + z\sin(\phi)$. For each (r,θ,ϕ) grid point computed from each *(x,y,z)* the corresponding accumulator counter is incremented. When all points from the image have been processed, the counter with the most counted pixels is selected. A least squares plane fit is then computed for the set of nearly coplanar pixels that corresponded to that counter. In practice I found that 50 equally spaced values for the radius (from the minimum height to the maximum height of the surface after subtracting the mean height from the image), 160 equally spaced values for θ from 0° to 360°, and 80 equally spaced values for ϕ from -90° to 90° produced reasonable results. An example of the output of the flattening procedure is shown in Figure 4-3

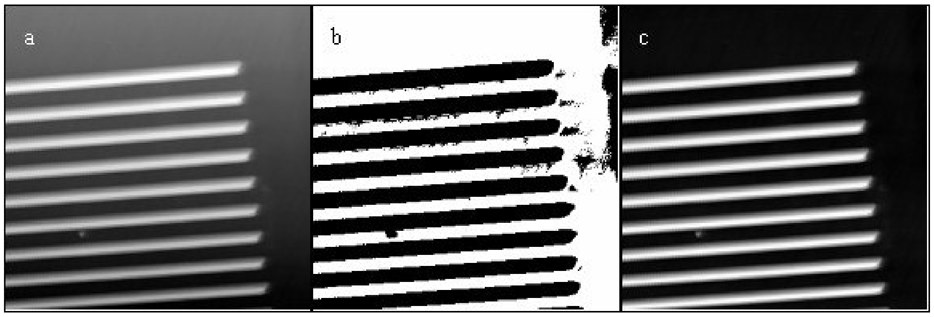


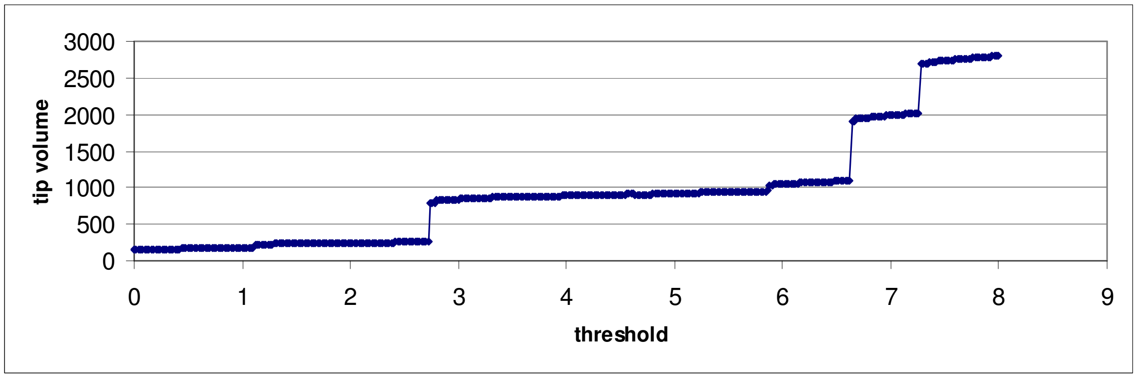
Figure 4-3: Flattening an AFM image using the 3D Hough transform. The largest set of nearly coplanar pixels in an input image (a) is first segmented using the Hough transform method (white pixels in image (b)). A plane fit to the selected pixels is subtracted from (a) to produce a flattened image (c).

4.2 Blind Tip Reconstruction

For estimating tip shape I use Villarrubia's noise-tolerant blind tip reconstruction algorithm introduced in section 2.1.2.2 and described in detail in Appendix A [Villarrubia97]. As previously described, this algorithm requires as one of its inputs a threshold parameter to control the degree of tolerance to noise. If the threshold is too small, noise will influence the reconstruction and result in a tip estimate that is much too sharp. If the threshold is too large, the algorithm will ignore too much structure in the image and the tip estimate will be too dull.

Previous studies have shown a sudden increase in the sharpness of the tip estimate to occur just below the threshold value that yields the most accurate tip reconstruction and as the threshold is reduced further there is a sudden increase in the sharpness of the tip estimate [Dongmo00]. The volume of the tip can be used to detect the threshold value at which noise starts to have a significant effect on the tip shape. I use the method described in [Dongmo00] and [Todd01] of plotting the tip volume as a function of the noise threshold parameter in order to select the best threshold value. In this method, a somewhat arbitrary regular grid of threshold values is selected and for each threshold value, the tip is reconstructed and the volume of the tip estimate computed.

[Dongmo00] and [Todd01] suggest that there is typically a single threshold value at which the derivative of volume with respect to the threshold is significantly higher than at any other values. The threshold value is typically chosen by visual inspection of the volume plot, choosing a point slightly above the value at which the fastest change in tip volume occurs. Above this threshold value, the error in the tip reconstruction is said not to increase very quickly with increasing threshold [Dongmo00]. My experiments with synthetic data indicated multiple large jumps in the tip volume as the threshold was varied making it difficult to identify a unique single jump much larger than the others. Using synthetic AFM data, it was also possible to plot the actual error between the reconstructed tip and the tip used to simulate the synthetic data. The error was computed as the sum of squared differences between the heights in the reconstructed tip and tip used to simulate the synthetic AFM data. The tip volume and tip error for Villarrubia's algorithm are plotted as a function of the threshold value in Figure 4-4.



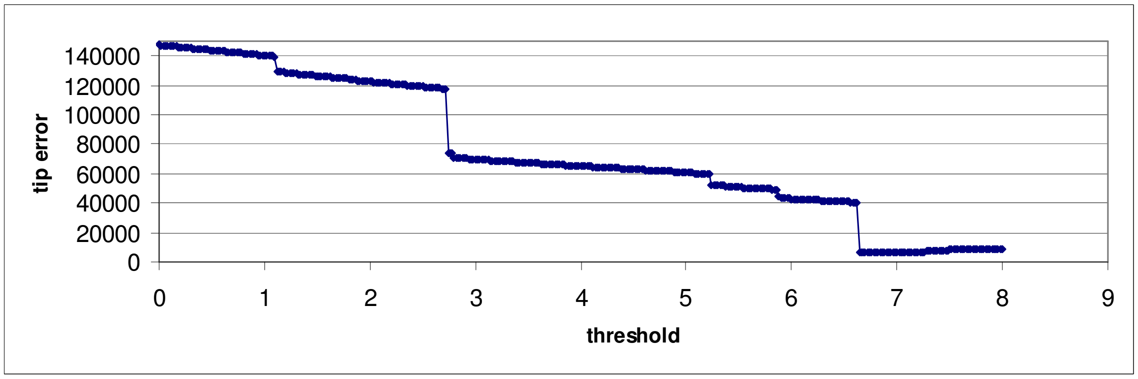


Figure 4-4: Plots of tip volume vs. threshold and tip error vs. threshold using Villarrubia's blind reconstruction algorithm

Villarrubia's algorithm iteratively reconstructs the tip shape and for each intermediate tip shape, for each point in the AFM image, the algorithm considers translations of the tip relative to the AFM image such that points on the tip coincide with the point in the AFM image. For each such translation, the algorithm determines whether or not the apex of the tip lies below the AFM image. The original algorithm does not account for uncertainty in the height of the apex point relative to the height of the AFM image. Neglecting this uncertainty may cause problems in the case of tips that have multiple projections or slightly flattened ends. Although this is not the usual shape of a tip, it was necessary for me to handle this case because the dulled tip I used had such a shape. When there are multiple local extrema near the tip apex, noise-induced oversharpening may occur that cannot be prevented by increasing the threshold value. An example constructed to demonstrate this oversharpening phenomenon is illustrated in Figure 4-5. The tip reconstruction result after considering two points labeled (1) and (2) in the AFM image in shown in the lower right of Figure 4-5. Noise causes the reconstruction to be essentially the intersection of the two separate peaks on the tip which makes the reconstruction much sharper than the actual tip shown in the upper left.

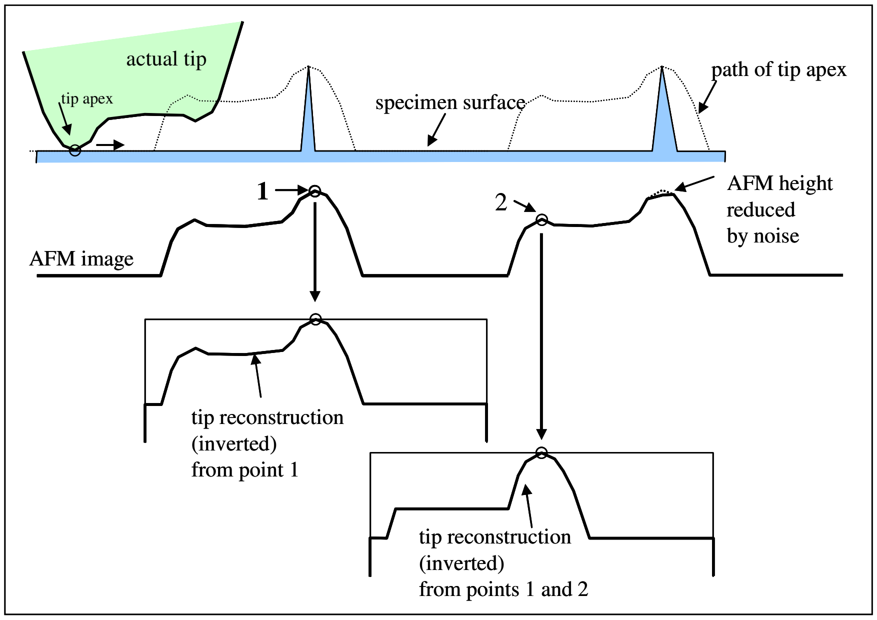
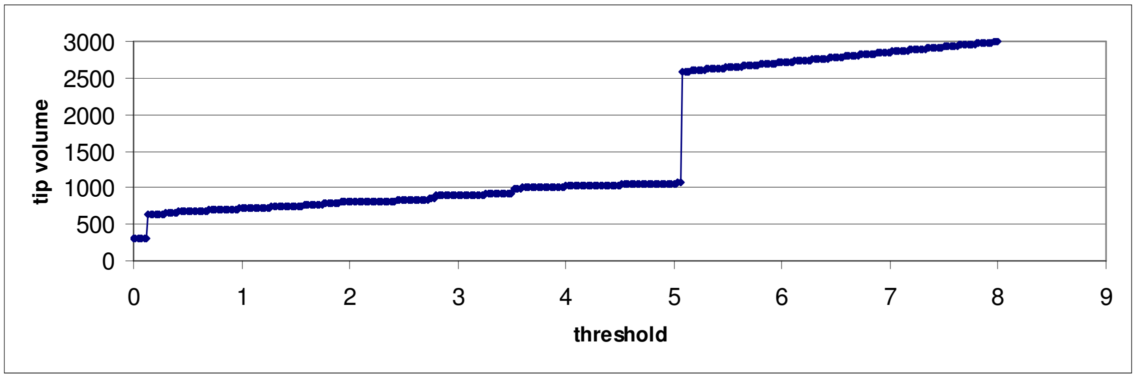


Figure 4-5: Problem case for blind tip reconstruction. The problem occurs after consideration of just 2 points in the AFM image (labeled 1 and 2). The first refinement of the tip using point 1 results in a valid outer bound on the true tip. If not for a perturbation of the AFM image due to noise, the tip shape from point 1 would be able to touch point 2 while having the tip apex below the AFM image surface and the algorithm would not change the tip after considering point 2. The perturbation causes the algorithm to erroneously erode the tip further at point 2.

To take into account the uncertainty in whether or not the tip apex is below the AFM image surface, I made a slight modification that causes the algorithm to be more conservative in the way it carves out the shape of the tip. In my modified version of the tip reconstruction algorithm, I introduce an apex tolerance parameter that represents the extent to which the apex of the tip is allowed to fall outside the surface as the tip scans the surface from below. Given a point on the AFM image, the existing algorithm took the union of all translations of the AFM surface umbra (volume under the AFM surface) such that the point lies on some point on the current tip estimate and the apex (the unique known point on the tip surface) lies beneath the AFM surface. This union is then intersected with the current tip estimate to refine the estimate. With my modification, this union also includes translations of the surface where the apex of the tip sticks above the AFM surface by a distance less than the apex tolerance. This has the effect of reducing the amount of refinement at each point but with a sufficiently large apex tolerance, it prevents the problem illustrated in Figure 4-5.

I repeated the procedure used to generate the graphs in Figure 4-4 using the modified tip reconstruction algorithm and an apex tolerance of 3nm. The resulting tip volume plot showed a single jump in the volume that was much larger than any others as the threshold was varied and this corresponded to the optimal threshold parameter where the tip error was minimized (Figure 4-6).



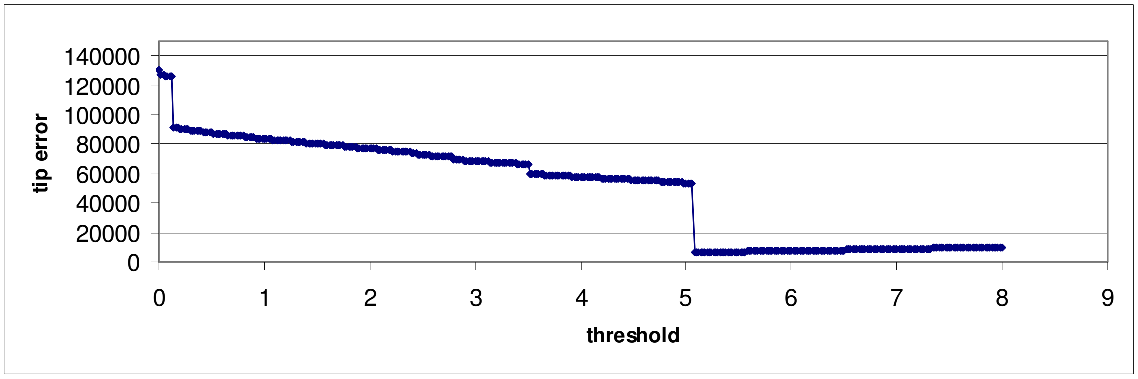


Figure 4-6: Plots of tip volume vs. threshold and tip error vs. threshold using Villarrubia's algorithm modified with an apex tolerance parameter to account for uncertainty in whether the apex of the tip lies below the AFM image surface. The apex tolerance for these plots was 3 nm.

The synthetic AFM image used for the experiments described above is shown in Figure 4-7a. Assuming a particular threshold and apex tolerance, the tip estimate is initially determined using a subset of the points in the AFM image that are local maxima (see Figure 4-7b for an example). The set of local maxima is found by first taking all points for which all 8 neighbors have a value less than or equal to the central value and at least 2 of the neighbors have a strictly lower value. Local maxima near the border are not considered because at those points we don't have complete information about the surface beneath the tip. After the initial estimate is found, it is refined using all points in the image (except near the border). Some of the final tip reconstructions along with the synthetic tip shape are illustrated in Figure 4-8.

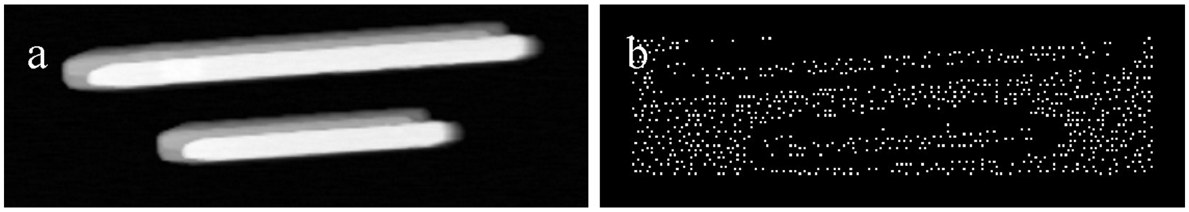


Figure 4-7: (a) synthetic AFM image including synthetic correlated noise designed to match a real AFM image (described in section 8.1); (b) set of local maxima (minus those near the border) in AFM image used for initial tip estimate

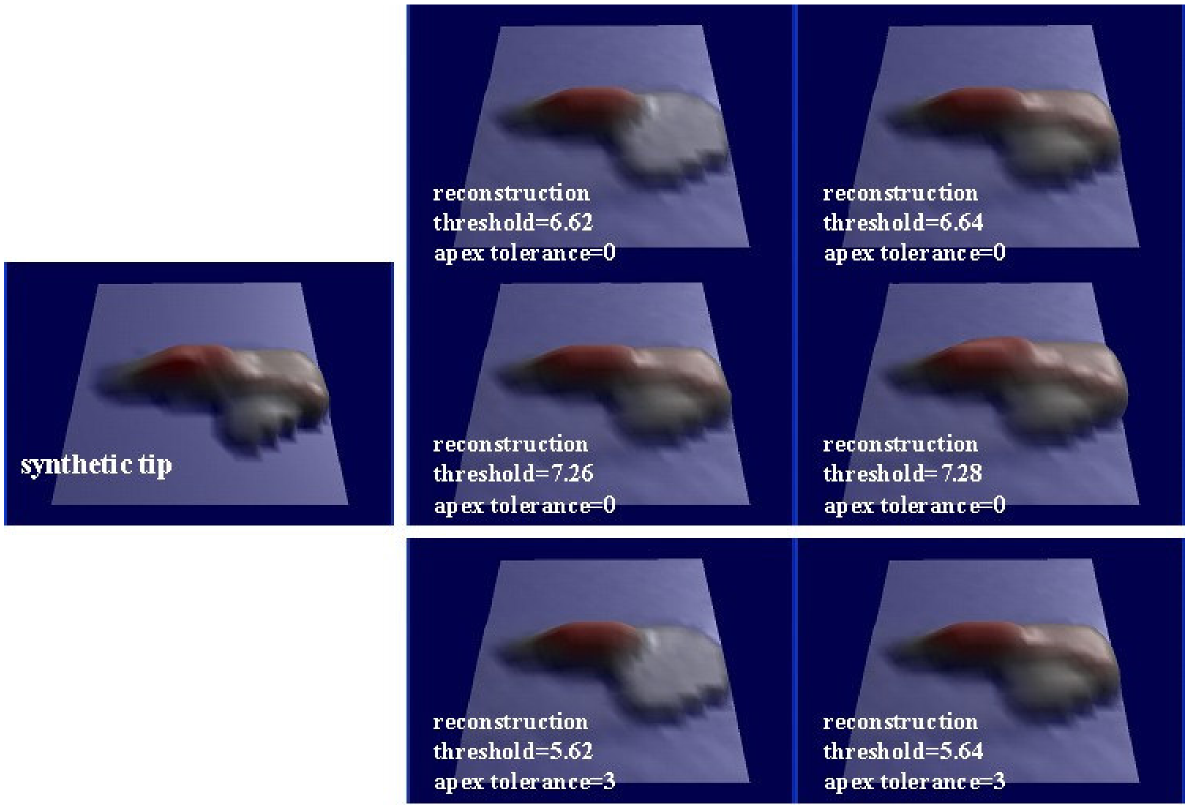


Figure 4-8: Results for final tip estimate using threshold values just above and below the two largest jumps in the tip volume plotted in Figure 4-4 and just above and below the largest jump in the tip volume plotted in Figure 4-6. Smaller thresholds reduce the tolerance of the algorithm to noise and result in oversharpening. (*z* is scaled by a factor of 2). The raised part of the plateau on the right side of the tip which disappears for threshold value 6.62 is critical to accurately predicting the appearance of the AFM image given an estimate of the surface.

4.3 Computing an Initial Surface Estimate

My algorithm first produces a flattened AFM image, *A*, from the original image using the method described in 4.1. It produces a tip estimate, $\hat{T}$, from *A* using the modified blind reconstruction algorithm described in section 4.2. I compute an initial estimate for the specimen surface by eroding the flattened AFM image, *A*, with the reconstructed tip image $\hat{T}$. The eroded image is computed using a discretized form of erosion described in [Villarrubia97] and defined as

$$H'_A(x,y) = \min_{i,j}\left\{A\left(x+i, y+j\right) - \hat{T}\left(i,j\right)\right\}$$

where *x, y, i* and *j* are integers, *x* and *y* range over the coordinates in the AFM image and *i* and *j* range over the coordinates of the tip image with center at (*i,j*)=(0,0). This estimate is used as the initial specimen model for the optimization described in section 9.4.

5 Monte Carlo Simulation of SEM Images

I considered using Monte Carlo simulation for computing the likelihood of a candidate surface reconstruction but to be practical for use at the core of an iterative surface optimization, the existing simulation code had to be accelerated by many orders of magnitude. I have implemented several techniques based on precalculation to accelerate the simulation. My main algorithmic innovation involves using a single sample of some number of electron trajectories in an infinite volume of material to estimate the signal for any arbitrary surface. In this chapter I first give some background on the Monte Carlo method as applied to SEM signal modeling and then describe my implementation of the Monte Carlo method and the acceleration techniques I developed in the hope of achieving a practical level of performance. Although Monte Carlo simulation is the most physically accurate model for SEM, and although my implementation is typically about 10 times faster than existing implementations, it is still too slow for this use. Details of this Monte Carlo simulation work were published in [Seeger03]. In my surface reconstruction method, Monte Carlo simulation provides examples of simulated SEM images from synthetic surfaces from which the relationship between the SEM signal and the surface topography is automatically learned and represented by a much faster simulation algorithm described in chapter 6.

5.1 Background and Theory

The Monte Carlo method is a widely used technique for finding an approximate solution to some problem by performing many randomized experiments in a computer. In many cases, solving a mathematical problem exactly is intractable but an acceptable approximation can be easily found by the Monte Carlo method. As applied to estimating an

SEM signal, the method consists of simulating many scattering electrons and computing various statistics such as how many electrons escape the specimen. For each electron, a trajectory is computed. At each step of a trajectory, a distance to the next scattering event (as described in section 1.5) and the outcome of that event (scattering angle or secondary electron generation) is computed by randomly sampling from a probability distribution based on some mixture of theory and experimental measurements.

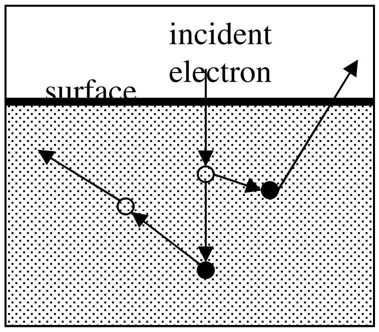


Figure 5-1: Monte Carlo simulation of electron trajectories

One common approach for accelerating the scattering simulation embodied in so-called *plural scattering* methods is to compress a sequence of trajectory segments into a single segment, considering the net effect of multiple scattering events [Joy95][Ly95]. Although there are many such potentially useful approximations, I have chosen to use the relatively detailed MONSEL model because it was the only model available to me that had previously been quantitatively matched to experimental data [Lowney95] [Lowney96a]. Also, MONSEL provides a realistic simulation of secondary electron emission that is not handled by many other simulators [Lowney96b]. Single-scattering models as used in MONSEL and several other programs attempt to model all elastic collisions and those inelastic scattering events most important for SE generation. Because some of the inelastic scattering is not taken into account, a continuous energy loss formula is commonly used to estimate the energy of an electron as it slows down in the solid. The energy loss formula is usually based on a combination of theory and experiment and the one used in MONSEL is notable for its reasonable behavior at very low energies [Lowney96b]. More detailed models that avoid the use of a continuous energy loss formula do exist but such models require data on the specimen material that is not available for most materials [Shimizu76][Ding96][Ding01].

Scattering simulations are based on the concept of a scattering cross-section that describes the probability that an electron will undergo a particular type of scattering event through interaction with a particular type of particle. For example, the elastic scattering cross-section gives the probability that an electron will be scattered by an atomic nucleus. The units of cross-section are area and the elastic cross-section represents the effective projected area of the nucleus onto the plane perpendicular to the electron's trajectory. For example, given a monolayer of atoms, the elastic cross section multiplied by the atomic surface density in atoms/area would give the probability that an electron traveling perpendicular to the monolayer would be scattered by an atomic nucleus. The mean free path, describing the expected straight-line distance between scattering events, is related to the scattering cross-section and the number of scattering particles per unit volume by

$$\frac{1}{\lambda}\left[\frac{1}{\text{dist}}\right]=\rho\left[\frac{1}{\text{dist}^3}\right]\cdot\sigma\left[\text{dist}^2\right]$$

(with the balance of units shown in square brackets) where λ is the mean free path in units of distance, ρ is the density of scatterers in units of 1/volume and σ is the cross-section per scatterer in units of area. At each step of the simulation one samples from an exponential distribution to get the distance to the next scattering event.

The concept of a cross-section can be broken down further. For example, the elastic cross-section as described so far gives the probability that an electron will collide with a nucleus and scatter in any direction. If we want to know the probability that an electron will scatter in a particular direction we can consider the probability that the scattering angle falls within some small solid angle containing that direction. However, as the solid angle shrinks about that direction, the probability goes to 0. The differential scattering cross section is the derivative of the cross section with respect to solid angle and gives the limit of the cross section per solid angle for each scattering angle. The integral of the differential scattering cross-section over all scattering angles equals the total cross-section. Dividing the differential scattering cross-section by the total cross-section results in a probability density for the scattering angle once a collision has taken place. When a scattering event is simulated, one randomly samples from this density to determine a new direction for the electron.

In MONSEL, there are three types of scattering events: elastic scattering by atomic nuclei, and two different kinds of inelastic scattering that result in the generation of secondary electrons. The elastic scattering cross section is computed using the empirical fit by Browning et al. to the Mott cross-section, a model based on quantum mechanical theory [Browning94]. One type of inelastic scattering event is the collision between the incident electron and a valence electron, the type of electron that is most weakly bound to an atom in the specimen. The cross section for the collisions that liberate a valence electron from an atom (or *ionize* the atom) is called the Moller cross section and is described in [Reimer85]. Another kind of inelastic scattering event involves plasmons. Plasmons are waves in the density of electrons in a conductor. Plasmons can travel in the gas of electrons in a conductor similar to the way sound travels in air. Some of the kinetic energy of an incident electron goes into producing a plasmon and the plasmon can in turn cause a secondary electron to be ejected from the specimen. Secondary electron generation through plasmon excitation is handled using the model of Kotera et al. [Kotera90]. When an inelastic scattering event is simulated, the energy of the resulting secondary electron is not subtracted from the primary electron. Instead, the expected loss is accounted for using a continuous energy loss model [Lowney96b]. Including inelastic scattering means that an electron trajectory is actually a tree describing the cascade of secondary electrons. My optimizations treat the scattering model as a black box and operate on the electron cascade as a tree of line segments.

Next, I describe two optimizations for computing intersections between the electron cascade and a specimen surface, followed by a description of the effect of parallelization and speed comparisons between my optimized techniques and other existing algorithms.

5.2 Implementation Details

5.2.1 Calculation of Ray-Surface Intersections

I represent the specimen surface by a two-dimensional array of height values. This array of points is tessellated with triangles to create a surface. Following an approach described in [Musgrave90], I implemented a ray intersection test for this surface that runs in O(max(N_X,N_Y)) time where N_X and N_Y are the x and y resolution of the grid. Identifying the triangles that must be tested requires a procedure similar to Bresenham's line drawing algorithm [Bresenham65] (Figure 5-2).

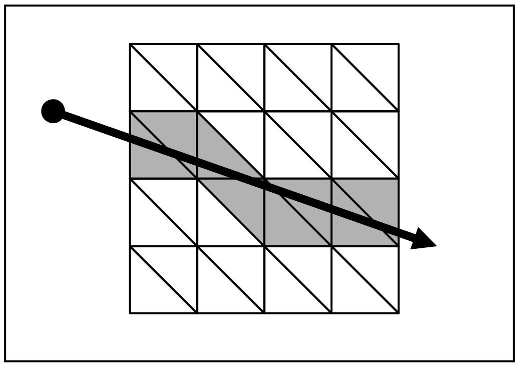


Figure 5-2: Overhead view of surface and ray to be tested for intersection. The calculation only requires testing intersections with the grey triangles.

5.2.2 Precalculated Electron Trajectories

I used two optimizations to reduce computation while leaving the accuracy unaffected. The first optimization is to precalculate and store electron trajectories in such a way that they can be reused later for an arbitrary surface. During simulation of an image, using precalculated trajectories lets me avoid complex calculations for determining probabilities of various scattering events and allows me to focus more computing resources on determining intersections of the electron trajectories with specimen surfaces as described in section 5.2.2.1. The second optimization is to precalculate intersections of the precalculated trajectories with various representative surfaces. To do this I consider the set of points on the surface initially intersected by the electron beam (each point corresponds to a pixel to be computed in the simulated SEM image). These initial intersections are clustered according to surface slope and for each cluster a representative surface is constructed as described in section 5.2.2.2. Stored intersection data for a representative surface are used to accelerate the intersection tests for the surfaces in the corresponding cluster. I essentially share computation between parts of the specimen that have similar topography.

5.2.2.1 Precalculation and Retracing of Electron Trajectories

I precalculate trajectories in an infinite volume of material and then retrace those trajectories at each point on the surface. The only difference between trajectories that are computed in an infinite volume and those computed with a surface is that the ones for the surface have segments of vacuum inserted where the electron has escaped and reentered and terminate prematurely if the electron escapes completely from the specimen. An example of a precalculated trajectory and its corresponding retraced trajectory for a particular surface is shown in Figure 5-3.

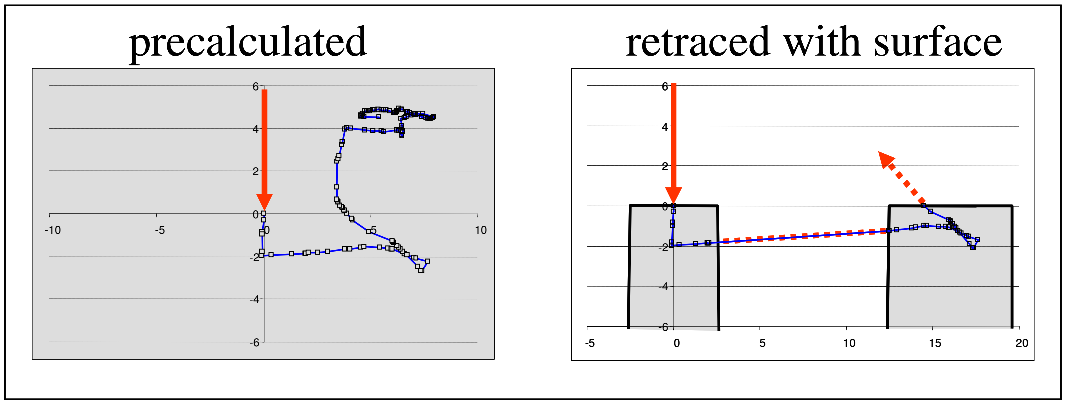


Figure 5-3: Example of precalculated trajectory (left) in an infinite volume of material (shaded gray) and its corresponding retraced trajectory (right) within material (shaded gray) bounded by a surface. The simulation in the infinite volume of material enables re-use of the trajectory for any surface topography. The incident beam is shown with a solid red arrow. A section of the trajectory traversing the vacuum is shown with a dotted red line and the point of escape from the specimen is shown with a dotted red arrow.

5.2.2.1.1 Reducing Overall Error in an Image Using a Larger Set of Trajectories

If one were to retrace the same set of trajectories for every pixel in an image, the resulting image would contain sampling bias. For a relatively small increase in computational cost one can produce a less biased result by precalculating a larger set of trajectories and retracing a random subset of the trajectories for each pixel. For example, I precalculated a set of 10000 trajectories and then for each pixel a random subset containing 1000 of the 10000 precalculated trajectories was retraced to determine the BSE and SE yields for that pixel. By using a larger set of precalculated trajectories, the statistical correlation of the error between any pair of pixels is closer to being independent (as is the case in a real SEM image) because on average one would expect each pixel to share only 10% of its trajectories with any other pixel.

5.2.2.2 Precalculation of Intersections with Approximating Surfaces

I have developed an algorithm for constructing a set of approximating surfaces with associated trajectory intersection information and then reusing this information to accelerate the calculation of intersections with an arbitrary surface. In the first step of the algorithm, a bounding box is constructed that contains all precalculated trajectories. The width of this bounding box is used to define a set of neighborhoods (one neighborhood per pixel in the image) containing each beam entry point on the surface. For a given pixel, when a beam electron enters the surface it is guaranteed not to escape the surface outside of the corresponding neighborhood because no simulated trajectory passed that far from the entry point. Next, the surface patches corresponding to the neighborhoods are clustered into a relatively small number of clusters according to slope (the clusters are labeled A, B and C in Figure 5-4). For each cluster, a plane is defined that has a normal equal to the average surface normal for all surface patches in the cluster and with a height such that it bounds all neighborhood surfaces in the cluster from below: no portion of a trajectory that lies below the plane approximation can escape from any of the surface patches sharing this approximation. Next, for every precalculated trajectory and for every plane approximation, the points in the trajectory just after the trajectory intersects the plane from above and just before the trajectory intersects the plane from below are computed and marked in a separate data structure.

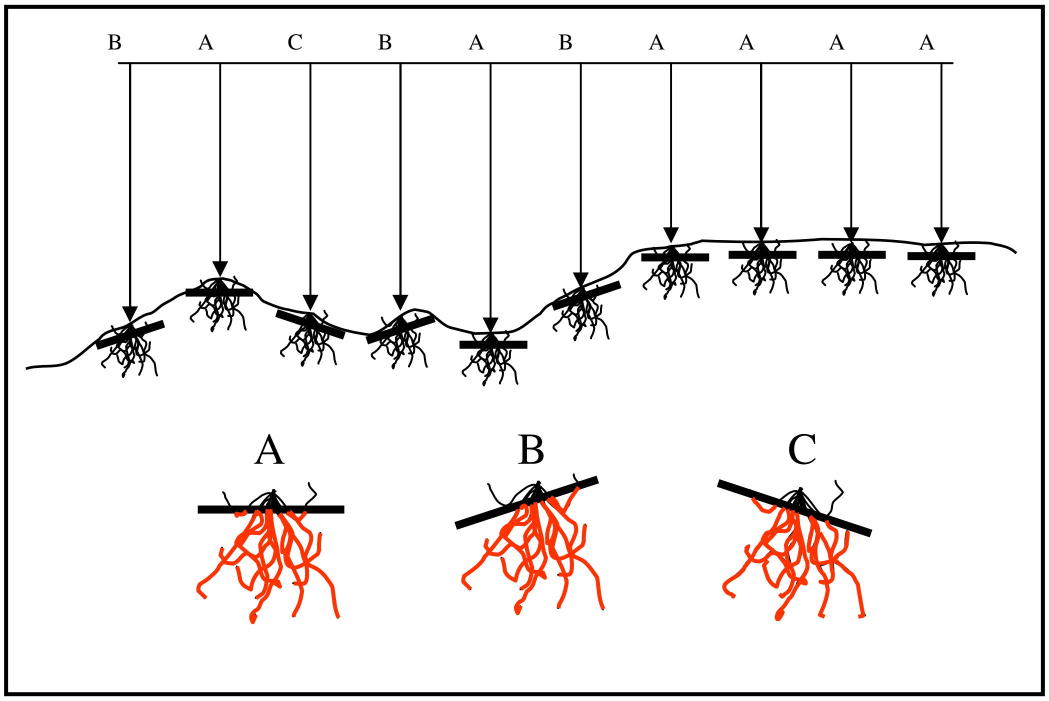


Figure 5-4: Similar regions on the surface can reuse calculations. In this case, a set of three approximating planes (A, B and C) have been matched to the surface patches corresponding to pixels in the image. Precalculated intersections with these planes allow us to ignore certain parts of the trajectories (indicated in gray/red) that are known not to intersect the surface.

During retrace of a trajectory for the actual surface the program uses the precalculated intersection information to skip over those parts of the trajectory lying under the corresponding approximating plane. The program only performs this optimization up until the first escape of an electron from the surface (whether it reenters or not) or material boundary crossing because the precalculated results are no longer applicable after such an event. As a result, the amount of speedup provided by the optimization depends on the fraction of trajectory segments occurring before an initial escape. It is least useful when an electron hits a very thin or sharp structure from which it can quickly escape. The benefits of this optimization also depend on the self-similarity of the specimen surface. Also, because I use planes as approximating surfaces, the greatest advantage from this technique is achieved when the surface is relatively flat at the scale of the interaction volume. At points on the surface where the surface has high curvature the improvement in efficiency is reduced. As a consequence, when the interaction volume increases due to increasing accelerating voltage, the performance gain from this technique may decrease as larger scale undulations of the surface fall inside the interaction volume. Other classes of approximating surfaces might be used to overcome this problem.

The trajectory segments are stored in a linear array format in an order such that every branch of the trajectory lies in a contiguous section of the array. Precalculated intersection data only describes the subset of trajectory segments that stick above an approximating plane rather than an exact calculation of intersection points with the plane. The subset is stored as an array of pairs of indices into the precalculated trajectory array. Each pair of indices represents an interval that lies above the approximating plane. Because each trajectory segment is stored as a vector instead of storing absolute positions of vertices in a trajectory, it is also faster to precalculate special vectors that allow the program to step to a new starting position when it skips an unneeded part of a trajectory that lies beneath the approximating plane.

5.2.2.3 Orthographic (parallel) Image Projection and Detector Characteristics

My simulator assumes an orthographic projection, which means that the incident electron beam comes from the same direction for all pixels in the image. While this is not a critical design decision, it does facilitate reuse of scattering results from one part of the surface to another. If a perspective projection were used, the precalculated trajectories would need to be rotated separately for each point on the surface. For a typical 10-micron scan range and 10 mm working distance, the change in angle is only 0.06 degrees over the entire scan. Given how close the real projection is to being orthographic this seems like a reasonable approximation.

I do not attempt to model detector characteristics in any detail but follow the approach of MONSEL in modeling the trajectory and detection of electrons after they escape from the specimen. If an electron escapes with energy greater than 50eV then it is assumed to follow a straight path in the vacuum and may reenter the specimen depending on its direction. An electron escaping with less than 50eV is assumed to follow a curved trajectory in the vacuum such that it is collected by the detector regardless of its direction [Lowney95]. If one of the higher energy electrons escapes without reentering the specimen it is counted as a BSE and if one of the lower energy electrons escapes it is counted as an SE.

5.2.3 Parallelization

To speed up the simulation, both the precalculation of trajectories and the retracing of those trajectories were performed in parallel on up to 30 processors. Each processor computed a complete image using a subset of the trajectories and the resulting images were added together to get the final result.

5.3 Performance Measurements

Several tests were run to compare the performance of my simulator with that of the popular CASINO simulator [Hovington02][Hovington97a][Hovington97b][Drouin97] and the original MONSEL Fortran code.

5.3.1 Comparison with Original MONSEL Fortran Code

For this test I had both programs compute 10000 1keV electrons hitting a flat Silicon surface as in the previous test but with calculations of secondary electrons enabled. This test was run on an SGI Reality Monster with 32 processors but the programs were run on a single processor (300 MHz IP27 MIPS R12000 CPU with R12010 FPU). The results are given in Table 5-1.

program:	MONSEL (F77 Fortran)	my program	my program + 10000 precomputed trajectories	my program + 10000 precomputed trajectories and precomputed intersections with a flat plane
# electrons	10000	10000	10000	10000
time (seconds)	24.7	23.2	4.21	0.122
electrons/second	405	431	2375	81970
memory used	--	3 MB	397 MB	410 MB
# trajectory segments traversed	--	3872277	3884425	6184

Table 5-1: comparison with original MONSEL Fortran code showing performance gain with precomputed trajectories and the best case performance using precomputed intersections

The measurements of my program using precalculated trajectories and intersections give a sense of the best-case performance possible when simulating an image of a more complex surface. In practice I found that the additional calculations required when a backscattered electron escapes and reenters the specimen and the imperfect fit of precalculated surfaces to the simulated surface greatly reduced the effectiveness of the precalculated intersection technique (see example in section 5.5 simulated at a rate of about 5000 electrons/sec compared to the maximum of 81970 electrons/sec for a flat surface). An improvement that I leave for future work is the extension of the precalculated intersection technique or an alternative method to help with the case when a backscattered electron reenters the surface.

5.3.2 Comparison with CASINO

This test compared versions of my simulator with and without trajectory precalculation and/or intersection precalculation with the CASINO program. Although CASINO may not be the fastest Monte Carlo code available, it is comparable to all other available code for Monte Carlo electron scattering simulation. CASINO was also a good choice for comparison because it uses a scattering model similar to the one used by MONSEL. CASINO was only available for a PC running Microsoft Windows so this test was run on a Pentium III 300 MHz PC with Windows 2000. I disabled x-ray calculations and display of electron trajectories in CASINO and disabled secondary electron calculations in my simulator to make the comparison as fair as possible. I also matched scattering models as closely as possible by making CASINO use the Browning fit to the Mott partial and total cross sections and making both programs terminate trajectories at a minimum energy of 0.05 keV.

The test consisted of computing trajectories for 10000 1 keV electrons striking a flat silicon surface. The results are given in Table 5-2.

program:	CASINO	my program	my program + precomputed trajectories	my program + precomputed trajectories and intersections
# electrons	10000	10000	1000000	1000000
time (seconds)	5	4.2	57	3.56
electrons/second	2000	2380	17543	280900

Table 5-2: comparison with CASINO computing backscattered electrons from flat silicon surface

5.3.3 Speedup using Multiple Processors

These tests were done on an SGI Reality Monster with 32 300 MHz MIPS R12000 CPUs and 16384 Mbytes of main memory.

In the first test using multiple processors, images were simulated of a very simple model, an infinite plane, to get a sense of the best-case performance. This test was done using 10000 precalculated trajectories for 1keV electrons hitting Silicon, precalculated intersections, 1000 retraced trajectories per pixel, and 256x256 pixels. A single processor took 754.7 seconds (86837 electrons/second) to do this computation while 30 processors running in parallel took 27.9 seconds (2,325,000 electrons/second). The speedup for various numbers of processors is shown in Figure 5-5.

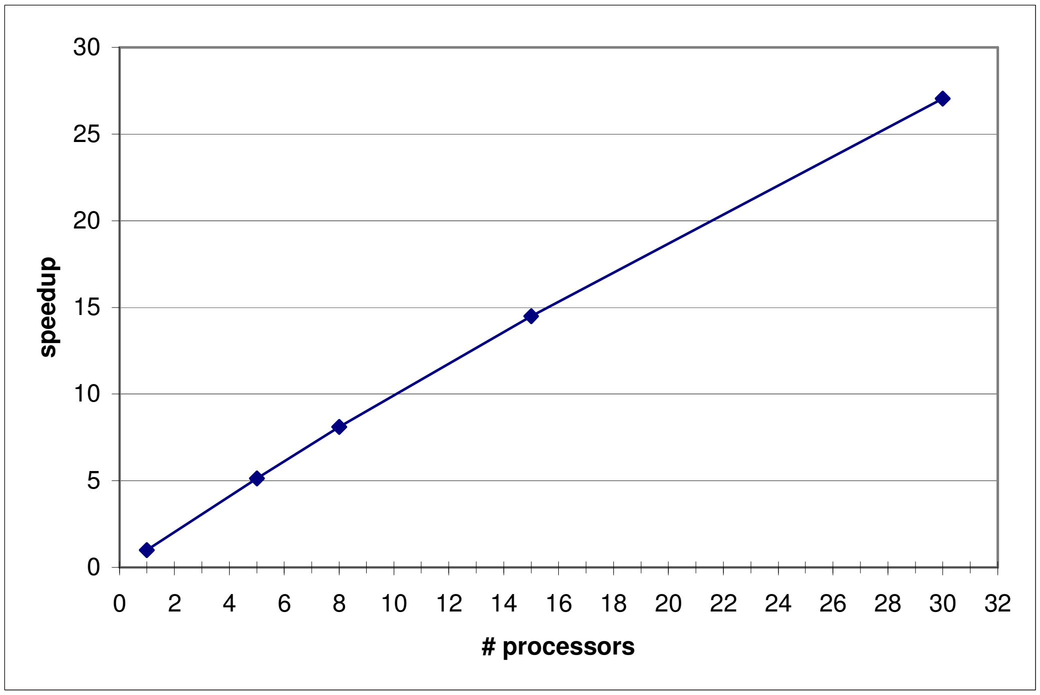


Figure 5-5: Speedup as a function of number of processors computing an image of an infinite flat Silicon surface using 10000 precalculated trajectories and precalculated intersections.

A similar test was performed for a more complex polygonal surface model composed of 79,202 triangles shown in Figure 5-6a.

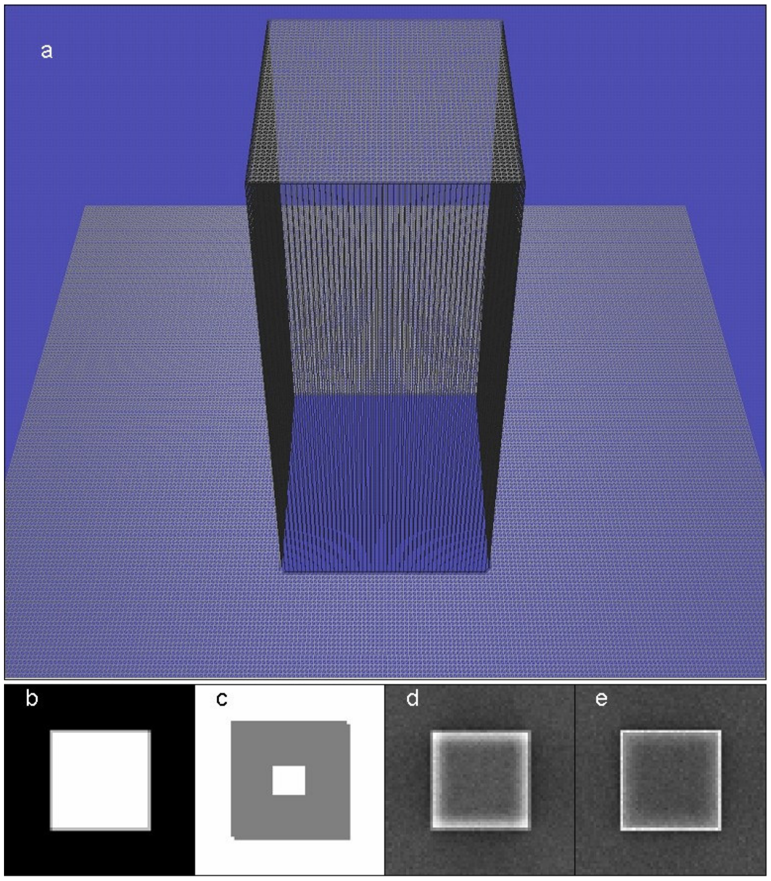


Figure 5-6: (a)Triangle mesh representing a specimen surface. The surface is 398nm x 398nm in x and y and the square projection is 200 nm high. (b) Image showing height of the surface for each pixel in the simulated images (region is 192nm x 192 nm). (c) Map of pixels for which precalculated intersections are used (grey indicates where they are not used). (d) Backscattered electron image. (e) Secondary electron image.

In this case 1000 trajectories were simulated for every pixel using 10000 precalculated trajectories and precalculated intersections. While the surface is 398 nm across, the view in the simulated images is limited to a square region in the center that is 192 nm across. The images generated are shown in Figure 5-6(d,e) and the speedup for various numbers of processors is shown in Figure 5-7. A single processor took 1620 seconds (2528 electrons/sec) while 30 processors running in parallel took about 93 seconds (43600 electrons/second) to do this simulation. I am not sure why there is a reduction in efficiency for larger numbers of processors but it might have to do with a change in patterns of memory access and the fact that the whole surface is accessed by each program thread.

It would be interesting to see if this problem occurs with the same geometry represented more compactly so it wouldn't be expected to have such a strong effect on memory use. In the SGI machine memory is divided into separate modules each shared by two processors. With only a single copy of the surface in memory and the fact that it won't fit in the relatively tiny data cache each processor must access the surface data over the internal network introducing delays into the intersection calculations. Therefore, one possible solution would be to have each processor create its own copy of the surface, forcing the surface to be replicated in each memory module.

This memory access issue does not affect precalculated trajectory data because this data is computed and stored locally by each thread of the program each with its own unique set of trajectories. If the simulation problem were divided by pixels instead of by trajectories, it would require each processor to access the trajectory data over the network. It would be much less practical to replicate the trajectories per processor because each 1keV trajectory takes about 50kBytes (500MBytes for 10000 such trajectories). Also, at 1keV precalculated intersections require a significant amount of additional memory when including secondary electrons because of the additional information that must be recorded each time a trajectory branches: 1600bytes/trajectory/plane (leaving out secondary electrons reduces the memory requirement to 13bytes/trajectory/plane). By contrast, my surface representation requires about 375 bytes per triangle or about 30MB for a surface specified with a 200x200 grid of points.

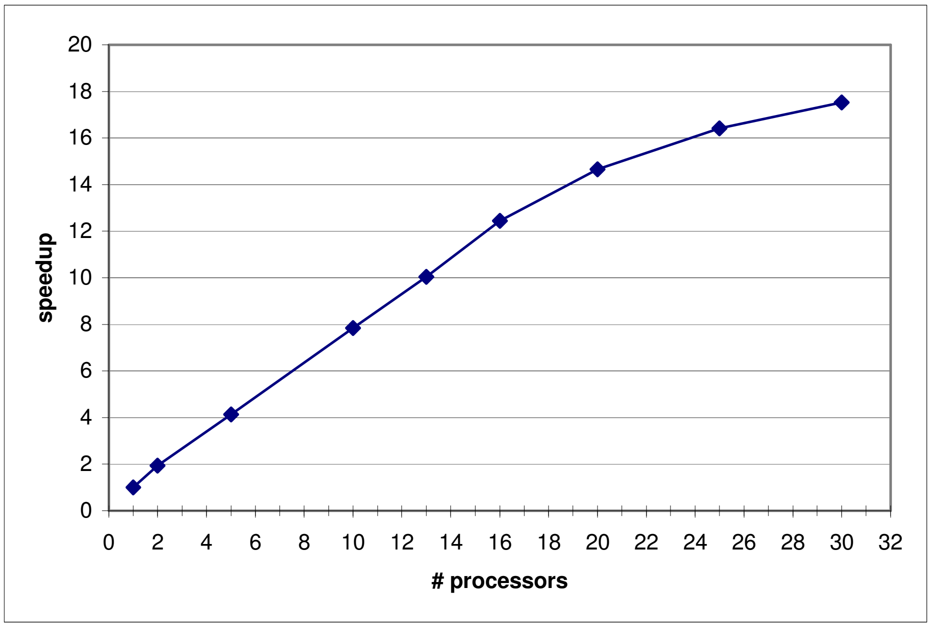


Figure 5-7: Speedup as a function of number of processors computing an image of a polygonal model with 79202 triangles

5.4 Object-Oriented Design and Extensibility

Two parts of the system that are worth noting from a software engineering standpoint are the basic electron scattering model and the specimen model. Both of these parts were designed to be very extensible. This flexibility enables one to easily adapt the simulation to use a different model for electron scattering or the specimen.

The scattering model is constructed in a modular fashion from any number of C++ objects describing various events that can occur. For the MONSEL model there are three different event objects: elastic scattering model based on Mott cross section, Kotera's inelastic scattering model for plasmon excitation, and Moller's model for valence electron ionization. Each event object is responsible for describing the total and differential scattering cross sections for a specific type of event including resulting products such as secondary electrons or x-rays. Events may be combined using a composite event object to form a single composite event that specifies everything that can happen to an electron in a given material. Different component scattering events may be from different models or may be the same model for different elements that compose the specimen material. Another part of the scattering model is a continuous process that may be used to describe changes in the electron energy over distance. The programmer has the option to use the continuous process object for a continuous energy loss approximation or the continuous process can be left out entirely if everything is to be handled through the event mechanism.

The parts of the program responsible for tracing electron trajectories make no assumptions about what specimen model or representation is used. The only requirement is that the material at every point in space is well defined. Testing the intersection of trajectories with boundaries is handled through a function pointer mechanism. The function specifying surface and material boundaries must take a ray or ray segment and return whether or not that ray intersects a boundary and if so it must also return the location of the intersection and what material if any is on the other side of the boundary. This was used by Charalampos Fretzagias to simulate images of nanometer-scale structures shown alongside real images of similar structures in Figure 5-8.

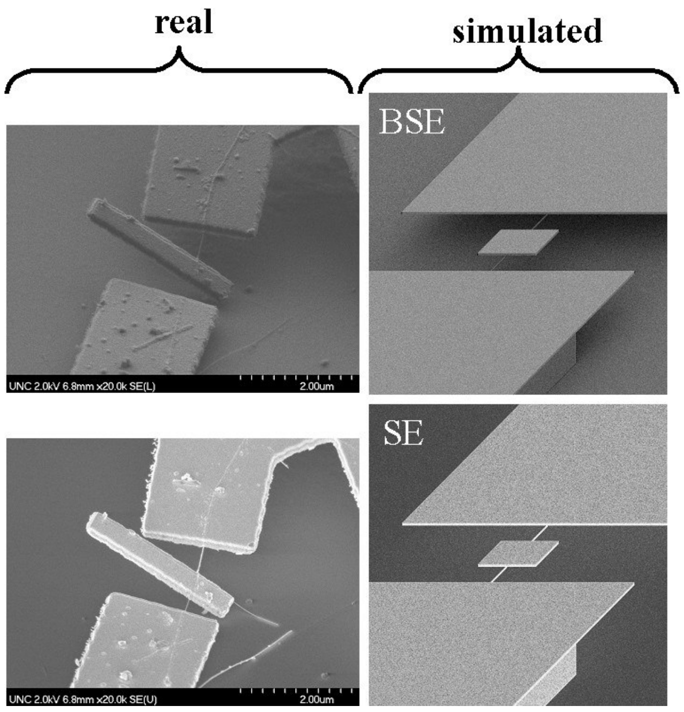

real
simulated
BSE
SE
UNC 2.0kV 6.8mm x20.0k SE(L) 2.00um
UNC 2.0kV 6.8mm x20.0k SE(U) 2.00um

Figure 5-8: Real and simulated images of paddle oscillator structures.

5.5 Examples of Image Simulations

Simulated SEM images were created based on topography measured with an AFM. In this case I took measurements of a surface composed only of silicon that had been etched with a grid pattern. This specimen was provided by John Dagata at NIST and was made using a dip-pen lithography technique described in [Chien02]. The AFM topography is shown as a 3D rendering in Figure 5-9a.

The 600x600 pixel image backscattered electron image shown in Figure 5-9c was generated using 16 processors on the SGI Reality Monster in 18 min 43 seconds at a rate of 19870 electrons/second/processor. This simulation included the precomputation of 10000 electron trajectories and retracing a random subset of 992 of these trajectories for each pixel in the image. This simulation also made use of precalculated intersection calculations for about 70% of the pixels (Figure 5-9b). The secondary electron image in Figure 5-9d took approximately four times longer to compute than did the backscattered electron image.

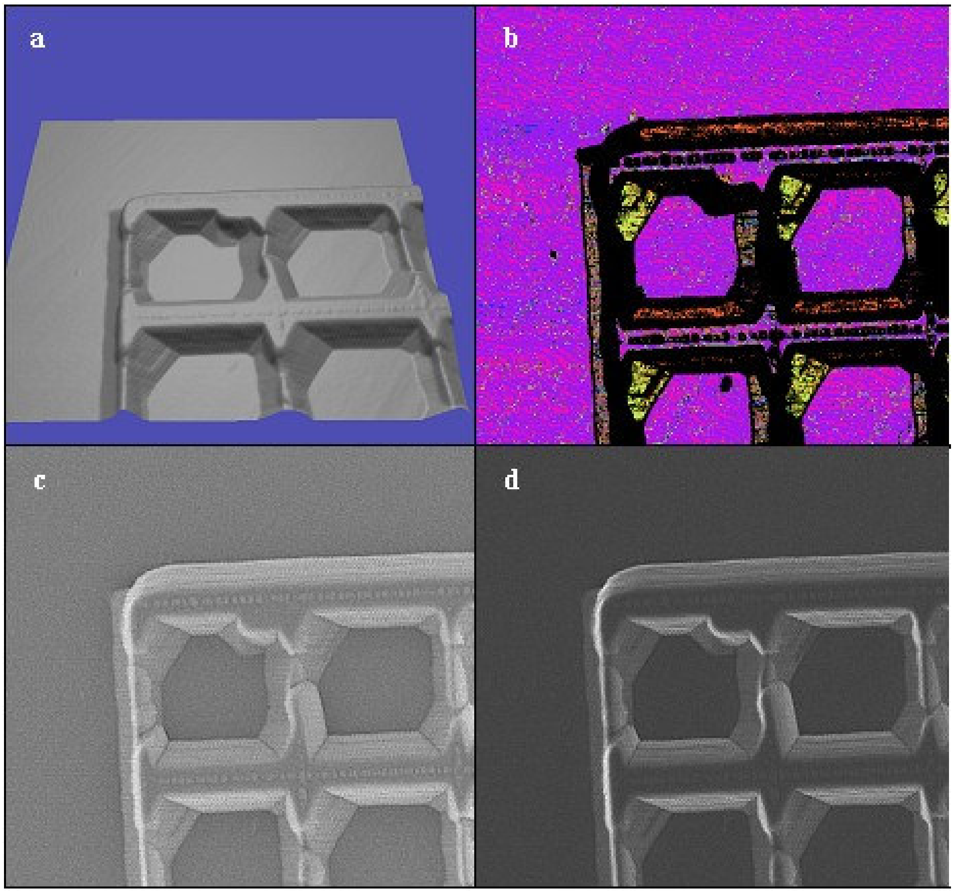


Figure 5-9: (a) 3D graphics rendering of surface acquired by an AFM. The surface is 3000 nm x 3000 nm and 260 nm high and composed of silicon. It is represented as a triangle mesh with 79202 triangles. (b) Map of areas with similar topography used to accelerate the SEM simulation. Each color corresponds to a set of surfaces that share scattering calculations. Black areas indicate parts of the surface that the algorithm chose not to optimize because they were not similar to a large enough set of other surface regions. (c) simulated backscattered electron image, (d) simulated secondary electron image

Examples of simulated images of objects defined by general 3D triangle meshes are shown in Figure 5-10 and Figure 5-11. A second example for a surface acquired by an AFM is shown in Figure 5-12.

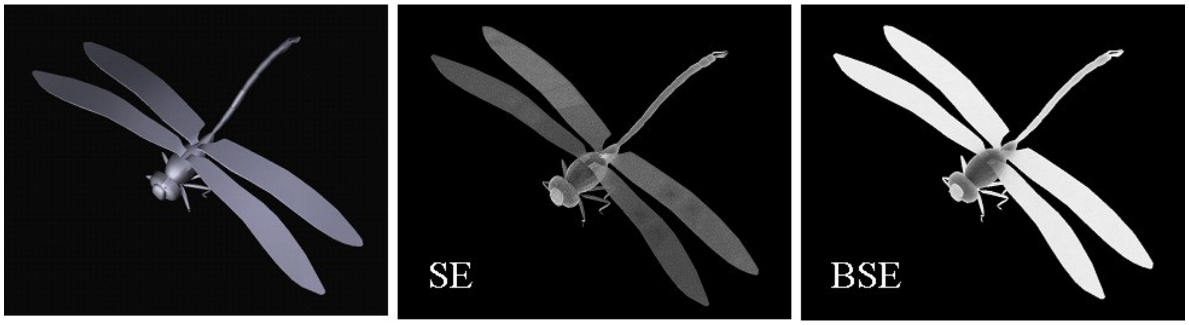


Figure 5-10: 3D graphics rendering of dragonfly model (left), simulated secondary electron image (middle) and simulated back-scattered electron image (right). These simulations were generated using general 3D mesh ray-intersection calculations.

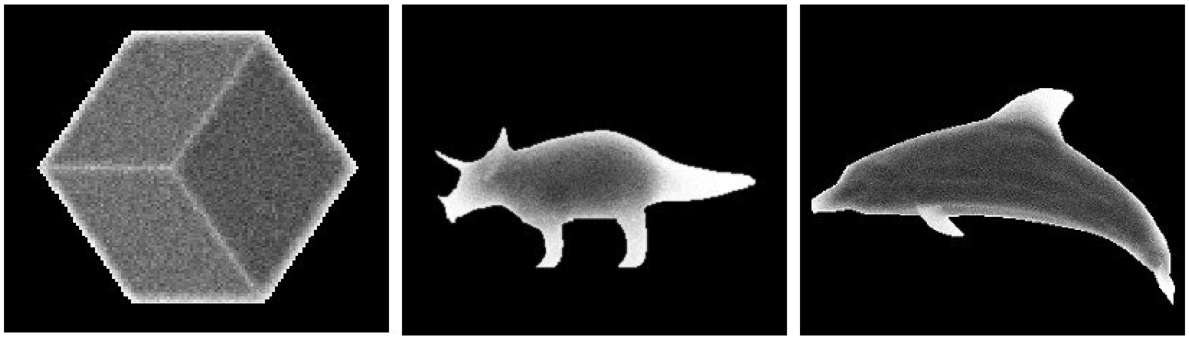


Figure 5-11: Simulated secondary electron images of cube (left), triceratops (middle), and dolphin (right). These simulations were generated using general 3D mesh ray-intersection calculations.

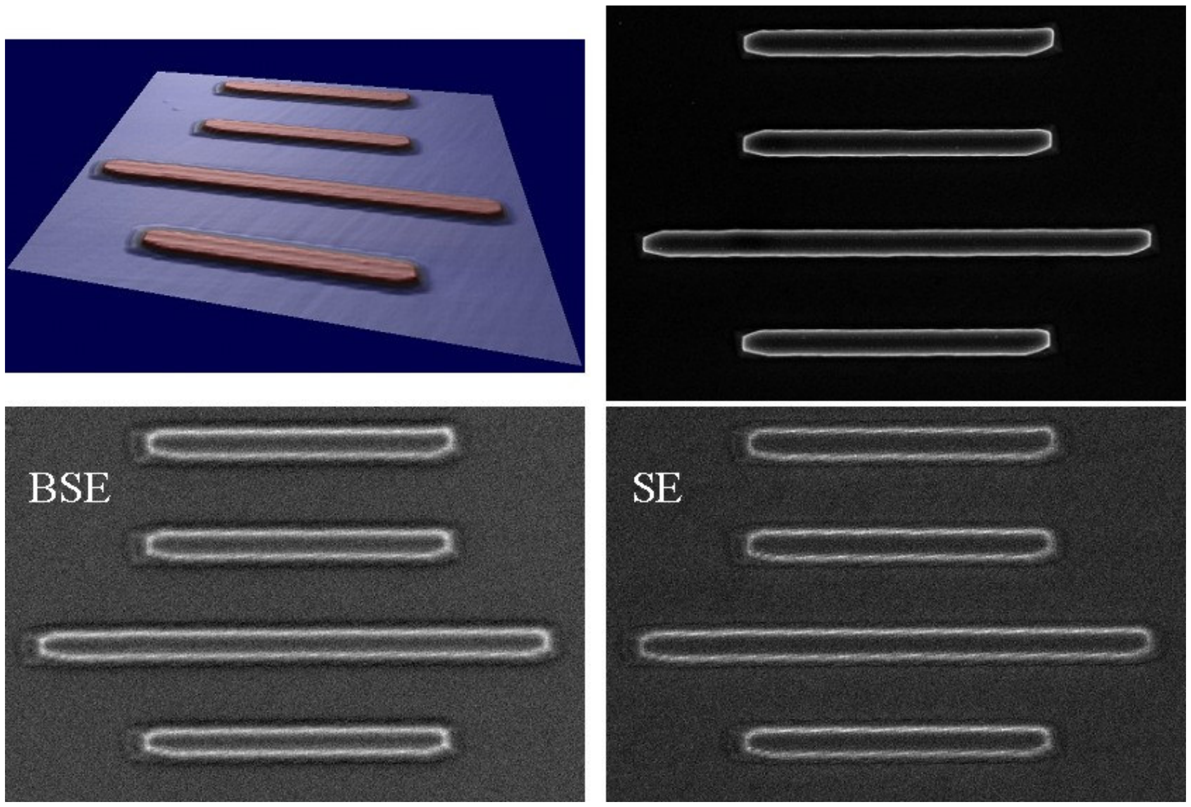


Figure 5-12: 3D graphics rendering of surface reconstructed from an AFM image (upper left), real SEM image of the surface (upper right), simulated BSE image of the AFM-based reconstruction (lower left), and simulated SE image of the AFM-based reconstruction (lower right). These simulations were generated using the height-field ray-intersection calculations.

6 Simulation of SEM Images Using Surface Slope and Curvature

6.1 Introduction

The optimization approach used in [Jones94] could in theory be adapted to use Monte Carlo simulation but this combination would be impractical because it would require the simulation of thousands of SEM images and thousands of hours of computation. In an effort to bridge the gap in performance/accuracy between the Monte Carlo model and slope-based models of SEM intensity, I developed a generalization of the slope-based approach that uses combinations of additional convolution-based filters (besides those computing slope) to improve accuracy. This approach provides a more powerful phenomenological model that more closely mimics the output of Monte Carlo simulation for nanometer-scale topography but is fast enough to be part of an iterative optimization algorithm similar to what has been used for shape-from-shading. In this chapter I first explain some of the inspiration for my approach by giving an intuitive description of what is measured by an SEM and how it relates to surface shape. After this I describe the new simulation method and provide performance measurements and simulation examples.

6.2 SEM Signal Components and Rationale for My Approach

For each pixel in an SEM image, the electron beam is focused at a corresponding point on the specimen surface. Electrons from the beam are scattered within the volume of the specimen and some escape to hit a detector where they contribute to the measured signal that determines the brightness of the pixel. As described in section 1.5, electrons detected in an SEM can be roughly divided into two groups by energy because there is a distinct division in the energy spectrum between high energy backscattered electrons (BSE) and low energy secondary electrons (SE). The SE component can be considered as having three subcomponents labeled SE-I, SE-II, SE-III [Goldstein92]. Because secondary electrons (SE) have much lower energy they can only travel a short distance before being stopped and as a result only escape the specimen if they are generated within a few nanometers of the surface. BSE on the other hand can travel much farther before escaping. The distance an electron travels in the specimen before it escapes determines in some sense the resolution of shape information carried by that electron. As described in section 1.5, the SE-I signal represents SE escaping very near the point where the electron beam enters the specimen surface and is considered to carry the highest resolution information. The BSE signal represents high energy back-scattered electrons that have traveled much farther in the specimen and escape from a much larger region of the surface. The SE-II signal represents low energy secondary electrons produced near the surface by escaping BSE and therefore provides a sort of amplifying effect that depends on the topography near the point of escape in the same way that the SE-I signal depends on the topography near the point of beam entry. The SE-III signal represents secondary electrons produced when BSE strike various parts of the SEM other than the specimen. The SE-III signal provides an amplifying effect for BSE that depends on the shape and materials of the SEM specimen chamber.

The models for SEM used in [Ikeuchi81] and [Jones94] were all based on the slope of the surface with no notion of the scale for measuring slope. If the SEM signal is modeled as a function of slope, then the slope should be measured at a scale (or scales) comparable to the electron-specimen interaction volume. This concept is illustrated in Figure 6-1. In Figure 6-1 I show constant slope approximations to a surface about a point where electrons hit the surface. The approximating surfaces are fit only to an area comparable to the area from which electrons escape the specimen. As I have described, the SEM signal consists of multiple components, each representing interaction with the specimen at different scales so I use estimates of the slope at multiple scales to predict the SEM signal. Although slopes at multiple scales would certainly be better than a single slope measurement for estimating the SEM signal, slope by itself is insufficient for modeling the appearance of sharp spikes, corners and pits on the surface so I also make use of local curvature estimated at multiple scales. I call this SEM model based on slope and curvature the filter bank model because it is expressed in terms of the outputs of a set of intermediate filters (the filter bank). Each filter computes either a slope or curvature value at a particular scale.

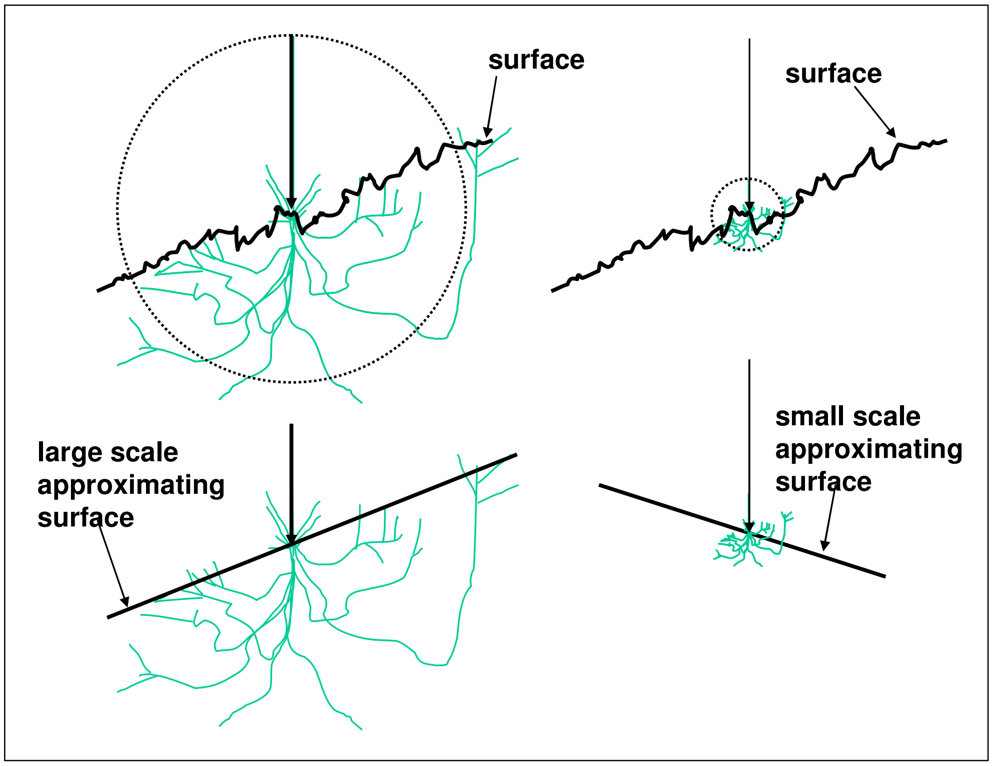

surface
surface
small scale approximating surface
large scale approximating surface

Figure 6-1: Taking into account the scale of the interaction volume when modeling the number of escaped electrons as a function of the local slope. When the interaction volume is large (left) the slope should represent the surface over a large neighborhood. When the interaction volume is small (right) the slope should represent the surface over a small neighborhood.

6.3 SEM Modeling Overview

The filter bank model consists of a set of filters each computing either a slope or curvature value and a function that maps the slope and curvature values to SEM intensity. A schematic of how the filter bank model computes an SEM image from a height image is illustrated in Figure 6-2. The mapping from slope and curvature values to SEM intensity is determined automatically by fitting to example Monte Carlo simulated images and real SEM images. Synthetic surfaces and corresponding Monte Carlo simulated images were used to determine a pair of filter bank models that mimic the BSE and SE outputs of the

Monte Carlo model. Using an approach described in section 6.7.2, a real SEM image along with AFM-based surface reconstruction was used to determine how to combine the BSE and SE images to best match the real SEM image. An outline of the method including the fitting procedure is illustrated in Figure 6-3.

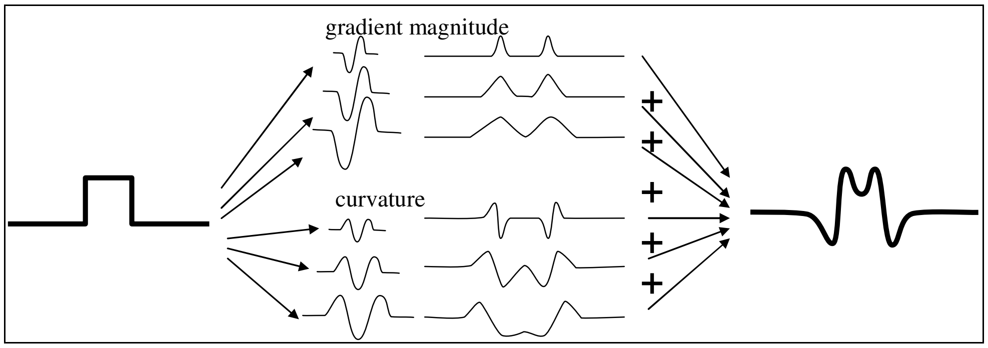


Figure 6-2: Filter bank model applied to simulating the SEM signal. Surface height (left) is filtered by each of a set of slope and curvature convolution filters to generate corresponding intermediate images that are summed to compute the simulated SEM signal (right). Parameters of the model control weighting of the intermediate images in the sum.

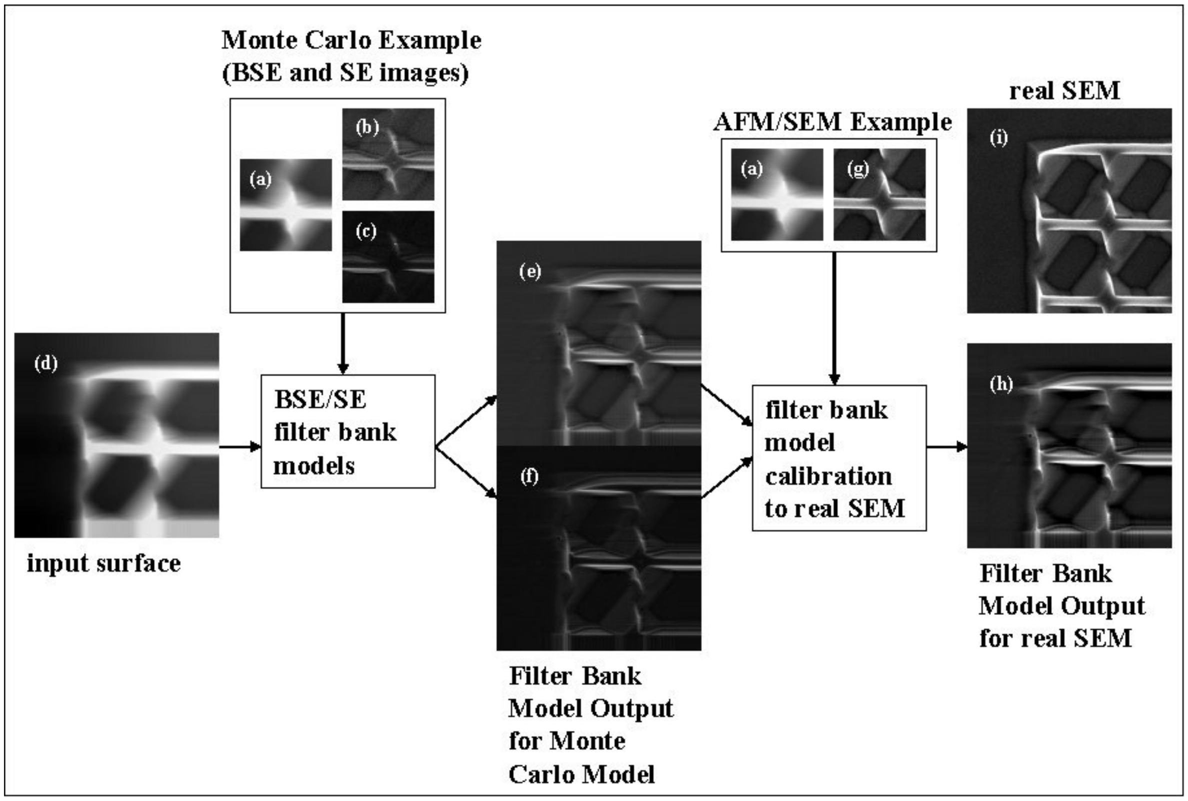

Monte Carlo Example
(BSE and SE images)
(a)
(b)
(c)
real SEM
AFM/SEM Example
(a)
(g)
(i)
(d)
input surface
BSE/SE filter bank models
(e)
(f)
Filter Bank Model Output for Monte Carlo Model
filter bank model calibration to real SEM
(h)
Filter Bank Model Output for real SEM

Figure 6-3: SEM model-fitting framework for filter bank model. A surface reconstruction from AFM (a) along with Monte Carlo simulated BSE (b) and SE (c) images made from this reconstruction are used to determine models that mimic the Monte Carlo simulation (BSE/SE filter bank models). The same AFM-based surface (a) and a real SEM image of that surface (g) are used to determine how the BSE and SE images (e) and (f) should be combined to best match the real SEM image. Given an input surface (d) (in this case a larger area of the same surface used to construct the filter bank model but in general a novel surface), the filter bank model computes an output (h) that is comparable to a real SEM image of that surface (i).

6.4 Estimating First and Second Derivatives of Height Using the Cubic Facet Model

I use a facet model to estimate slope and curvature as described by Haralick and Watson [Haralick81]. For the facet model, a *facet* is not just a planar face (as in the common definition) but can be any approximating function over a subregion of an image. One common choice for the approximating function, and the one that I use, is a two-dimensional cubic polynomial fit to a local neighborhood at each pixel in the image. Each pixel has a different facet represented by the cubic polynomial fit about the pixel. A local neighborhood for a pixel (x,y) in an image H is approximated by a two-dimensional cubic polynomial

$$H\left(x+t_x, y+t_y\right) \approx$$
$$f\left(t_x, t_y\right) = K_1 + K_2 t_y + K_3 t_x + K_4 t_y^2 + K_5 t_y t_x + K_6 t_x^2 + K_7 t_y^3 + K_8 t_y^2 t_x + K_9 t_y t_x^2 + K_{10} t_x^3$$

where $t_y \in Y$ and $t_x \in X$ represent row and column indices for a rectangular-shaped neighborhood with center at (0,0). For example, for a 5x5 neighborhood, $X = Y = \{-2, 1, 0, 1, 2\}$. K_1, K_2, K_3... K_{10} are functions of H (the image being fit) and the pixel location (x,y). The first and second derivatives of H are estimated as

$$\begin{aligned}
H_x(x,y) &\approx K_3(x,y) \\
H_y(x,y) &\approx K_2(x,y) \\
H_{xx}(x,y) &\approx 2K_6(x,y) \\
H_{xy}(x,y) &= H_{yx}(x,y) \approx K_5(x,y) \\
H_{yy}(x,y) &\approx 2K_4(x,y)
\end{aligned}$$

A detailed description of how I compute the cubic polynomial coefficients (K_1, K_2, K_3... K_{10}) is in Appendix D.2. I compute the first and second derivatives for 10 different neighborhoods centered about each pixel with sizes (in pixels) 5x5, 7x7, 9x9, 11x11, 15x15, 21x21, 29x29, 41x41, 59x59, and 83x83.

6.5 Estimating the SEM Signal from First and Second Derivatives of the Height

After computing the first and second derivatives of the height for all the neighborhood sizes, the set of derivative values for each pixel is transformed to two different values: the estimated BSE and SE yields (ratio of BSE and SE emitted to the number of incident electrons). Given examples of height field images and corresponding BSE and SE yield images computed by Monte Carlo simulation, my algorithm optimizes a pair of functions that estimate the BSE and SE yields from the derivatives of the height field at each pixel. The BSE and SE yields computed by Monte Carlo simulation are (in the limit as the number of simulated electrons goes to infinity) circularly symmetric. This means that the first derivatives can be first transformed to the gradient magnitude without loss of information needed to predict the BSE or SE yield. I also convert the second derivatives into a rotationally invariant form: the minimum and maximum principal curvatures (for an explanation of principal curvature, see Appendix D.3). For each pixel (x,y) in the height field and each neighborhood size n, I compute gradient magnitude ($G_n(x,y)$) and two principal curvatures ($\kappa_{+,n}(x,y), \kappa_{-,n}(x,y)$) as

$$G_n(x,y) = \sqrt{\left(K_{2,n}(x,y)\right)^2 + \left(K_{3,n}(x,y)\right)^2}$$

$$\kappa_{+,n}(x,y) = K_{6,n}(x,y) + K_{4,n}(x,y) + \sqrt{\left(K_{6,n}(x,y) - K_{4,n}(x,y)\right)^2 + \left(K_{5,n}(x,y)\right)^2}$$

$$\kappa_{-,n}(x,y) = K_{6,n}(x,y) + K_{4,n}(x,y) - \sqrt{\left(K_{6,n}(x,y) - K_{4,n}(x,y)\right)^2 + \left(K_{5,n}(x,y)\right)^2}$$

I used 10 neighborhoods (using a Gaussian weighting function as described in Appendix D.2) with sizes (in pixels) (5x5, 7x7, 9x9, 11x11, 15x15, 21x21, 29x29, 41x41, 59x59, 83x83) and standard deviations for the weighting function (in pixels) (0.6, 0.8485, 1.2, 1.697, 2.4, 3.394, 4.8, 6.788, 9.6, 13.58) so the BSE and SE yields for each pixel were each represented by functions of 10x3=30 different values. I chose to use linear functions to combine the gradient and curvature values into a BSE yield (FB_{BSE}) and SE yield (FB_{SE}):

$$FB_{BSE}(x,y) = d_{BSE} + \sum_{n=1..N} a_{BSE,n} G_n(x,y) + b_{BSE,n}\kappa_{+,n}(x,y) + c_{BSE,n}\kappa_{-,n}(x,y)$$

$$FB_{SE}(x,y) = d_{SE} + \sum_{n=1..N} a_{SE,n} G_n(x,y) + b_{SE,n}\kappa_{+,n}(x,y) + c_{SE,n}\kappa_{-,n}(x,y)$$

Equation 6-1: Filter bank models for simulating BSE and SE images

where N is the number of neighborhood sizes, and $G_n(x,y), \kappa_{+,n}(x,y), \kappa_{-,n}(x,y)$ are the gradient magnitude, maximum principal curvature and minimum principal curvature for neighborhood size n computed from the cubic fit at each pixel in the height field image. Using the method described in section 6.6, the parameters d_{BSE}, $a_{BSE,n}$, $b_{BSE,n}$, $c_{BSE,n}$ and d_{SE}, $a_{SE,n}$, $b_{SE,n}$, $c_{SE,n}$ (n=1..N) are determined automatically from training examples generated using Monte Carlo SEM simulation. These parameters will vary depending on the physical dimension of the pixels, accelerating voltage, and the beam shape.

6.6 Optimizing the Parameters of a Filter Bank Model

This section describes the fitting procedure I use to find the parameters of the filter bank model from example input/output images computed by Monte Carlo simulation. The procedure takes as input a height image (H_{train}) and a corresponding Monte Carlo simulated image (I_{train}). First the H_{train} image is filtered by the filter bank to generate a set of filter bank output images $\{F_i,\ i=1..N_f\}$. Next, I do a principal component analysis (PCA) of the output images to eliminate redundant dimensions in the filter basis [Jollife86]. The principal components are computed as the eigenvectors of the N_fxN_f correlation matrix of filter bank outputs. The eigenvalues of the correlation matrix measure the amount of variation in the filter bank outputs explained by the corresponding principal components. It is common to ignore those principal components with eigenvalues less than 1.0 but to provide sufficient accuracy I had to use components with eigenvalues down to 0.1. The risk in including components with small eigenvalues is that the problem of determining the optimal combination of these components can become poorly conditioned and accuracy may suffer due to numerical errors. The result of the PCA is a smaller set of linear transformations of the outputs of the filter bank:

$$F_{PC,i}(x,y) = \sum_{j=1..N_f} PC_i[j] \cdot F_j(x,y),\ i = 1..M$$

where $PC_i[j]$ is the jth element of the ith principal component and M is the number of principal components with eigenvalues greater than 0.1 ($M \le N_f$).

The next step is to find parameters D and $\{B_i,\ i=1..M\}$ such that

$$I_{\text{train}}(x,y) = D + \sum_{i=1..M} B_i F_{PC,i}(x,y)$$

To do this I solve the linear least squares problem

$$\begin{bmatrix} I_{\text{train}}(1,1) \\ I_{\text{train}}(1,2) \\ \vdots \\ I_{\text{train}}(N_x,N_y) \end{bmatrix} = \begin{bmatrix} F_{PC,1}(1,1) & F_{PC,2}(1,1) & \cdots & F_{PC,M}(1,1) & 1 \\ F_{PC,1}(1,2) & F_{PC,2}(1,2) & \cdots & F_{PC,M}(1,2) & 1 \\ & & \vdots & & \\ F_{PC,1}(N_x,N_y) & F_{PC,2}(N_x,N_y) & \cdots & F_{PC,M}(N_x,N_y) & 1 \end{bmatrix} \begin{bmatrix} B_1 \\ B_2 \\ \vdots \\ B_M \\ D \end{bmatrix}$$

Finally, I transform the B_i into the A_i as

$$A_i = \sum_{j=1..M} PC_j[i] \cdot B_j$$

The offset D and the filter bank coefficients A_i are then used in Equation 6-1.

For example, for the BSE filter bank model in Equation 6-1 the training input would be a height field and the training output would be the BSE image output from the Monte Carlo simulation with that height field. In practice, one must choose a particular order for the filter bank outputs $G_n(x,y), \kappa_{+,n}(x,y), \kappa_{-,n}(x,y)$ so I assume the order

$$\left(F_1(x,y), F_2(x,y), \dots, F_{N_f}(x,y)\right) = \begin{pmatrix} G_1(x,y), \kappa_{+,1}(x,y), \kappa_{-,1}(x,y), \\ G_2(x,y), \kappa_{+,2}(x,y), \kappa_{-,2}(x,y), \dots, G_N(x,y), \kappa_{+,N}(x,y), \kappa_{-,N}(x,y) \end{pmatrix}$$

With this order, the parameters in Equation 6-1 would be given by

$$d_{BSE} = D$$

$$\left(a_{BSE,1}, b_{BSE,1}, c_{BSE,1}, a_{BSE,2}, b_{BSE,2}, c_{BSE,2}, \dots, a_{BSE,N}, b_{BSE,N}, c_{BSE,N}\right) = \left(A_1, A_2, \dots, A_{N_f}\right)$$

6.7 Calibrating an SEM Model to a Real SEM Image

As described in section 2.2.1, the Monte Carlo simulation image does not quantitatively match experimental data so an additional transformation is applied to the simulated signal to make it match the experimental data. I first review the approach used in [Davidson99] and then describe my approach for matching Monte Carlo simulation image to experimental images. Because the filter bank model is constructed to mimic the Monte Carlo simulation, the same calibration procedure is used to match this model to experimental images.

6.7.1 Calibrating Monte Carlo Simulation to Experimental Data

Differences between Monte Carlo simulation and experimental data are handled in [Davidson99] by only comparing the simulation to the experimental data up to a slowly varying multiplicative factor:

$$E(x) = MC(x) \cdot F(x) + \varepsilon(x)$$

where E is the experimental data, MC is the Monte Carlo simulation data, ε represents the residual error between the simulation and experimental data, and F is a slowly varying function defined by

$$F(x) = \frac{\int_{x-R}^{x+R} E(x')\,dx'}{\int_{x-R}^{x+R} MC(x')\,dx'}$$

where R is an arbitrary range parameter. If one wishes to optimize a surface, minimizing ε, this approach could easily generate additional local minima that would make local optimization methods fail. For example, a simulated image that matches the experimental data up to multiplication by 0.5 would be scored the same as a simulated image that matches up to multiplication by 1.0. This approach makes sense for the exhaustive search used in [Davidson99] but because it creates multiple local minima it does not make sense for local optimization.

I scale and offset the simulated image to match it to a real image. Using the same notation as [Davidson99], I model the experimental image as

$$E(x) = MC(x) \cdot \text{gain} + \text{offset} + \varepsilon(x)$$

where the gain and offset are constants for an image. This makes sense given the fact that the real signal undergoes such a transformation that is set when the SEM user adjusts brightness and contrast controls. The lack of variation in the gain and offset across the image means that this approach cannot take into account non-uniformities (making one part of the image look brighter than another part) due to the SEM detector geometry, charging or surface contamination but I assume that such non-uniformities are not significant in my experimental data. The Monte Carlo simulation that I use actually gives two different signals, one for BSE and one for SE and I combine these to approximate the experimental signal using two different gain parameters:

$$E(x) = MC_{\text{BSE}}(x) \cdot \text{gain}_{\text{BSE}} + MC_{\text{SE}}(x) \cdot \text{gain}_{\text{SE}} + \text{offset} + \varepsilon(x)$$

where MC_{BSE} is the simulated BSE signal and MC_{SE} is the simulated SE signal.

I assume that gain_{BSE}, gain_{SE} and offset are constants that can be determined for a particular SEM and operating conditions including the brightness/contrast setting. One of the difficulties in using an SEM for quantitative analysis is that the amplifier gain and offset set by brightness/contrast settings are not typically stored or even accessible to the user except indirectly by visual inspection of the position of analog knobs. However, given an experimental SEM image of a specimen with known shape and materials, one could estimate the gain_{BSE}, gain_{SE} and offset parameters by simulating BSE and SE images (MC_{BSE} and MC_{SE}) and minimizing the cost function

$$f\left(\text{gain}_{\text{BSE}}, \text{gain}_{\text{SE}}, \text{offset}\right) = \sum_{x} \left[MC_{\text{BSE}}\left(x\right) \cdot \text{gain}_{\text{BSE}} + MC_{\text{SE}}\left(x\right) \cdot \text{gain}_{\text{SE}} + \text{offset} - E\left(x\right) \right]^2$$

Equation 6-2: cost function minimized to find calibration gain and offset parameters

I use this approach except instead of using a specimen with known shape I estimate the shape from an AFM image. Consequently, MC_{BSE} and MC_{SE} are only estimates of the simulated images for the actual specimen. I minimize Equation 6-2 using the least squares solver DGELS in the LAPACK library [Anderson94].

Given an AFM and SEM image of a specimen, I estimate the true surface shape by reconstructing the AFM tip shape and eroding the AFM image as described in chapter 4. Next, I align the eroded AFM image with the SEM image using the method to be described in chapter 7 and resample the heights by bilinear interpolation onto the SEM image grid. I then simulate SEM images from the resampled eroded AFM image and these simulated images become the MC_{BSE} and MC_{SE} in Equation 6-2.

6.7.2 Calibrating the Filter Bank Model to Experimental Data

To calibrate the filter bank model I followed the same procedure from the end of the previous section but replaced MC_{BSE} with FB_{BSE} and MC_{SE} with FB_{SE}. The formula for the final simulated image (that quantitatively approximates the experimental SEM image) is also just a linear combination of the same filter bank outputs

$$S_{FB}\left(x, y\right) = d + \sum_{n=1..N} a_n G_n\left(x, y\right) + b_n \kappa_{+,n}\left(x, y\right) + c_n \kappa_{-,n}\left(x, y\right)$$

Equation 6-3: Calibrated filter bank model

where

$$\begin{aligned}
a_n &= \text{gain}_{\text{BSE}} a_{BSE,n} + \text{gain}_{\text{SE}} a_{SE,n} \\
b_n &= \text{gain}_{\text{BSE}} b_{BSE,n} + \text{gain}_{\text{SE}} b_{SE,n} \\
c_n &= \text{gain}_{\text{BSE}} c_{BSE,n} + \text{gain}_{\text{SE}} c_{SE,n} \\
d &= \text{gain}_{\text{BSE}} d_{BSE} + \text{gain}_{\text{SE}} d_{SE} + \text{offset}
\end{aligned}$$

This calibrated filter bank model can be used to estimate the experimental SEM image for an arbitrary surface and this is the model that I used to simulate SEM images inside the surface reconstruction algorithm described in chapter 9.

6.8 Acceleration Methods and Performance Measurements

6.8.1 Sharing the Filter Bank between SE and BSE

The same filter bank is used to estimate both the SE and BSE outputs of the Monte Carlo simulation. While the SE signal tends to have higher spatial frequencies than the BSE signal, there is a significant amount of overlap because the SE-II component is approximately half of the total SE signal and the SE-II component is a sort of amplification of the BSE signal. Because almost all of the running time for the filter bank model is in the convolutions and square roots required to compute the slope and curvature estimates, by sharing this computation I can compute both the BSE and SE images in nearly the same time as it takes to compute either one, yielding a nearly 2x speedup.

6.8.2 Separability

For a 300x300 pixel input image and a variety of filter sizes ranging from 5x5 to 83x83, separating the coefficient filters into 1D convolutions in x and y gave a speedup of 12x over an implementation that did not take advantage of separability. The filters for K_4, K_5, and K_6 are directly separable and K_2 and K_3 can be decomposed into two parts (as described in Appendix D.2) and each part is then separable.

6.8.3 Vector Acceleration

The most time consuming operation is the 1D convolution required to compute the separable filter outputs. An additional speedup of 5x was achieved by using SIMD CPU instructions (MMX on Intel Pentium M, 1.5 GHz) that help to parallelize 1D convolutions and were accessed through the Intel Performance Primitives library.

6.8.4 Multi-resolution Gaussian-weighted Facet Model

Cubic fits over large neighborhood sizes require large filter kernels and convolution with these kernels becomes the main performance bottleneck. A subsampling scheme was developed to help speed up the calculation of cubic fits over large neighborhoods. For example, the cubic fit over a large neighborhood with a Gaussian weighting function with standard deviation 13.58 pixels may be approximated by subsampling the input height field by a factor of 16, computing the cubic fit with a Gaussian weighting function with a standard deviation of 13.58/16=0.84375 and then upsampling the result by a factor of 16. Subsampling was implemented efficiently by recursively subsampling by a factor of 2. The subsampling version of a cubic fit is in general not the same as the non-subsampling version because it is only an approximation based on a subsampled image. These differences are somewhat compensated by the optimization procedure described in section 6.6. I did not use this optimization in practice because among all the optimizations this is the only one that significantly affected accuracy and the additional speedup was not critical for my purpose.

6.8.5 Summary of Performance Gains

Here I compare the performance benefits of the various optimizations. The test input image (Figure 6-4) has 300x300 pixels. The output (Figure 6-5) is an SE and a BSE image both with 300x300 pixels. The filter bank consists of 3 filters for each neighborhood: gradient and the two principal curvatures. These are computed from the 5 linear and quadratic coefficients of the cubic polynomial fit for each neighborhood. There are 10 neighborhood sizes: 5x5, 7x7, 9x9, 11x11, 15x15, 21x21, 29x29, 41x41, 59x59, and 83x83. Thus, in total there are 50 2D convolutions for the coefficient filters and the output of these convolutions gets converted to 30 slope/curvature basis images. The initial implementation with no optimization took 110 seconds to run. After all optimizations were implemented, the running time was 0.12 seconds giving a speedup of about 912x. Compared with Monte Carlo simulation using 1000 electrons per pixel, this represents a speedup of about 150,000x. The performance gains and running time after each optimization technique was implemented are summarized in Table 6-1 in the order they were added.

acceleration technique	incremental speedup	cumulative speedup	running time
sharing filter bank between SE and BSE images	1.9x	1.9	58 seconds
separability of filters	12x	22.8	4.83 seconds
SIMD hardware	5x	114	0.97 second
miscellaneous optimization using profiler (Intel VTune)	2x	228	0.48 seconds
subsampled filtering	4x	912	0.12 seconds

Table 6-1: Summary of performance gains. The incremental speedup refers to the speedup when using an acceleration technique compared with not using that acceleration technique but keeping the rest of the implementation the same.

6.9 Simulation Results

6.9.1 Comparison with slope function

Both the filter bank model and a slope function described in Appendix Appendix E were used to simulate SEM images from the input height field shown in Figure 6-4.

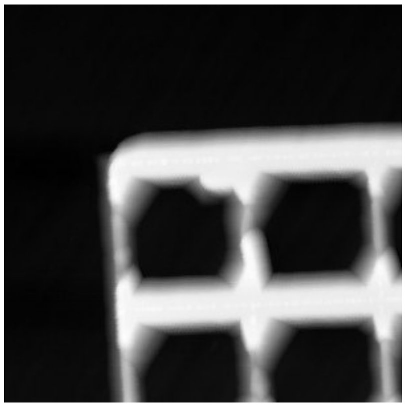


Figure 6-4: Test input height field

Figure 6-5 shows BSE and SE SEM images predicted by Monte Carlo simulation, the piecewise linear function of slope constructed by fitting to the Monte Carlo simulation using the two-plane method described in Appendix Appendix E, and filter bank models fit to the Monte Carlo simulation.

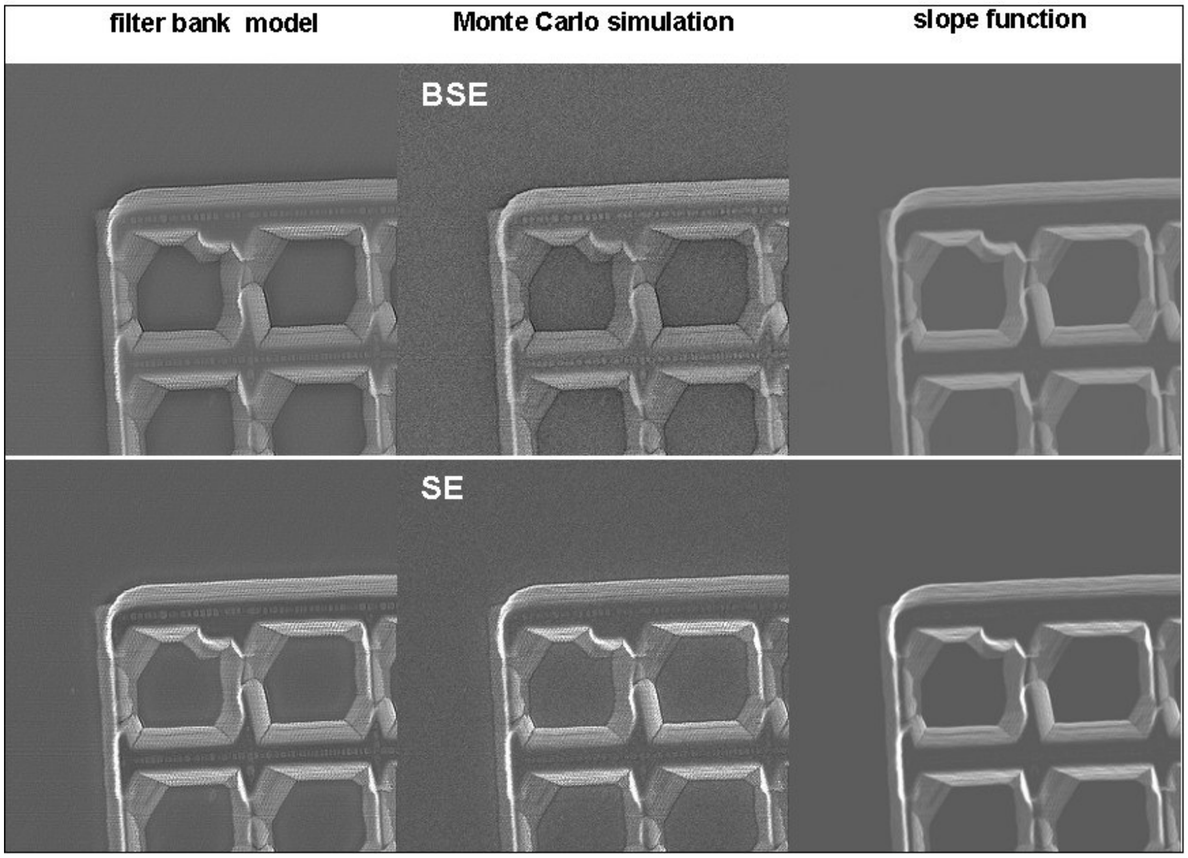


Figure 6-5: Comparison of simulated SEM images output by Monte Carlo model, a function of slope (two-plane slope model - see Appendix Appendix E), and the filter bank model. A single intensity scale is used for all the BSE images and another intensity scale is used for all the SE images. Note especially the darkening in crevices and the bright high-curvature bumps present in the filter bank and MC simulations that are missing in the slope model.

Figure 6-6 shows the difference in error between the slope function and the filter bank model. Large errors for the slope function in areas of high curvature are significantly reduced for the filter bank model. The differences shown in Figure 6-6 are described more quantitatively in Table 6-2. The errors listed in Table 6-2 are relative errors in units of the true signal value as estimated by the Monte Carlo simulation. The signal values are SE or BSE yield which refers to the ratio between the number of SE or BSE and the number of incident electrons. When averaged over the whole image, the relative errors for the filter bank model are about half those for the slope function but because the error is concentrated in a relatively small area where the surface is highly concave or convex (as can be seen in Figure 6-6), the difference must be significantly larger in these parts of the image.

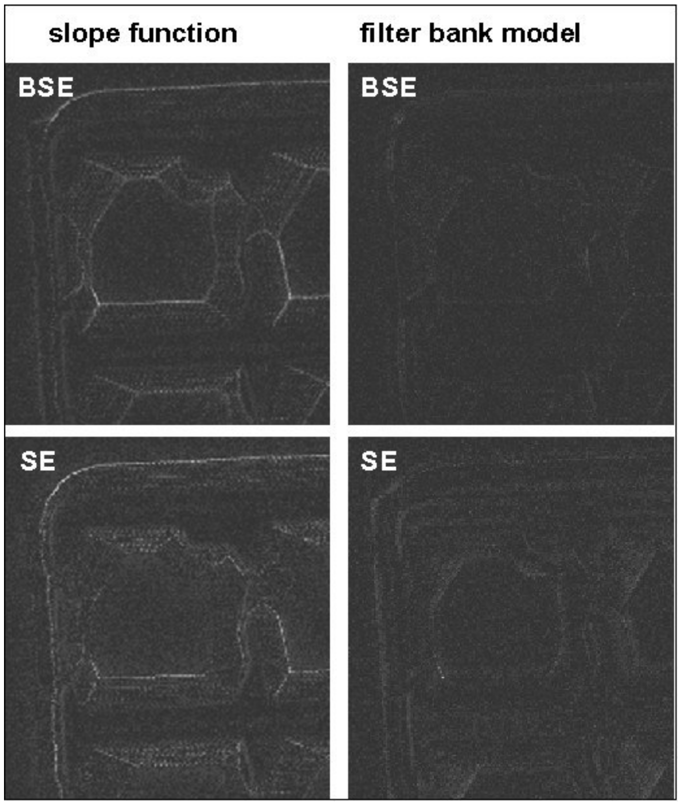


Figure 6-6: Estimates of absolute value of relative error in intensity from slope function (left) and for the filter bank model (right). The intensity scales are identical between the two BSE images and between the two SE images. The relative error is with respect to the Monte Carlo images.

	mean	max	median	std. dev.
$\left\|\left(BSE_{slope} - BSE_{MC}\right)/BSE_{MC}\right\|$	0.0986	2.89	0.0731	0.104
$\left\|\left(BSE_{FB} - BSE_{MC}\right)/BSE_{MC}\right\|$	0.0545	0.480	0.0446	0.0440
$\left\|\left(SE_{slope} - SE_{MC}\right)/SE_{MC}\right\|$	0.110	2.32	0.0872	0.109
$\left\|\left(SE_{FB} - SE_{MC}\right)/SE_{MC}\right\|$	0.0714	1.22	0.0589	0.0569

Table 6-2: Some statistics from the images shown in Figure 6-6. The slope subscript signifies the image computed using the function of slope, the FB subscript signifies the image computed using the filter bank model, and the MC subscript signifies the image computed by Monte Carlo simulation.

The superior ability of the filter bank model to emulate Monte Carlo simulation compared with a function of slope becomes more evident as the structures get smaller. This was tested using the example in Appendix Appendix E (shown in Figure E-3 and Figure E-4). Before doing this test, the filter bank model had to be trained using images with the same pixel size as the images to which it would be applied. The training images for the two pixel sizes are shown in Figure 6-7 and Figure 6-8. The results from applying the filter bank models to the test structures are shown along with the slope function and Monte Carlo output in Figure 6-9 and Figure 6-10. These results demonstrate that the filter bank captures qualitative aspects of the Monte Carlo simulation that the slope function cannot. For example, there is no way for the slope function to predict a lower intensity for highly concave parts of the surface than for flat parts of the surface as the filter bank model does.

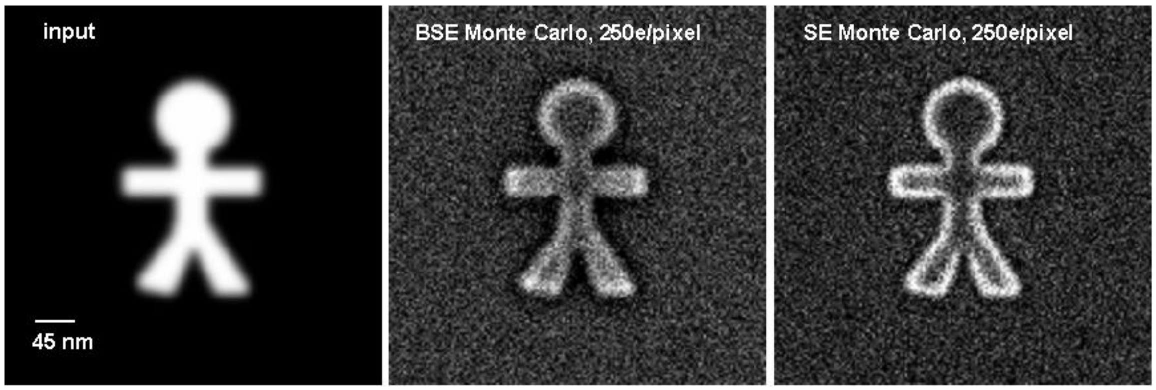


Figure 6-7: Filter bank model training data for 3nm/pixel resolution. Images are 150x150 pixels. 250 electrons/pixel were used to generate the training output. The input structure is 20nm high.

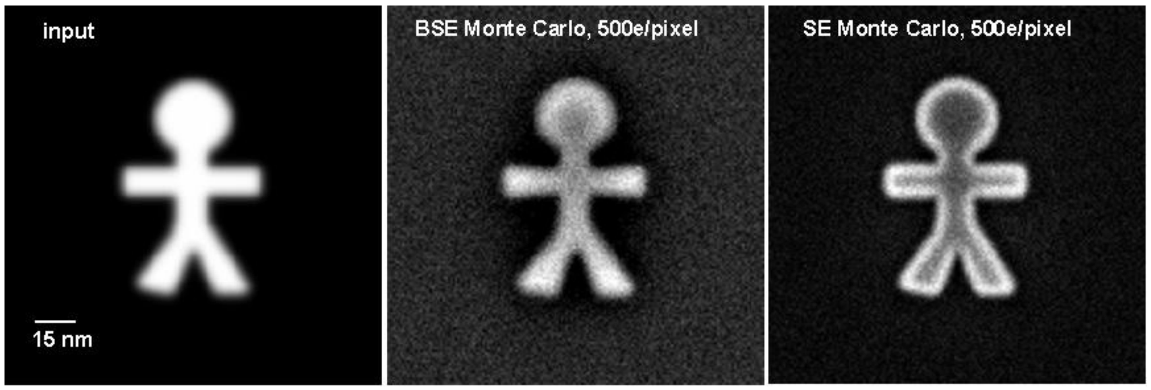


Figure 6-8: Filter bank model training data for 1nm/pixel resolution. Images are 150x150 pixels. 500 electrons/pixel were used to generate the training output. The input structure is 20nm high.

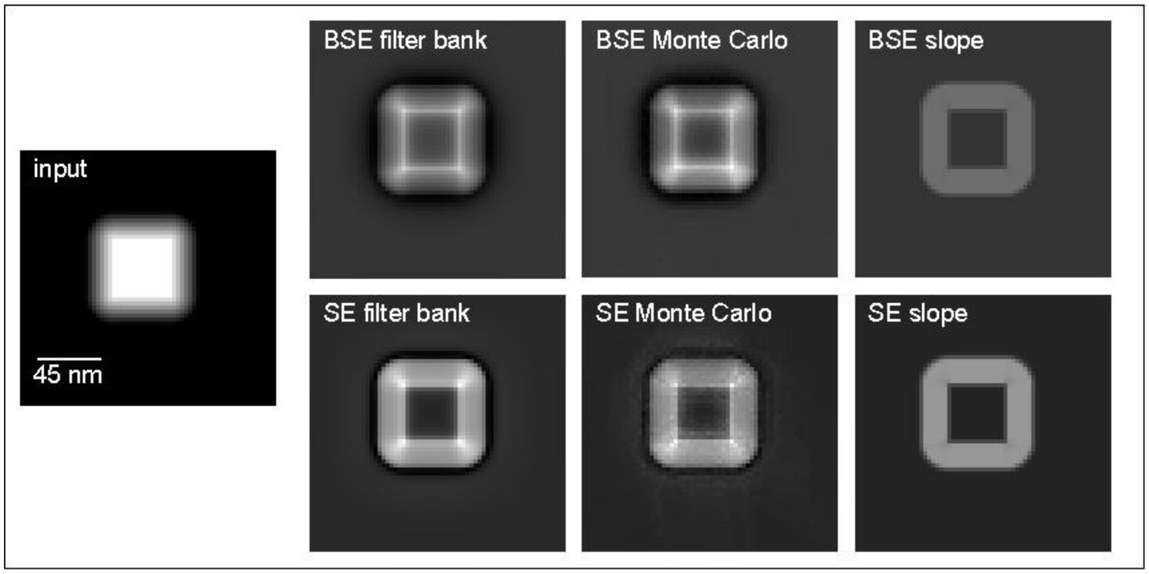


Figure 6-9: Monte Carlo simulation compared with a function of slope and the filter bank model on a 20nm high raised square 45nm wide at the top. Images are 64x64 pixels and 3nm/pixel.

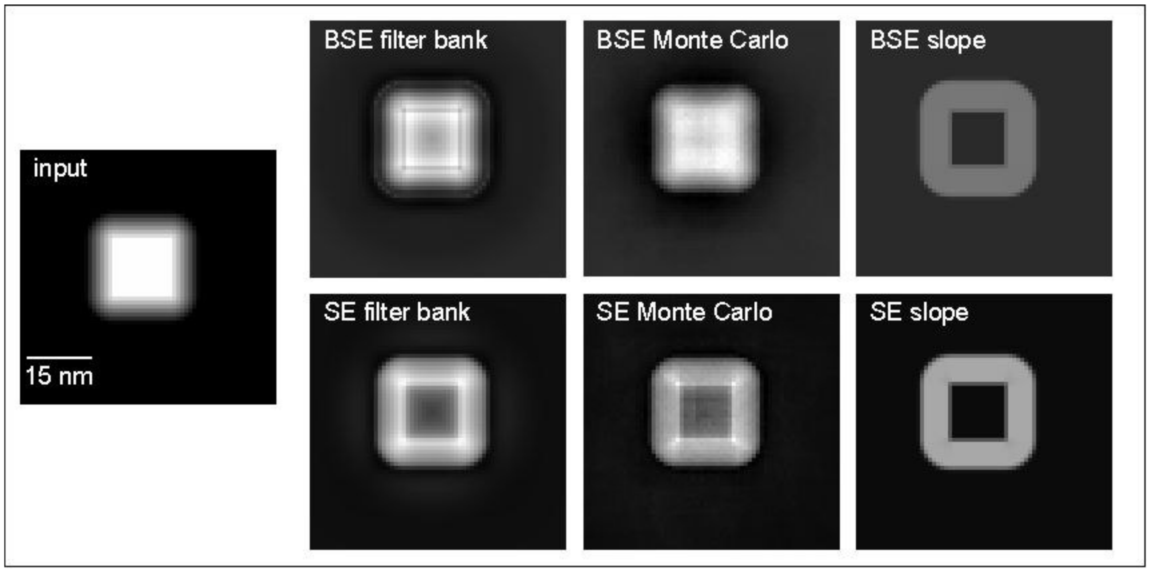


Figure 6-10: Monte Carlo simulation compared with a function of slope and the filter bank model on a 20nm high raised square 15nm wide at the top. Images are 64x64 pixels and 1nm/pixel.

7 Registration of AFM and SEM Images

In order to relate the intensities in an SEM image to those in an AFM image of the same specimen one must define a coordinate transformation between the images. Registration methods solve for a transformation that maps points in one image to corresponding points in another image. In the first part of this chapter I review some of the previous work in registration focusing on methods relevant to multi-modal registration and registration involving either AFM or SEM data. In the second part I describe the method that I used to register AFM and SEM images.

7.1 Background

The two main aspects of a registration method are the class of transformations that one considers and the cost function that ranks transformations and determines the optimal transformation. In addition, some sort of optimization method is needed to actually find the optimal or near optimal transformation.

7.1.1 Registration Cost Functions

As described in the review in [Maintz98], registration methods relying only on image content can be classified into three groups: landmark-based methods, segmentation-based methods, and intensity-based methods. These categories are distinguished by the type of cost function that determines the optimal transformation.

7.1.1.1 Landmark-Based Cost Functions

In landmark-based methods, one identifies a set of pairs of corresponding points, one from each image. One simple cost function is the sum of squared distances between corresponding points and this function can be efficiently minimized by solving a linear least squares problem (see section 7.2.4). Landmark-based methods are commonly used to determine an affine or rigid transform that serves as a starting point for local optimization of a more complicated cost function that would get stuck in a suboptimal local minimum without a good starting point. An extension by Rohr et al of the basic landmark-based approach takes into account orientation attributes and anisotropic errors for each landmark [Rohr99]. An example of anisotropic error is when a landmark position can be much more accurately determined in the x-direction than it can be determined in the y-direction. One would ideally want to weight the x-position information more than the y-position in calculating the optimal transformation.

7.1.1.2 Segmentation-Based Cost Functions

Segmentation or model-based methods incorporate some kind of geometric model for each image. Chamfer matching [Borgefors88] relies on taking a distance transform of a binary image describing edge locations. A distance transform assigns to each pixel in an image a measure of the distance to the nearest edge pixel. A template curve is matched to the distance-transform image by minimizing the root-mean-square distance value for the pixels overlapped by the curve. This method relies on having a model (the template curve) and a way of segmenting edges in the image. When the template curves are constructed by segmenting two images, then the method can be used to register the two images by registering their curves.

Fritsch et al applied chamfer matching to align images taken just before radiation treatment with previously acquired images used for planning the treatment [Fritsch95]. The alignment in this case served to ensure that radiation was delivered to the appropriate part of the body. Instead of identifying edges of objects and aligning with a template using existing methods, a core or medial representation of the object of interest was constructed which offered the advantage of robustness in the presence of noise and also served to emphasize the structures in the image that were most useful for alignment.

Lorenzen et al convert the images to be registered into a different sort of model before registering them [Lorenzen04]. In their work, images of the brain are registered by using a model that describes the probabilities that a pixel in an image belongs to a particular class of tissue. Each image, which may consist of multiple channels, is first converted into the probability model, which is essentially an image with probabilities of each tissue class at each pixel. Next, the images are registered using a cost function that operates on the probabilities.

7.1.1.3 Intensity-Based Cost Functions

Intensity-based methods use the grayscale values in each image to determine the transformation. One approach that is suited to isolated objects considers the moments of the intensity in each image. For example, one can identify a common point in each image by taking the "center of mass" of the intensity in each image. A more general approach tries to use all the image information. Methods that attempt to use the full image content include cross-correlation and maximization of mutual information.

Cross-correlation is well suited to registration of images where the relationship between corresponding intensities in the two images is known to be linear. Maximization of mutual information is appropriate when the relationship between intensities is not easily characterized. Mutual information is a measure of how well intensities in one image can be used to predict corresponding intensities in another image. Methods based on mutual information do not assume a particular relationship between intensities but do assume that there is some systematic relationship (the intensities in one image are in some sense predictable from the intensities in the other image).

In an intensity-based registration method the objective function for a transformation *T* between two images *A* and *B* is based on two sets of corresponding intensities $\{a_i\}$ in image *A* and $\{b_i\}$ in image *B* where if $a_i = A(x, y)$, then $b_i = B(x', y')$ where $(x', y') = T(x, y)$ is the transformed location from image *A* to image *B*. Given two corresponding sets of *n* pixel intensities $\{a_i\}$ in image *A* and interpolated intensities $\{b_i\}$ in image *B* one can compute any of a number of objective functions which are maximized (or minimized) to determine the optimal transformation. Here I describe two examples of objective functions: normalized correlation and mutual information.

The normalized correlation coefficient is a variation on simple correlation that includes a division by the number of matched points (*n*) to avoid biasing the transformation towards increased overlap between the two images [Studholme97]. Intuitively, the correlation measures the degree to which the intensity in image B is linearly increasing as the intensity in image *A* increases. If the intensities were proportional with a negative constant of proportionality, the correlation would be negative. If the intensities were completely independent, the correlation would be 0. The normalized correlation coefficient (the cost function) is defined by

$$\operatorname{cor}(A,B)=\frac{1}{N}\frac{1}{\sigma_A\sigma_B}\sum_{i=1..N}(a_i-\mu_A)(b_i-\mu_B)$$

where the sample means μ (measuring average intensity in each image) and standard deviations σ (measuring spread of intensity in each image) are computed as

$$\mu_A=\frac{1}{N}\sum_{i=1..N}a_i\ ,\ \mu_B=\frac{1}{N}\sum_{i=1..N}b_i\ ,\ \sigma_A=\sqrt{\frac{1}{N-1}\sum_{i=1..N}(a_i-\mu_A)^2}\ ,\ \sigma_B=\sqrt{\frac{1}{N-1}\sum_{i=1..N}(b_i-\mu_B)^2}$$

A mutual-information-based approach can be used as long as there is some systematic relationship between intensities at corresponding locations in the two images. Mutual information measures the degree of statistical dependence between two random variables regardless of the form of the relationship between them. Mutual information is defined in terms of the entropy of a random variable. Given a random variable *X*, with probability density *p*(*X*), the entropy of *X* is defined as the expected value of the log of the probability density:

$$H(X)=-E\left[\log p(X)\right]=-\int_{-\infty}^{\infty}p(X=t)\log p(X=t)\,dt$$

where $p(X=t)\log p(X=t)$ is defined as 0 if $p(X=t)=0$.

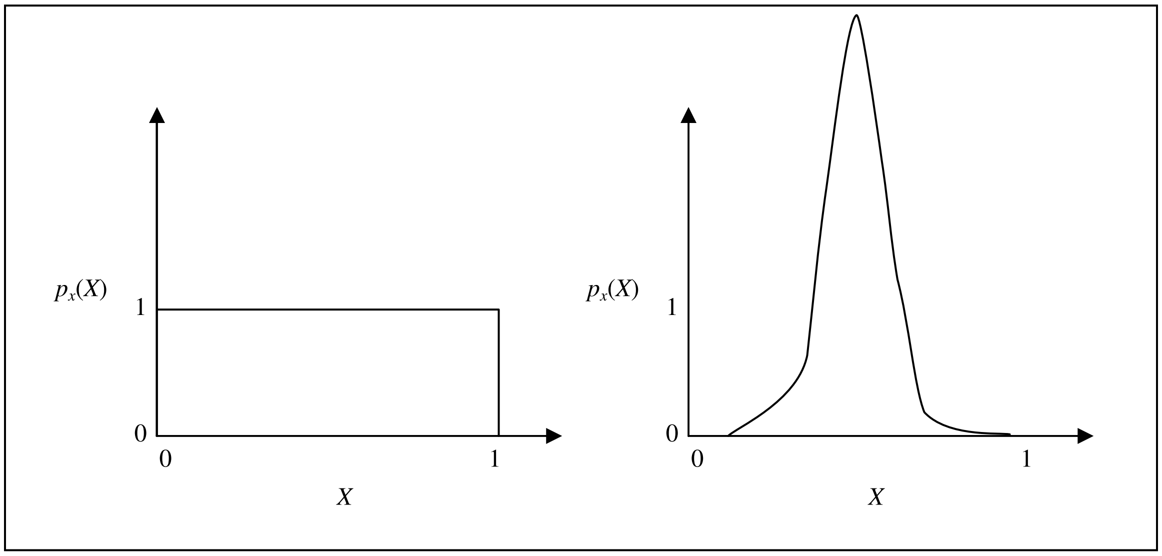


Figure 7-1: Two examples of probability density functions, one with higher entropy (left) and one with lower entropy (right)

The entropy equation turns out to provide a useful measure of the uncertainty for a random variable. Entropy ranges from plus infinity in the case of a random variable with an infinite uniform distribution (high uncertainty) to minus infinity in the case of a random variable that can only have a finite number of values (low uncertainty).

Given two random variables *X* and *Y* with joint probability density *p*(*X,Y*), the joint entropy is defined as

$$H(X,Y) = -E\left[\log p(X,Y)\right] = -\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} p(X=s,Y=t)\log p(X=s,Y=t)\,dsdt$$

The conditional entropy is defined as

$$H(X|Y) = -E\left[\log p(X|Y)\right] = -\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} p(X=s,Y=t)\log p(X=s|Y=t)\,dsdt$$

The joint entropy and conditional entropy are related by

$$H(X,Y) = H(Y) + H(X|Y)$$

The mutual information between *X* and *Y* is defined as

$$\begin{aligned} MI(X,Y) &= H(X) + H(Y) - H(X,Y) \\ &= H(X) - H(X|Y) \end{aligned}$$

Informally, the mutual information is the degree to which knowing the value of Y reduces the uncertainty in the value of X, or alternatively, knowing the value of X reduces the uncertainty in the value of Y. When two images are aligned by maximizing mutual information, the intensity in one image will best predict the intensity in the second image. A detailed description of numerical methods for computing entropy and mutual information is provided in Appendix A.

7.1.1.4 Choosing an Intensity-Based Objective Function for Aligning AFM and SEM Images

In the case of AFM and SEM images, there is not a clear relationship between intensities in the two images. AFM intensity is a measurement of surface height. For small structures, as the height increases, the intensity in the SEM can increase but higher intensity may also be correlated with steeper slopes. If one changes the height over a large area, the SEM intensity well within this region may not change at all. However, one can make use of the models for transforming a surface to SEM intensity to estimate an SEM image from the AFM data and then this AFM-based SEM image can be compared directly to the real SEM image. I use this approach (described in section 7.2.5) along with the normalized correlation coefficient as a cost function to align the simulated SEM image to the real SEM image.

7.1.2 Transformations

In order to ensure a robust and accurate registration algorithm, one should choose a type of transformation capable of representing all possible deformations to the desired accuracy with the fewest parameters. The possible deformations may be constrained by the physics of the imaging process and the physical properties of the specimen. If there is too much uncertainty in the image data, then solving for a very general transformation with many parameters may be a poorly-conditioned problem. If the transformation is less constrained than it should be given the physical system governing the transformation then one will tend to find a solution that overfits to the data (fitting to the noise) and is less accurate. In this section I review some of the previous work on AFM-AFM and SEM-SEM registration with a focus on the types of transformations that have been used with these types of images.

Romer et al developed a procedure based on cross correlation to register AFM images before a chemical treatment to AFM images after the treatment [Romer00]. They used a rigid body transformation (rotation plus translation) and accelerated the calculations by restricting the correlation calculation to three small square regions in each image. The positions of these squares were translated until the correlation was maximized. In the case where the processing changed the surface so much that it became difficult to relate the before and after images, three markers that didn't change over the course of chemical treatment were deposited on the surface using an electron beam technique. Regions centered on these markers were then used to compute the correlation.

The transformation used in [Romer00] cannot model scanner non-linearities common in scanning probe microscopes although perhaps they assume such non-linearities had been corrected by some other means. It is common for much of the non-linearity in the SPM scanner to be removed automatically using stored calibration data or reduced using closed-loop scanning hardware. For example, the Topometrix Explorer microscope uses linear displacement sensing strain gauges connected to the scanner in a closed-loop feedback system [Topometrix95]. Hadjiiski et al [Hadjiiski96] compare electron microscope and scanning tunneling microscope (STM) images of a regular grid structure to determine non-linear distortions in the STM image. These non-linearities were assumed to be fixed for a given scan range so that once found they could be used to correct distortions in new STM images. The authors consider the electron microscope image to provide accurate locations for the points in the grid. The decision to use the electron microscope image as the reference suggests that scanning non-linearities for an SEM are much less significant than those for an AFM. I adopt Hadjiski's use of the SEM image to define the regular space in which to perform alignment because the nonlinearities in the SEM image are likely to be smaller than those in the AFM image. Hadjiiski et al used three different representations for the transformation based on aligning the points in the grid: a layered back-propagation neural network with 5 neurons in the hidden layer, a 3rd order polynomial (possibly thin-plate spline) approximation, and a piecewise linear approximation. The cost function was the mean-square error or the average squared distance between corresponding points (Procrustes method).

Some non-linearities in SPM scanning are caused by thermal expansion and contraction. This distortion is not normally reproducible and cannot usually be removed by closed-loop scanning hardware or software correction. Thermal drift is expected to decrease significantly after about an hour of scanning, as the instrument gets closer to thermal equilibrium. I assume that the rate at which the rate of thermal drift changes is slow relative to the rate of image acquisition so that the drift is approximately constant for a single image. Constant drift will in general cause a shearing distortion that can be modeled by an affine transformation.

7.2 Registration Procedure

7.2.1 Summary

The SEM image intensity depends mainly on surface slope and curvature while the AFM image is most dependent on surface height. This implies that the intensity of a single pixel in the AFM image is not in general a very good predictor of the intensity of a single pixel in the SEM image. To enable a comparison of the AFM and SEM images I transform the AFM image into a simulated SEM image that serves as a proxy for the AFM image for the purpose of registration. I first erode the AFM image with the tip determined by the blind tip reconstruction method (as described in section 4.2 and 7.2.3) to estimate the actual specimen shape, and then simulate an SEM image from the eroded surface. I then align the simulated SEM image with the real SEM image.

My registration procedure consists of two steps: a manual landmark-based initialization and an automatic intensity-based refinement. I assume that the transformation from a point in the AFM image (x_{AFM}, y_{AFM}) to a corresponding point in the SEM image (x_{SEM}, y_{SEM}) is given by an affine transformation

$$\begin{bmatrix} x_{\mathrm{SEM}} \\ y_{\mathrm{SEM}} \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \end{bmatrix} \begin{bmatrix} x_{\mathrm{AFM}} \\ y_{\mathrm{AFM}} \\ 1 \end{bmatrix}$$

The manual alignment is found from a set of 3 or more pairs of corresponding landmark points in the two images that have been placed by a human. From the corresponding points, the parameters of the affine transformation are found very quickly by solving a linear system as described in 7.2.4.

For automatic alignment based on intensity, I use normalized cross-correlation of the real and simulated SEM images. The normalized cross-correlation is maximized by automatically adjusting the landmark positions (in just one of the images) that were initialized during the manual alignment step. Powell’s method [Press95] is used to optimize the landmark positions.

7.2.2 Justification for Constraining the Transformation to Two Dimensions

I assume that the projection for the SEM is orthographic and that the direction of projection for the SEM (beam axis) is parallel to the z-axis of the AFM. The orthographic assumption is reasonable because the center of projection for the SEM is on the order of a centimeter above the specimen while the field of view at the specimen is about 10 micrometers (a ratio of 1:1000). The assumption about the SEM view direction being parallel to the AFM z-axis is reasonable because of the way the samples are mounted. The specimen is a flat piece of silicon on the order of a 1 cm wide. Any variations in its tilt are due to differences in glue thickness or variations between sample holders. The AFM is usually limited to a specimen that is tilted by no more than about 5 degrees, as the *z*-piezo range is typically about one tenth the *x*-*y* piezo range. If the angle between the AFM *z*-axis and SEM beam axis were 5 degrees it would correspond to a lateral shift in the SEM image of about 8 nm for features 100 nm high relative to the substrate and a possible foreshortening of about 4 nm for features 1000 nm wide. Our SEM is limited to a resolution on the order of 10 nm so this change would be hardly noticeable. As a test we intentionally tilted one of our specimens by plus and minus 5 degrees inside the SEM to see how this would affect the SEM image. The most obvious effect of this tilt was a change in brightness at edges. We also observed the following variations in pitch measured in the vertical image direction: 514 pixels at 0 deg., 508 pixels at +5 deg, 517 pixels at -5 deg. The images are shown in Figure 7-2. The pitch is approximately 1000 nm so the variation in pitch is about plus or minus 5 pixels or 10 nm which is about the same order of magnitude as the 4 nm variation due to foreshortening that I predict and may include error due to changes in brightness at the edges. I assume that distortion of the image due to such small tilts can be represented with sufficient accuracy by a 2D affine transformation.

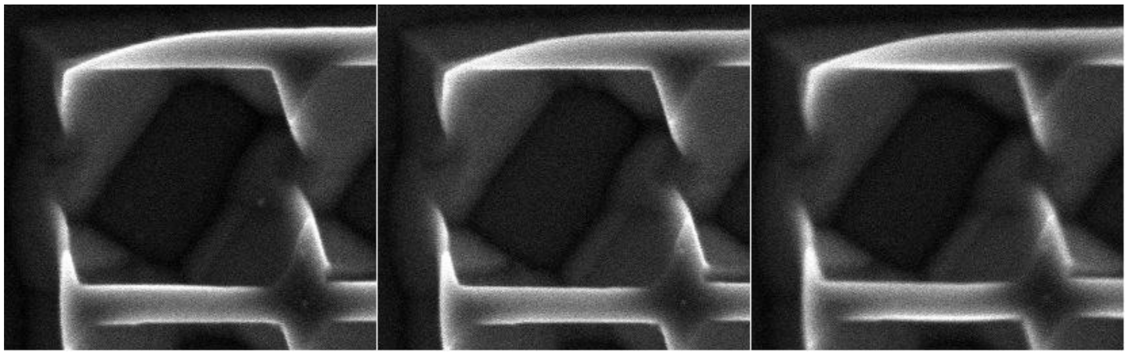


Figure 7-2: Effect of tilt on SEM image. Tilt is about a horizontal axis. Negative tilt pushes the lower part of the image farther away and brings the upper part of the image closer. The pitch of the grid is 1 micron. View at -5 degree tilt (left image) shows slight brightening of the upper edges. View at +5 degree tilt (right image) shows slight brightening of the lower edges. View at 0 tilt (middle image) does not show any obvious preferential edge brightening. The poorer focus in the right image was a result of accidental changes in the focus settings and not caused by tilt.

7.2.3 Preprocessing of the AFM Data for Alignment

One problem with using the raw AFM image for alignment is that there is a complicated distortion of the specimen that occurs due to tip dilation (described in section 2.1 and chapter 0). As the tip scans over the surface, the position on the tip that contacts the surface can vary so that while the AFM nominally samples on a regular grid in the (*x,y*) plane, the actual samples of the surface actually lie on an irregular (non-linearly distorted) grid of points on the surface in real space. The offset between the contact point and the position of a fixed reference on the tip is illustrated in the upper part of Figure 7-3.

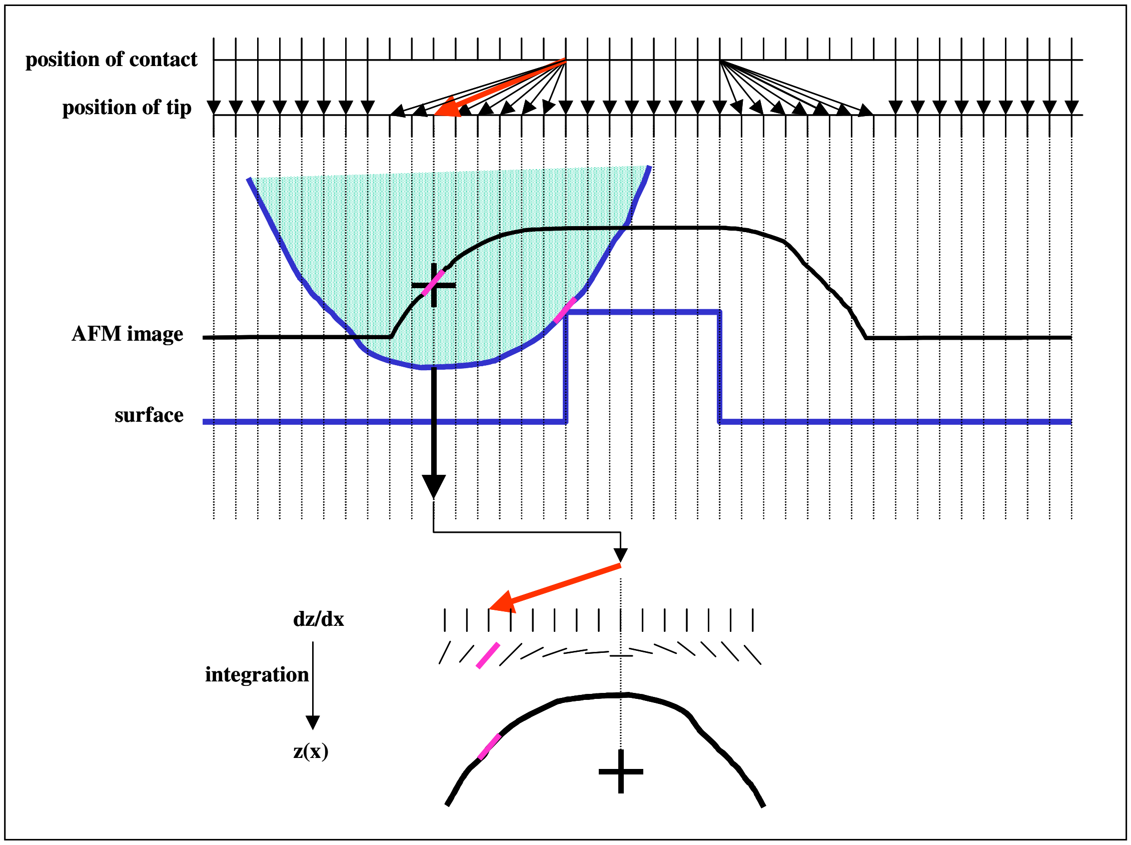


Figure 7-3: Relation between tip shape and the irregular sampling of surface points for an AFM scan. The slope in the AFM image corresponds to a point on the reflected tip (reflection is about a point) (shown at bottom) with the same slope.

I estimate the tip shape using the blind tip reconstruction algorithm and remove the non-linear component of the transformation by first eroding the AFM image with the tip shape as described in section 2.1 and then compute a simulated SEM image from the eroded AFM image.

The AFM image eroded by the true tip shape is a regular sampling (ignoring distortions other than tip dilation) of the lowest known upper bound on the true surface. Although the tip estimate is in general only an outer bound on the true tip shape, I assume that the AFM image eroded with this tip shape is a better estimate of the minimum upper bounding surface than the original AFM image is.

The SEM simulation is computed using the filter bank model as described in chapter 6. The filter bank model constructed as described in section 6.7 operates on height field images with the same physical pixel size as the SEM image data. So that the filter bank model can be applied to the eroded AFM image, the eroded AFM image is resampled such that it has the same pixel size (in physical distance units) as the SEM image. As described in section 6.7, a Monte Carlo SEM simulation is computed on the resampled AFM image to generate an SE and BSE image. The eroded AFM and simulated SEM images are then used to train the filter bank SEM simulator. Then the filter bank SEM simulator is used to generate noiseless SE and BSE images from the resampled eroded AFM image. The noiseless SE or BSE image is aligned to the real SEM image as described in sections 7.2.4 and 7.2.5. One advantage of the filter bank model over the Monte Carlo method to simulate the SEM image is that it produces no noise in the simulated image that could make the intensity-based alignment less robust.

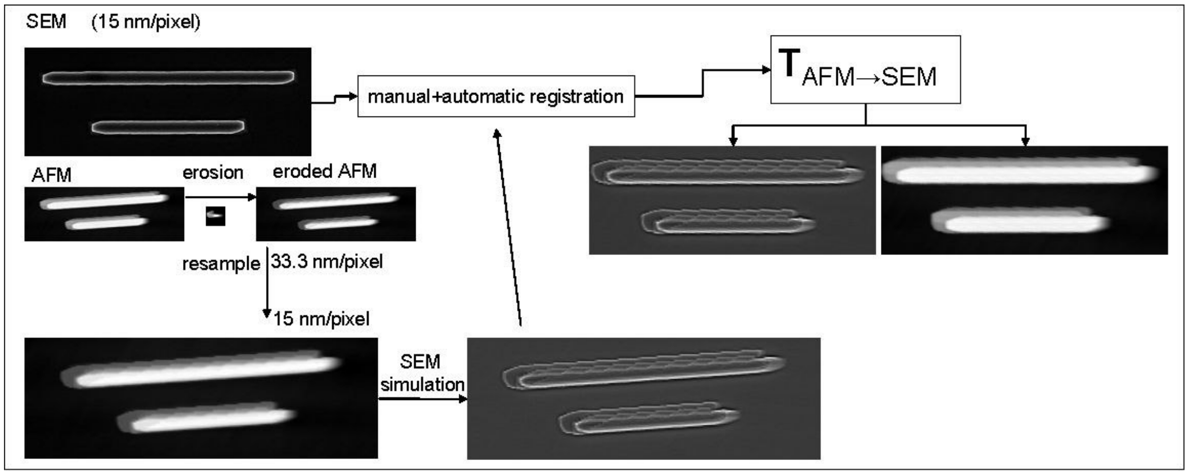


Figure 7-4: To compare AFM and SEM images, the AFM image is eroded by the tip shape estimated by blind tip reconstruction, upsampled to the resolution required by the convolution-based SEM simulator (simulator is calibrated to pixel size for SEM image), and converted to a simulated SEM image. Existing registration methods are then used to compute a transformation that aligns the simulated SEM image with the actual SEM image. The same transformation aligns the AFM image to the SEM image.

7.2.4 Landmark-Based Initial Alignment

I find an initial estimate of the transformation by having the user manually identify a set of pairs of corresponding *landmark* points in the two images (a typical example is shown in Figure 7-5). These landmark pairs are used to compute a transformation by minimizing the sum of the squared distances between points in one image and their corresponding points transformed from the other image. That is, given n points $\{\mathbf{p}_i\}$ in the SEM image and n corresponding points $\{\mathbf{q}_i\}$ in the AFM-based simulated SEM image, the desired transformation matrix ($\mathbf{T}_{S\rightarrow A}$) should minimize the objective function defined by

$$f(\mathbf{T}) = \sum_{i=1..n} \left|\mathbf{T}_{S\to A}\cdot \mathbf{p}_i - \mathbf{q}_i\right|^2 = \sum_{i=1..n} \left(\left(\mathbf{T}_{S\to A}\cdot \mathbf{p}_i\right)_x - \mathbf{q}_{ix}\right)^2 + \left(\left(\mathbf{T}_{S\to A}\cdot \mathbf{p}_i\right)_y - \mathbf{q}_{iy}\right)^2$$

Note that a point **p**=(**p**$_x$,**p**$_y$,1) refers to a position in the SEM image in units of pixels although **p**$_x$ and **p**$_y$ are not constrained to be integers and similarly for a point **q** in the AFM-based simulated SEM image.

The transformation matrix has the form

$$\mathbf{T}_{S\to A} = \begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix}$$

so multiplying this matrix by a point **p**$_i$ (in homogenous coordinates) gives us

$$\left(\mathbf{T}_{S\to A}\cdot \mathbf{p}_i\right)_x = a\mathbf{p}_{ix} + b\mathbf{p}_{iy} + c$$
$$\left(\mathbf{T}_{S\to A}\cdot \mathbf{p}_i\right)_y = d\mathbf{p}_{ix} + e\mathbf{p}_{iy} + f$$

I form two linear systems of equations:

$$\begin{bmatrix} \mathbf{p}_{0x} & \mathbf{p}_{0y} & 1 \\ \mathbf{p}_{1x} & \mathbf{p}_{1y} & 1 \\ \mathbf{p}_{2x} & \mathbf{p}_{2y} & 1 \\ & \cdots & \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} \mathbf{q}_{0x} \\ \mathbf{q}_{1x} \\ \mathbf{q}_{2x} \\ \cdots \end{bmatrix} \qquad \begin{bmatrix} \mathbf{p}_{0x} & \mathbf{p}_{0y} & 1 \\ \mathbf{p}_{1x} & \mathbf{p}_{1y} & 1 \\ \mathbf{p}_{2x} & \mathbf{p}_{2y} & 1 \\ & \cdots & \end{bmatrix} \begin{bmatrix} d \\ e \\ f \end{bmatrix} = \begin{bmatrix} \mathbf{q}_{0y} \\ \mathbf{q}_{1y} \\ \mathbf{q}_{2y} \\ \cdots \end{bmatrix}$$

Equation 7-1: System of equations to solve for T given a set of corresponding points.

If there are more than three pairs of points specified, then there will be more than three equations and the system will be over constrained. I solve this system using the least squares solver DGELS in the LAPACK library [Anderson94].

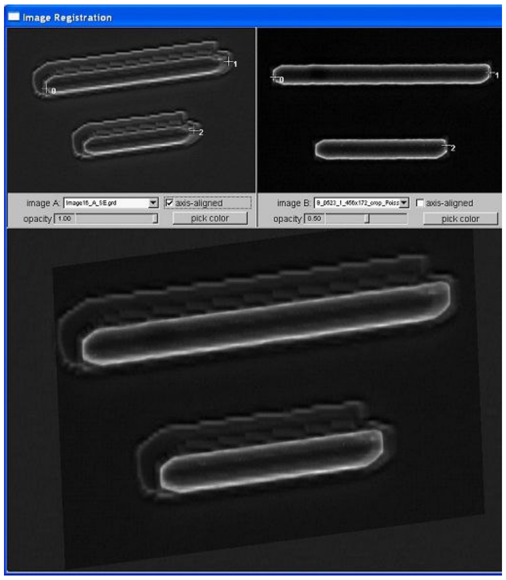


image A: Image16_A_SE.grd [✓ axis-aligned]
opacity 1.00

image B: B_0623_1_456x172_crop_Poiss axis-aligned
opacity 0.50

Figure 7-5: Screenshot of image registration user interface. Corresponding points are placed by the user in the upper two image panels and the resulting transformation is used to display the images blended together in the lower panel. The positions of the points placed by the user in the first image are the parameters for automatic alignment and the user-given positions taken as a starting point for optimization of the transformation.

This manual alignment stage finds rough approximation of the globally-optimal transformation, and is used to initialize the automatic optimization method.

7.2.5 Automatic Method for Refining the Alignment

To automatically refine the alignment I use Powell's method [Press95] to minimize the negative normalized correlation coefficient defined as

$$f\left(\mathbf{q}_0, \mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_k\right) = -\operatorname{cor}\left(A, B\right) = -\frac{1}{N}\frac{1}{\sigma_A \sigma_B}\sum_{i=1..N}\left(a_i - \mu_A\right)\left(b_i - \mu_B\right)$$

where i indexes over a set of N points that lie inside both the real and simulated SEM images given the coordinate transformation determined from the landmark points $\{\mathbf{p}_i\}$ in the real SEM image and $\{\mathbf{q}_i\}$ in the simulated SEM image. The points $\{\mathbf{p}_i\}$ are held constant while the points $\{\mathbf{q}_i\}$ are allowed to vary so the cost function $f()$ is treated as a function of the positions of the landmark points $\{\mathbf{q}_i\}$ in the simulated SEM image. k is typically 3 because this is the fewest number of points required to uniquely determine the transformation. The coordinates of these points are a parameterization of the transformation because for any coordinate values, a corresponding transformation can be found by the least squares method described in section 7.2.4. The values a_i and b_i are the intensity values for the simulated SEM and real SEM image respectively, taken from the set of pixel locations for the AFM image (or equivalently the simulated SEM image) that fall in the intersection of the two images so that

$$\begin{aligned} a_i &= S_A\left(x_i, y_i\right) \\ b_i &= S\left(\mathbf{T}_{A \to S} \cdot \left(x_i, y_i\right)\right) \end{aligned}$$

where $\mathbf{T}_{A \to S} = \mathbf{T}_{S \to A}^{-1}$.

Given a set of parameter values (the positions of the points in the simulated SEM image), *T* is computed by solving the system shown in Equation 7-1. Then, for all N pixel locations (*x*,*y*) in S_A such that $\mathbf{T}_{A \to S} \cdot \left(x, y\right)$ lies inside *S*, the image intensities at those points are inserted into the arrays *a* and *b*. The value inserted in *b*, $S\left(\mathbf{T}_{A \to S} \cdot \left(x, y\right)\right)$, requires calculation of an interpolated value in the SEM image and this is done by bilinear interpolation of the 4 nearest neighbors. The summations required to compute the mean, standard deviation and correlation are performed on sorted arrays of values in order to help reduce numerical error. Although not necessarily required, adding numbers in order from smallest to largest helps to reduce the possibility of cancellation error in floating point calculations. Ideally one would re-sort an array after each partial sum is computed.

Although I assume that the optimization is manually initialized near the globally optimal transformation, the automatic alignment could still get caught in local optima very close to the global optimum. To ensure this doesn't happen, a coarse to fine approach is used. The simulated and real SEM images are Gaussian blurred and the intensity values for the *a* and *b* arrays are taken from the blurred images. Because the simulated and real SEM images have the same pixel size in physical units, the standard deviation (σ) of the Gaussian blur kernel is set to the same value in units of pixels for both images. The alignment is optimized at each blur level in order from most to least blurred images using values for σ from {8, 5, 4, 3, 2, 1, 0} (σ=0 indicates no blurring). For σ=8, the optimization is started from the manually-placed points. At the smaller values of σ, the optimization is started from the solution found using the previous value. Examples of the blurred images for the simulated SEM image and for the actual SEM image before and after warping with *T* are shown in Figure 7-6. Plots of the objective function along one possible search direction (translation in y-direction) using these example images are shown in Figure 7-7.

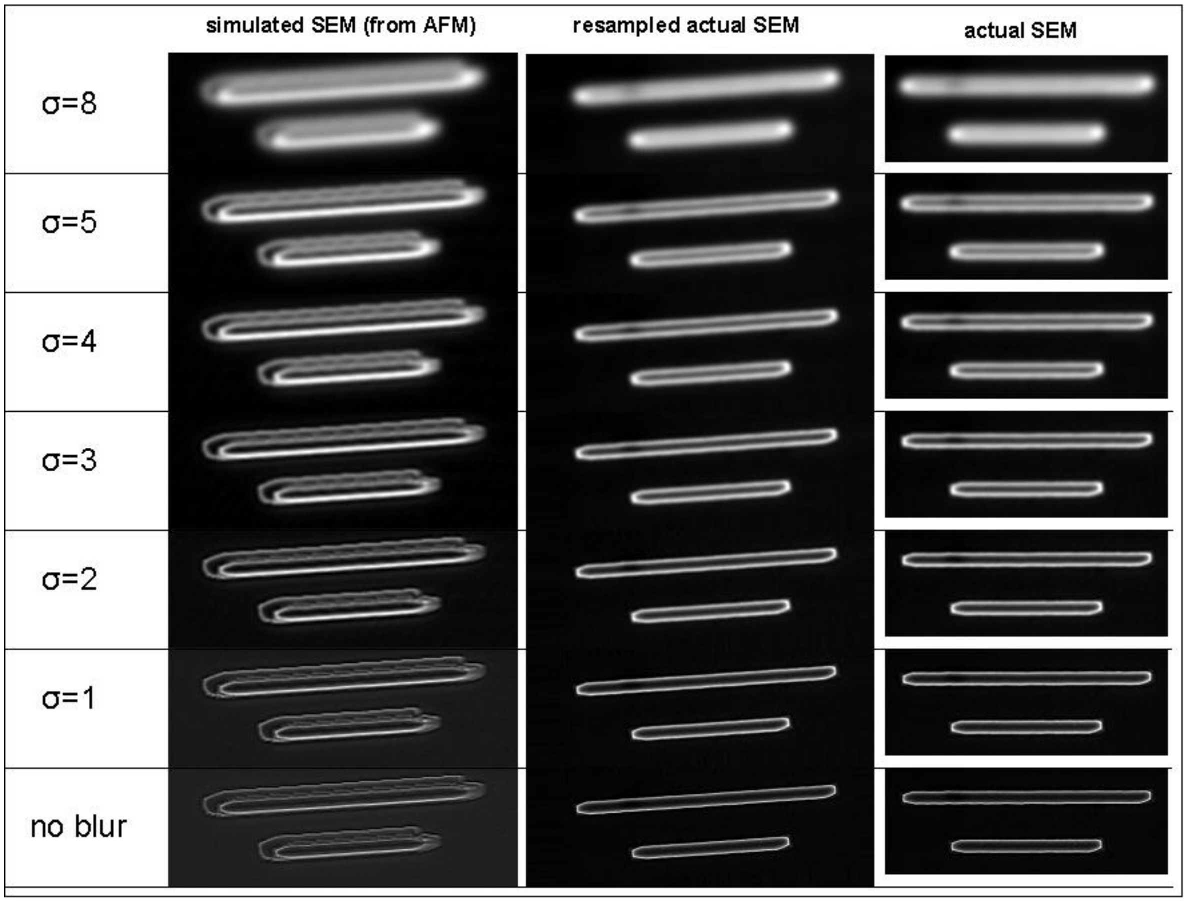


Figure 7-6: Series of blurred images used to compute the cost function for intensity-based registration. Tracking the solution from that for the coarsest scale (most blurred) down to that for the finest scale (no blurring) helps to ensure a robust alignment algorithm. Except for manual initialization at the coarsest scale, the solution for each scale level is found by local optimization starting from the solution for the previous coarser scale.

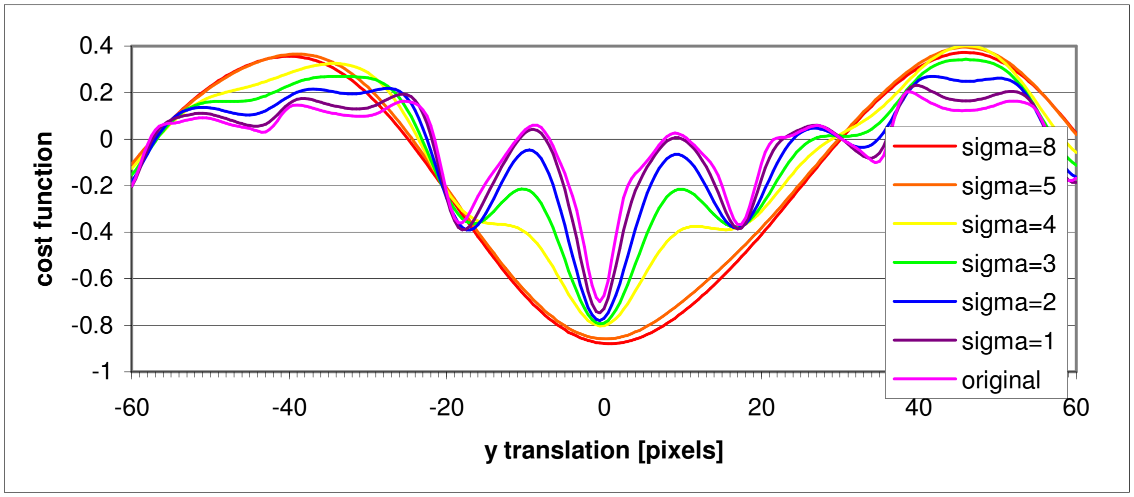


Figure 7-7: Plot of cost function at different levels of image blur for intensity-based registration. The cost function is plotted as a function of translation of the two images relative to each other in the y-direction starting from the near optimal alignment shown in Figure 7-6. The optimal translation at each blur level is located within the capture basin for the global optimum of the next smaller blur level allowing the global optimum to be tracked from coarse to fine scale.

8 Noise Models and Log-likelihood Calculation for AFM and SEM

Having an accurate model for noise is critical for accurate calculation of likelihood. In combining AFM and SEM images, the noise model for each image helps determines the relative weight each image should have in computing the surface reconstruction at each surface location. In this chapter I describe my approach to modeling noise in AFM and SEM images.

8.1 Noise in AFM Images

The noise in AFM images is typically correlated more in the fast scan direction than in the slow scan direction. The reason for this is that pixels that are acquired more closely in time are more likely to be affected in the same way by mechanical vibrations, thermal expansion or contraction, feedback artifacts, acoustic vibrations, and changes to tip geometry. This section describes how I model the correlated noise in AFM images.

8.1.1 An Autoregressive Model

I treat the noise as a signal that is added to the distortion of the surface due to dilation by the tip shape. Assuming an optimal surface reconstruction *H*, the flattened AFM image (*A*) is modeled as

$$A = H \oplus \hat{T} + E_{AFM}$$

where $\oplus\hat{T}$ represents dilation by the estimated tip shape $\hat{T}$ and E_{AFM} is the noise image. I model the noise image with a first order autoregressive model (AR1 model) that represents the noise in each pixel as the sum of some fraction of the noise the previous pixel in time and an independent noise component. The noise is assumed to have the form

$$E_{\text{AFM}}(x_t, y_t) = \alpha \cdot E_{\text{AFM}}(x_{t-1}, y_{t-1}) + v_t$$

where α is a constant and v_t is normally distributed with mean 0 and constant variance σ^2_{AFM}. The subscript *t* indexes the pixel locations (*x*,*y*) in chronological order.

In order to estimate the optimal parameters for the noise model I first acquire a calibration image of a mica surface from which an estimate of the noise image, $E_{\text{AFM}}(x_t, y_t)$, can be computed. Although the mica surface is atomically flat, the geometry of the AFM scanner can introduce tilt and bowing of the image. I compute a quadratic fit to the AFM image of the mica surface and subtract this fit from the image. The purpose of subtracting the quadratic fit is to ensure that the mean value of each pixel in the resulting image as close as possible to 0. Because later calculations to model the noise assume the mean value of each pixel is zero, deviations from this assumption would affect the noise estimate. I define $E_{\text{AFM}}(x_t, y_t)$ as the result after subtracting the quadratic fit from the AFM image of the mica surface. I assume that deviations from the quadratic fit are mainly due to errors in measuring the height of the surface because the surface is an atomically flat piece of mica. This assumption is valid because the surface is known to be inherently flat.

I do not subtract a quadratic fit from the AFM image used for surface reconstruction because the variation in height due to the quadratic component for the 10 micron wide mica scan was less than 2 nm which is less than other sources of error (such as non-linear distortions of the AFM image coordinates from a variable rate of drift) in the AFM image. The scans used as input to my algorithm are all less than 10 microns wide and use the same 100 micron scanner used to acquire the image of the mica surface.

I compute the parameters of the AR1 model (α and σ^2_{AFM}) from $E_{\text{AFM}}(x_t, y_t)$ using the Levinson-Durbin method [Jain89] as

$$\alpha = -\frac{\hat{r}(1)}{\hat{r}(0)}$$

$$\sigma^2_{\text{AFM}} = \left(1-\alpha^2\right)\hat{r}(0)$$

where $\hat{r}(\Delta t)$ is an estimate of the autocorrelation function for E_{AFM} for offset Δt:

$$\hat{r}(0)=\frac{1}{N}\sum_{t=1}^{N}E_{\mathrm{AFM}}(x_t,y_t)E_{\mathrm{AFM}}(x_t,y_t) \quad \text{and} \quad \hat{r}(1)=\frac{1}{N-1}\sum_{t=2}^{N}E_{\mathrm{AFM}}(x_t,y_t)E_{\mathrm{AFM}}(x_{t-1},y_{t-1}).$$

This choice of parameters approximately minimizes σ^2_{AFM} [Jain89].

From a 200x200 pixel image of the mica surface, I found $r(0)=0.628622\,[\mathrm{nm}^2]$, $r(1)=0.518537\,[\mathrm{nm}^2]$, $\alpha=0.825$, and $\sigma^2_{\mathrm{AFM}}=0.201\,[\mathrm{nm}^2]$. The flattened AFM image ($E_{\mathrm{AFM}}(x_t,y_t)$), corresponding prediction error ($v_t=E_{\mathrm{AFM}}(x_t,y_t)-\alpha\cdot E_{\mathrm{AFM}}(x_{t-1},y_{t-1})$) from the AR1 model with α = 0.825, and a simulation of the AFM noise using α = 0.825 and σ^2_{AFM} = 0.201 are shown in Figure 8-1.

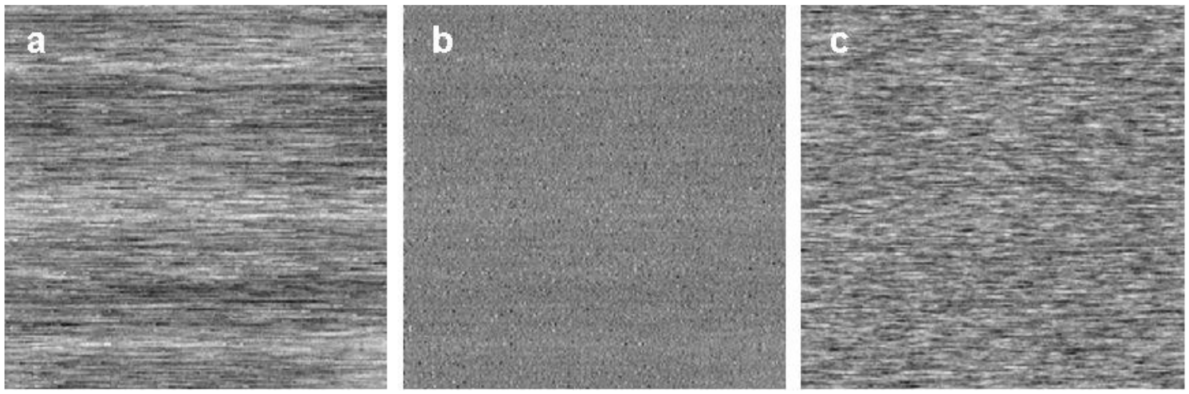


Figure 8-1: (a) Flattened AFM image, (b) whitened flattened AFM image, (c) simulation of noise using AR1 model estimated from the flattened AFM image

The degree to which the AR1 model actually captures the noise characteristics can be objectively evaluated by looking at the correlation remaining in the prediction error because the intensities in this image would ideally be zero mean, independent and identically normally distributed pixel values (based on the assumptions about v_t). If the model fits perfectly, the prediction error must have auto-covariance that is 0 everywhere except at 0. Plots of the auto-covariance function for both the flattened AFM image and the whitened flattened AFM image are shown in Figure 8-2. This shows that the whitened image has auto-covariance that is nearly 0 everywhere except when the offset in time is 0 indicating that noise each pixel can be quite accurately predicted using only the noise value at the previous pixel. Also, a plot of the distribution of intensities in the whitened image (Figure 8-3) shows that a normal distribution is a reasonable approximation.

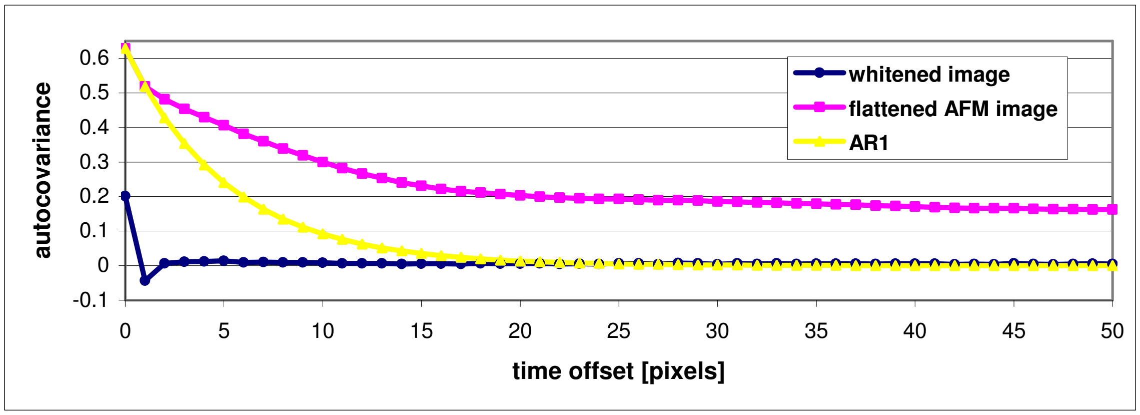


Figure 8-2: Plot of auto-covariance function for the flattened AFM image of the mica surface ($E_{\text{AFM}}(x_t, y_t)$) and whitened image (v_t). The auto-covariance function for the AR1 model is also shown for comparison with that for $E_{\text{AFM}}(x_t, y_t)$.

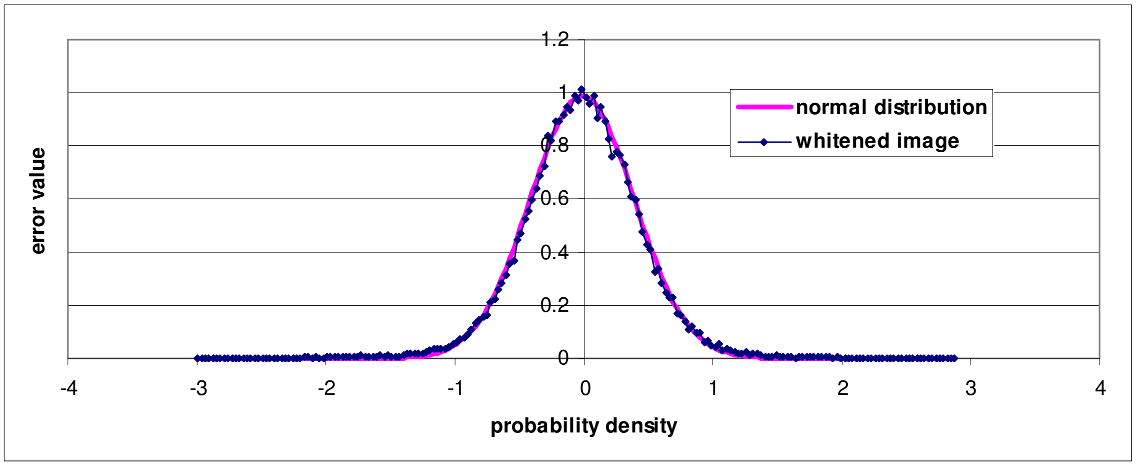


Figure 8-3: Density for the whitened image v_t shows that the normal distribution is a good fit. The normal distribution shown here has a mean of -0.01248 and standard deviation of 0.4044.

8.1.2 Calculating Log-Likelihood of an AFM image Based on the AR(1) noise model

Given a real AFM image with N pixels and an estimate for the true surface I compute a simulated AFM image from the surface estimate using gray-scale dilation as described in section 4.3. The log-likelihood for the difference between the simulated and real AFM image is computed (up to a constant offset) to determine the AFM part of the objective function. The log-likelihood tells us the probability that the difference between the simulated and real AFM images is due to noise assuming that the noise is described by the AR1 model. In this section, I derive the formula used to compute the log-likelihood for the AFM image.

The distribution of noise predicted by the AR1 model is a special case of a normal distribution. For a general normal distribution, the probability density for the error vector **e**=(e_0,e_1,e_2,…e_{N-1}) where $e_t = E_{\text{AFM}}(x_t, y_t)$ is (by definition of a normal distribution [Lee98])

$$p_{\text{error}}(\mathbf{e}) = \frac{1}{\sqrt{(2\pi)^{N_{AFM}} |\mathbf{R}|}} e^{-\frac{1}{2}\mathbf{e}^t \mathbf{R}^{-1} \mathbf{e}}$$

Equation 8-1: Normal density for errors in fitting to the AFM image

where **R** is the NxN covariance matrix for the errors. **R**={ $r_{ij} = r(|i-j|) = \text{cov}(e_i, e_j)$ }. The elements of the covariance matrix are $r_{ij} = r(|i-j|) = \alpha^{|i-j|} r(0)$ which can be shown by induction using the recursive relation:

$$\begin{aligned} r(n) &= \text{cov}(e_i, e_{i-n}) = \text{cov}(e_{i-n}, e_i) \\ &= \text{cov}(\alpha \cdot e_{i-1} + v_i, e_{i-n}) \\ &= \text{cov}(\alpha \cdot e_{i-1}, e_{i-n}) + \text{cov}(v_i, e_{i-n}) \\ &= \alpha \cdot r(n-1) + 0 \\ &= \alpha \cdot r(n-1) \end{aligned}$$

This means that for the AR1 error model, the covariance matrix is

$$\mathbf{R} = r(0) \begin{bmatrix} 1 & \alpha & \alpha^2 & \alpha^3 & & \alpha^{N-1} \\ \alpha & 1 & \alpha & \alpha^2 & & \alpha^{N-2} \\ \alpha^2 & \alpha & 1 & \alpha & \cdots & \vdots \\ \alpha^3 & \alpha^2 & \alpha & 1 & & \alpha^2 \\ & & \vdots & & \ddots & \alpha \\ \alpha^{N-1} & \alpha^{N-2} & \cdots & \alpha^2 & \alpha & 1 \end{bmatrix}$$

Gauss-Jordan elimination can be used to find the matrix inverse. In Gauss-Jordan elimination, we start from the matrix Equation 8-2

$$r(0) \left[\begin{array}{cccccc|cccccc} 1 & \alpha & \alpha^2 & \alpha^3 & & \alpha^{N-1} & 1 & 0 & 0 & 0 & & 0 \\ \alpha & 1 & \alpha & \alpha^2 & & \alpha^{N-2} & 0 & 1 & 0 & 0 & & 0 \\ \alpha^2 & \alpha & 1 & \alpha & \cdots & \vdots & 0 & 0 & 1 & 0 & \cdots & \vdots \\ \alpha^3 & \alpha^2 & \alpha & 1 & & \alpha^2 & 0 & 0 & 0 & 1 & & 0 \\ & & \vdots & & \ddots & \alpha & & & \vdots & & \ddots & 0 \\ \alpha^{N-1} & \alpha^{N-2} & \cdots & \alpha^2 & \alpha & 1 & 0 & 0 & \cdots & 0 & 0 & 1 \end{array}\right]$$

Equation 8-2: Starting point for Gauss-Jordan elimination to invert the covariance matrix.

and apply a series of row combination operations to get a matrix of the form

$$\left[\begin{array}{cccccc|cccccc}
1 & 0 & 0 & 0 & & 0 & b_{0,0} & b_{0,1} & b_{0,2} & b_{0,3} & & b_{0,N-1} \\
0 & 1 & 0 & 0 & & 0 & b_{1,0} & b_{1,1} & b_{1,2} & b_{1,3} & & b_{1,N-1} \\
0 & 0 & 1 & 0 & \cdots & \vdots & b_{2,0} & b_{2,1} & b_{2,2} & b_{2,3} & \cdots & \vdots \\
0 & 0 & 0 & 1 & & 0 & b_{3,0} & b_{3,1} & b_{3,2} & b_{3,3} & & b_{N-3,N-1} \\
 & & \vdots & & \ddots & 0 & & & \vdots & & \ddots & b_{N-2,N-1} \\
0 & 0 & \cdots & 0 & 0 & 1 & b_{1,0} & b_{1,1} & \cdots & b_{N-1,N-3} & b_{N-1,N-2} & b_{N-1,N-1}
\end{array}\right]$$

where the matrix

$$\begin{bmatrix}
b_{0,0} & b_{0,1} & b_{0,2} & b_{0,3} & & b_{0,N-1} \\
b_{1,0} & b_{1,1} & b_{1,2} & b_{1,3} & & b_{1,N-1} \\
b_{2,0} & b_{2,1} & b_{2,2} & b_{2,3} & \cdots & \vdots \\
b_{3,0} & b_{3,1} & b_{3,2} & b_{3,3} & & b_{N-3,N-1} \\
 & & \vdots & & \ddots & b_{N-2,N-1} \\
b_{1,0} & b_{1,1} & \cdots & b_{N-1,N-3} & b_{N-1,N-2} & b_{N-1,N-1}
\end{bmatrix}$$

is the inverse of **R.**

Subtracting α times the second row from the first row and subtracting α times the second to last row from the last row we get

$$r(0)\left[\begin{array}{cccccc|cccccc}
1-\alpha^2 & 0 & 0 & 0 & & 0 & 1 & -\alpha & 0 & 0 & & 0 \\
\alpha & 1 & \alpha & \alpha^2 & & \alpha^{N-2} & 0 & 1 & 0 & 0 & & 0 \\
\alpha^2 & \alpha & 1 & \alpha & \cdots & \vdots & 0 & 0 & 1 & 0 & \cdots & \vdots \\
\alpha^3 & \alpha^2 & \alpha & 1 & & \alpha^2 & 0 & 0 & 0 & 1 & & 0 \\
 & & \vdots & & \ddots & \alpha & & & \vdots & & \ddots & 0 \\
0 & 0 & \cdots & 0 & 0 & 1-\alpha^2 & 0 & 0 & \cdots & 0 & -\alpha & 1
\end{array}\right]$$

For each row from the second row to the second to last, we subtract from each row α times the subsequent row in Equation 8-2 to get

$$r(0)\left[\begin{array}{cccccc|cccccc}
1-\alpha^2 & 0 & 0 & 0 & & 0 & 1 & -\alpha & 0 & 0 & & 0 \\
\alpha-\alpha^3 & 1-\alpha^2 & 0 & 0 & & 0 & 0 & 1 & -\alpha & 0 & & 0 \\
\alpha^2-\alpha^4 & \alpha-\alpha^3 & 1-\alpha^2 & 0 & \cdots & \vdots & 0 & 0 & 1 & -\alpha & \cdots & \vdots \\
\alpha^3-\alpha^5 & \alpha^2-\alpha^4 & \alpha-\alpha^3 & 1-\alpha^2 & & 0 & 0 & 0 & 0 & 1 & & 0 \\
 & & \vdots & & \ddots & 0 & & & \vdots & & \ddots & -\alpha \\
0 & 0 & \cdots & 0 & 0 & 1-\alpha^2 & 0 & 0 & \cdots & 0 & -\alpha & 1
\end{array}\right]$$

Next, subtracting α times the previous row from each of the middle rows we get

$$r(0)\left[\begin{array}{cccccc|cccccc} 1-\alpha^2 & 0 & 0 & 0 & & 0 & 1 & -\alpha & 0 & 0 & & 0 \\ 0 & 1-\alpha^2 & 0 & 0 & & 0 & -\alpha & 1+\alpha^2 & -\alpha & 0 & & 0 \\ 0 & 0 & 1-\alpha^2 & 0 & \cdots & \vdots & 0 & -\alpha & 1+\alpha^2 & -\alpha & \cdots & \vdots \\ 0 & 0 & 0 & 1-\alpha^2 & & 0 & 0 & 0 & -\alpha & 1+\alpha^2 & & 0 \\ & & \vdots & & \ddots & 0 & & & \vdots & & \ddots & -a \\ 0 & 0 & \cdots & 0 & 0 & 1-\alpha^2 & 0 & 0 & \cdots & 0 & -\alpha & 1 \end{array}\right]$$

Dividing each row by the variance of the noise prediction error ($\sigma_{AFM}^2 = \operatorname{var}(v_t) = r(0)\left(1-\alpha^2\right)$) we get the inverse as

$$\mathbf{R}^{-1} = \frac{1}{\sigma_{AFM}^2}\begin{bmatrix} 1 & -\alpha & 0 & & & \\ -\alpha & 1+\alpha^2 & -\alpha & & 0 & \\ 0 & -\alpha & 1+\alpha^2 & \ddots & & \\ & & \ddots & \ddots & -\alpha & 0 \\ & 0 & & -\alpha & 1+\alpha^2 & -\alpha \\ & & & 0 & -\alpha & 1 \end{bmatrix}$$

The log-likelihood of the error (taking the log of Equation 8-1) is

$$\log p_{\text{error}}(\mathbf{e}) = -\frac{N}{2}\log(2\pi) - \frac{1}{2}\log|\mathbf{R}| - \frac{1}{2}\mathbf{e}^t\mathbf{R}^{-1}\mathbf{e}$$

From the form of R^{-1} the 3rd term can be calculated as

$$\mathbf{e}^t\mathbf{R}^{-1}\mathbf{e} = \frac{1}{\sigma_{AFM}^2}\begin{bmatrix} e_0 \\ e_1 \\ e_2 \\ \vdots \\ e_{N-3} \\ e_{N-2} \\ e_{N-1} \end{bmatrix}^t \cdot \begin{bmatrix} e_0 - \alpha e_1 \\ -\alpha e_0 + \left(1+\alpha^2\right)e_1 - \alpha e_2 \\ -\alpha e_1 + \left(1+\alpha^2\right)e_2 - \alpha e_3 \\ \vdots \\ -\alpha e_{N-4} + \left(1+\alpha^2\right)e_{N-3} - \alpha e_{N-2} \\ -\alpha e_{N-3} + \left(1+\alpha^2\right)e_{N-2} - \alpha e_{N-1} \\ -\alpha e_{N-2} + e_{N-1} \end{bmatrix}$$

$$= \frac{1}{\sigma_{AFM}^2}\left[\begin{array}{l} \left(e_0^2 - \alpha e_0 e_1\right) + \sum_{t=1..N-2}\left[-\alpha e_t e_{t-1} + \left(1+\alpha^2\right)e_t^2 - \alpha e_t e_{t+1}\right] + \\ \left(e_{N-1}^2 - \alpha e_{N-2}e_{N-1}\right) \end{array}\right]$$

This expression can be simplified by changing the expression inside the summation to be a perfect square and adding additional terms that balance the change to the summation to get

$$\begin{aligned}
\mathbf{e}^t\mathbf{R}^{-1}\mathbf{e} &= \frac{1}{\sigma_{AFM}^2}\left[\begin{array}{l}\left(e_0^2-\alpha e_0 e_1\right)+\sum\limits_{t=1..N-1}\left[e_t^2-2\alpha e_t e_{t-1}+\alpha^2 e_{t-1}^2\right]+\\ \left(-e_{N-1}^2-\alpha^2 e_0^2+\alpha e_{N-1}e_{N-2}+\alpha e_1 e_0\right)+\left(e_{N-1}^2-\alpha e_{N-2}e_{N-1}\right)\end{array}\right]\\
&= \frac{1}{\sigma_{AFM}^2}\left[\begin{array}{l}\left(e_0^2-\cancel{\alpha e_0 e_1}\right)+\sum\limits_{t=1..N-1}\left[e_t^2-2\alpha e_t e_{t-1}+\alpha^2 e_{t-1}^2\right]+\\ \left(-\cancel{e_{N-1}^2}-\alpha^2 e_0^2+\cancel{\alpha e_{N-1}e_{N-2}}+\cancel{\alpha e_1 e_0}\right)+\left(\cancel{e_{N-1}^2}-\cancel{\alpha e_{N-2}e_{N-1}}\right)\end{array}\right]\\
&= \frac{e_0^2\left(1-\alpha^2\right)+\sum\limits_{t=1..N-1}\left(e_t-\alpha e_{t-1}\right)^2}{\sigma_{AFM}^2}
\end{aligned}$$

The first and second terms are constants and are not needed because we are only concerned with how the objective function changes with changes in the surface:

$$\log p_{\text{error}}\left(\mathbf{e}\right)=\text{constant}-\frac{e_0^2\left(1-\alpha^2\right)+\sum\limits_{t=1..N-1}\left(e_t-\alpha e_{t-1}\right)^2}{2\sigma_{AFM}^2}$$

The part of this expression that depends on the errors ($e_0..e_{N-1}$) is used as the AFM part of the AFM-SEM objective function that determines the optimal surface reconstruction.

In estimating the AR1 model parameters (and later in applying this model to estimate image likelihood) I ignore the fact that there is typically a longer time between the last pixel on a scanline and the first pixel on the next scanline than there is between pixels on the same scanline. A better choice would be to account for the large gap in time between scanlines or model the noise in each line independently but I leave this for future work. My model will tend to underestimate the likelihood of an image because the pixel at the beginning of each line will probably have a larger change in error from the last pixel on the previous line than my model predicts. This will cause the reconstructed surface to tend to fit to the noise more closely near the left edge of the AFM image. The effect on the rest of the reconstruction should be insignificant because in the autoregressive model the correlation between the error at any pixel and the error for a pixel at the left edge falls off very quickly in a decreasing geometric sequence with distance from the edge.

8.2 Noise in SEM Images

8.2.1 Noise Sources and Distribution Models

Each pixel in an SEM image is assumed to have independent error due to noise. This is because the noise is generated by random processes that occur within the SEM during the time of acquisition for a single pixel and no memory of these processes is carried over into the time of acquisition for the next pixel. However, this assumption breaks down if there are significant noise sources external to the SEM such as mechanical/acoustic noise or electromagnetic interference.

An incident electron typically either escapes with a significant fraction of the incident energy and is counted as a backscattered electron or loses all of its energy before escaping. The incident electrons strike the surface so sparsely in time for typical beam currents and accelerating voltages that it is safe to assume that they scatter independently in the specimen. I follow [Cohen00] in modeling the distribution for the number of detected backscattered electrons as a Poisson distribution.

The secondary electrons are produced by a more complicated process. The detected secondary electrons depend to a large extent on the trajectories of the backscattered electrons because secondary electrons are generated by escaping backscattered electrons. Each backscattered electron may be responsible for generating multiple secondary electrons and each of these secondary electrons can also generate additional secondary electrons. I follow [Cohen00] in modeling the distribution for the number of secondary electrons as a log-normal distribution. From the definition of a log-normal distribution this means that the log of the number of secondary electron has a normal distribution.

Experimental evidence that the BSE signal has a Poisson distribution while the SE signal has a log-normal distribution is provided by [Cohen00]. Cohen et al used a quantile-quantile plot test to test the match between the distribution model and experimental data. Given a distribution for a variable, a quantile is the expected fraction of samples of that variable that fall below a given value. Given two distributions, two corresponding quantiles are generated for each of a set of values. These quantiles are plotted on a 2d graph and if the two distributions are of the same form, the graph will be linear. Deviations of the quantile-quantile plot from a line reflect differences in the two distributions. The experimental SE distribution matched much better to a lognormal distribution than it did to a Poisson distribution.

8.2.2 Comparison of Theoretical Noise Distributions with Empirical Distributions from Monte Carlo Simulation and Real SEM Images

One way to test the SEM noise model is to compare it to Monte Carlo simulations of electron scattering. I performed such a test, simulating electrons hitting a single point on a surface derived from AFM data shown in Figure 8-4.

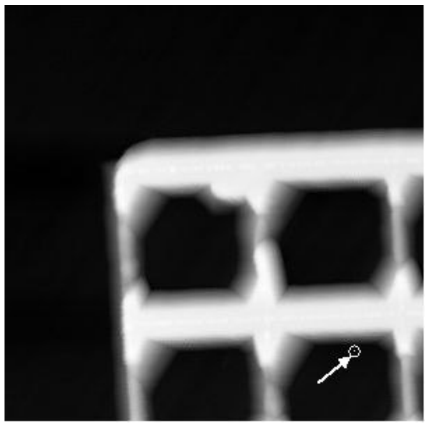


Figure 8-4: Test surface for generating test data on the distribution of measured electrons. The surface is shown as a grayscale height map. The circled point indicated by the arrow is the point at which electrons entered the surface. The range of heights is approximately 150 nm and the image is 3000 nm wide.

Three different sets of 10000 trials were performed using 1, 20, and 100 electrons per trial. All simulations used incident electrons with energy of 1keV and assumed a surface composed of silicon. For each trial, the number of secondary electrons and backscattered electrons were recorded. The results are plotted in Figure 8-5, Figure 8-6 and Figure 8-7 along with the best fit Poisson distribution (BSE and SE) and best fit log-normal distribution (SE only). These results showed that the simulator produces a BSE distribution that is closely fit by a Poisson distribution and a SE distribution that resembles more a log-normal distribution than a Poisson distribution. Also, as expected, as the number of incident electrons increases, the BSE and SE distributions become more like normal distributions.

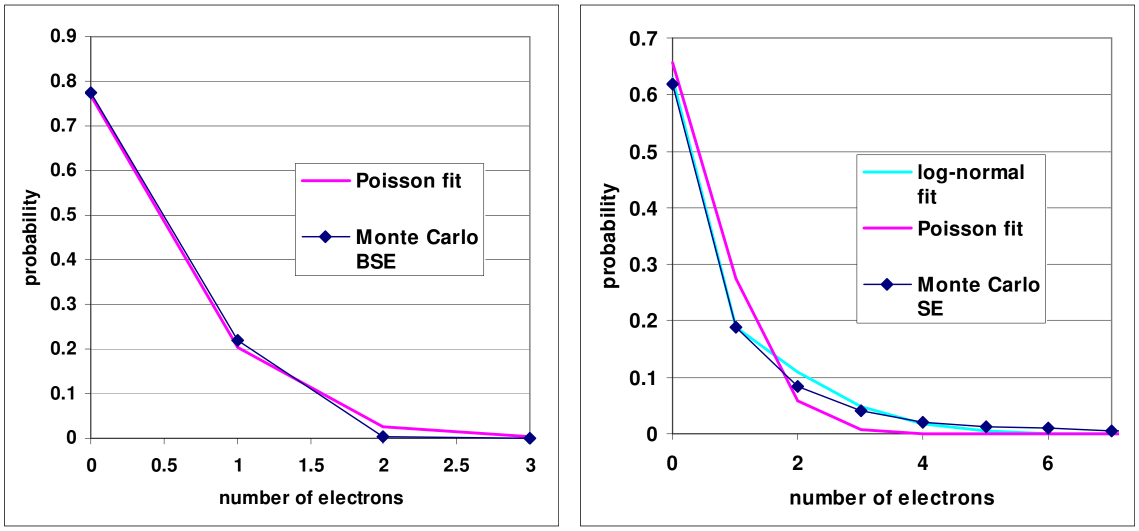


Figure 8-5: Estimated distribution for the number of BSE and the number of SE generated by a single incident electron with 1keV in silicon as simulated by the MONSEL simulator. The log-normal distribution is modified so that negative values are clamped to 0.

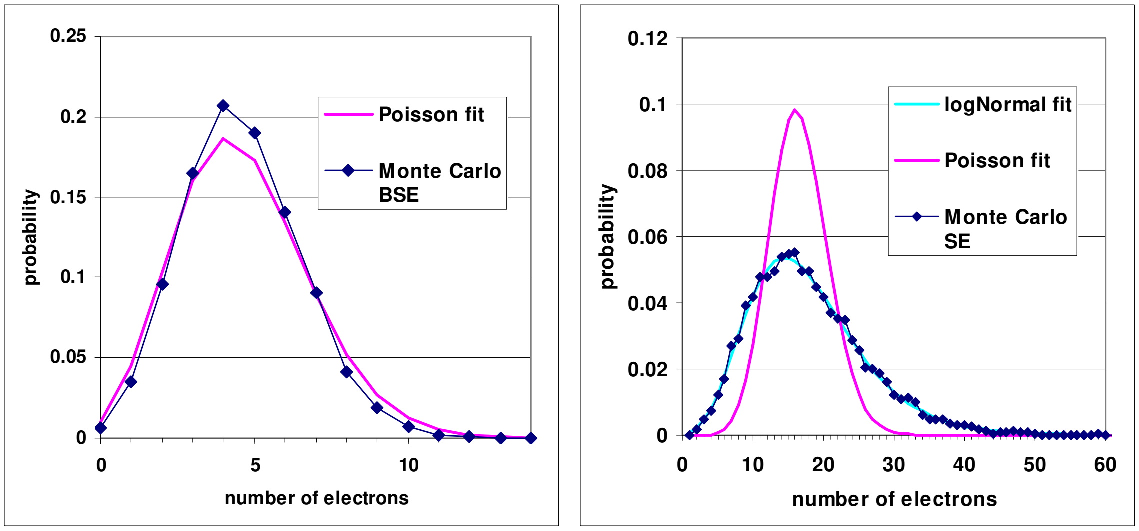


Figure 8-6: Estimated distribution for the number of BSE and the number of SE generated by 20 incident electrons with 1keV in silicon as simulated by the MONSEL simulator.

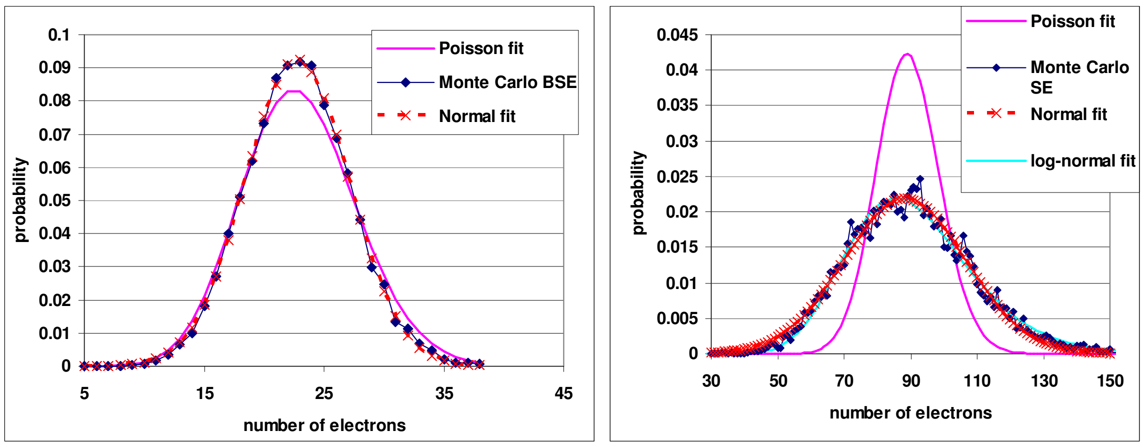


Figure 8-7: Estimated distribution for the number of BSE and the number of SE generated by 100 incident electrons with 1keV in silicon as simulated by the MONSEL simulator.

The analysis based on Monte Carlo simulations gives some insight into the origins and distribution of noise in a real SEM image but a real SEM image is typically a mixture of backscattered and secondary electrons and therefore the noise distribution may be some combination of Poisson and log-normal. Also, a real SEM image is not the raw electron counts from the detector. The raw detector signal is scaled and offset before conversion to a digital image. However, the Monte Carlo simulations, particularly for larger numbers of incident electrons (Figure 8-7) show that regardless of the mixture of BSE and SE, the signal for a particular pixel has an approximately normal distribution. Any scaling and offset applied to the signal will result in a signal that still has an approximately normal distribution, only changing the mean and standard deviation. I model the noise in real SEM images with a normal distribution. If the real distribution were either a Poisson or log-normal distribution or some combination, the variance of the normal approximation should change depending on the mean. For example, for the Poisson distribution, the variance is equal to the mean. If a scaling and offset is applied to the data, the variance will no longer be equal to the mean but there will still be a linear relationship between them. For this reason I use an affine relationship to model the dependence of the variance on the mean.

To analyze the noise in a real SEM image I first manually identified some regions in the image that had approximately constant mean intensity. Examples of such regions are shown in Figure 8-8 and Figure 8-10. Next, histograms of the intensity were compiled to get an empirical distribution for each region. The empirical distributions were fit by normal distributions (Figure 8-9 and Figure 8-11) and the mean and variance of the normal distributions were plotted per image as shown in Figure 8-9 and Figure 8-12. The results showed that the relationship between mean and variance values for each image is well-approximated by a line. This model (variance being a scaling and offset of the mean) was later used to estimate the variance in the measured values from an SEM image simulation representing the mean intensities. As I will describe in more detail in section 9.2.2, the variance was used to estimate the probability that an observed SEM image was a noisy version of the simulation and how well a specimen model matched the data.

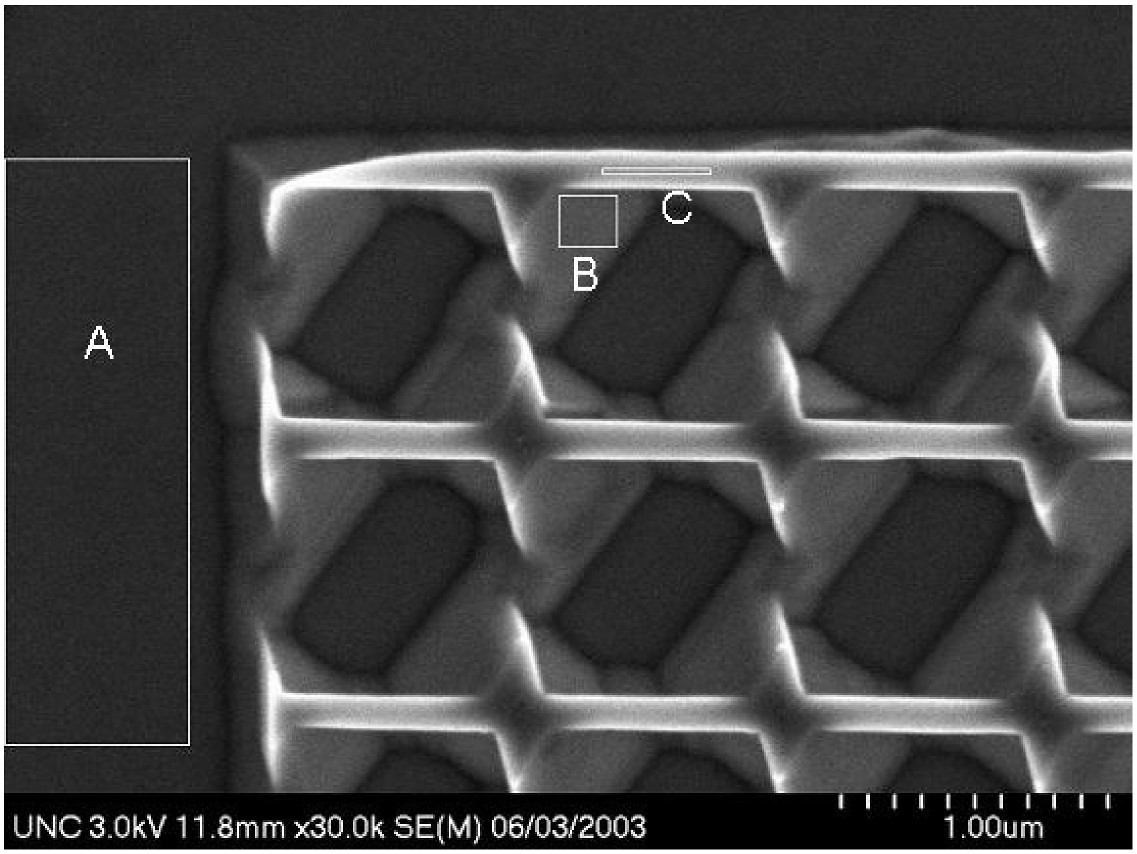


Figure 8-8: Various regions in an image were selected that had approximately constant mean intensity. The distribution of intensity within each region was used to estimate the noise distribution for the entire image.

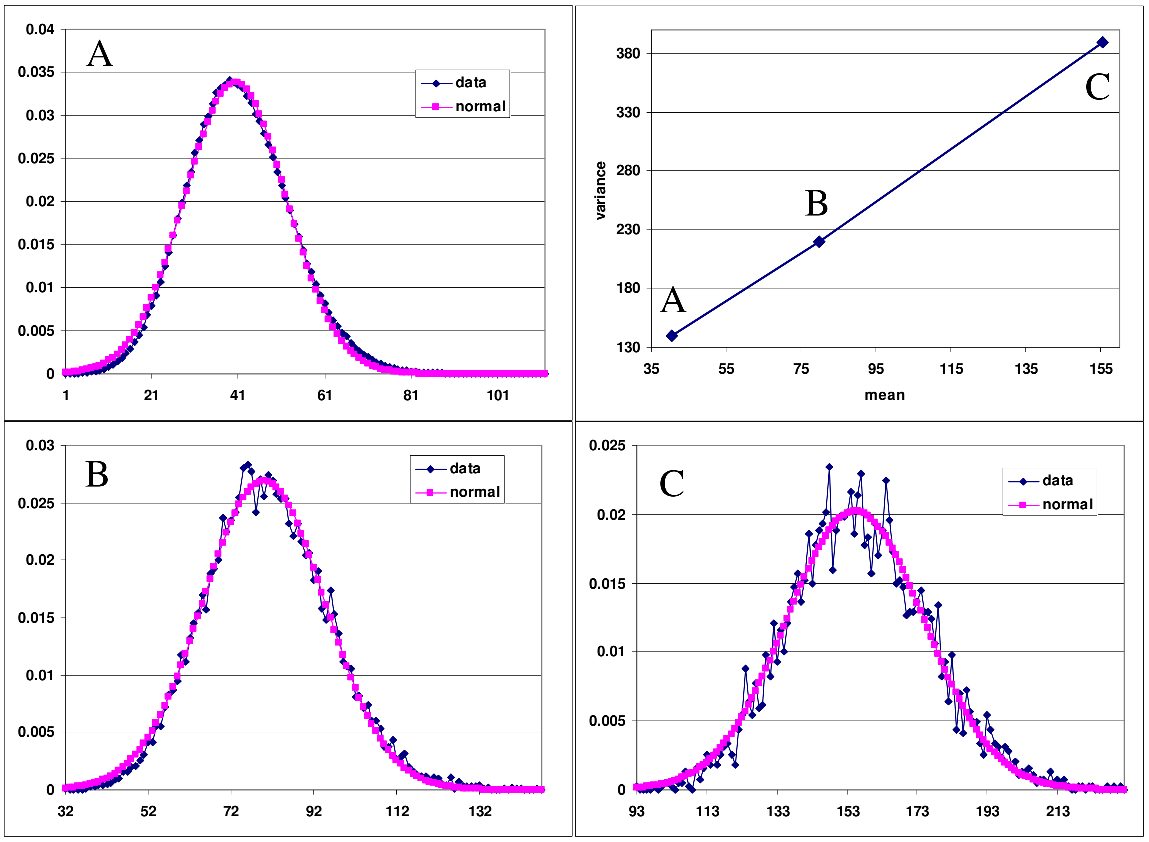


Figure 8-9: Gaussian fits to empirical intensity distributions for the three regions identified in Figure 8-8. A plot of the mean vs. variance shows a nearly linear relationship that is consistent with the linearly-transformed Poisson distribution model for noise. Units of intensity on the horizontal axis are in arbitrary digital units determined by unknown signal processing and analog to digital conversion in the SEM. The vertical axis is the probability density (fraction of pixels per unit intensity value).

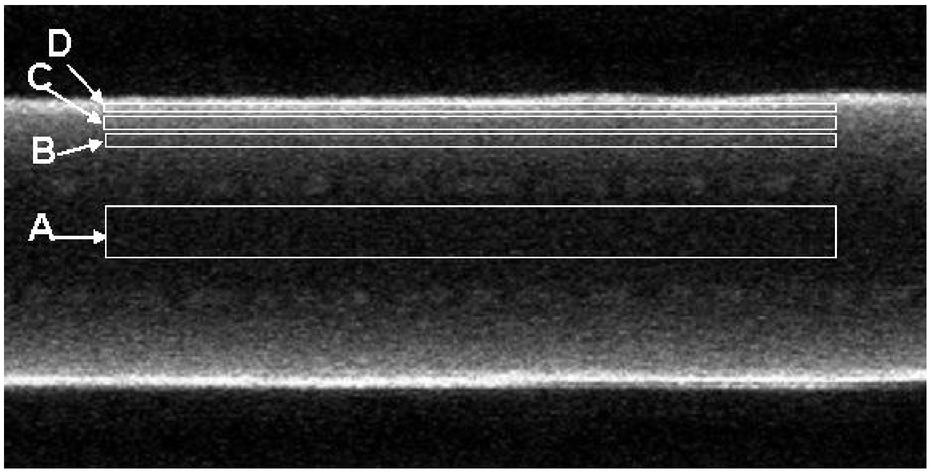


Figure 8-10: Second example of regions that were manually segmented and assumed to have constant mean intensity for the purpose of estimating the noise characteristics.

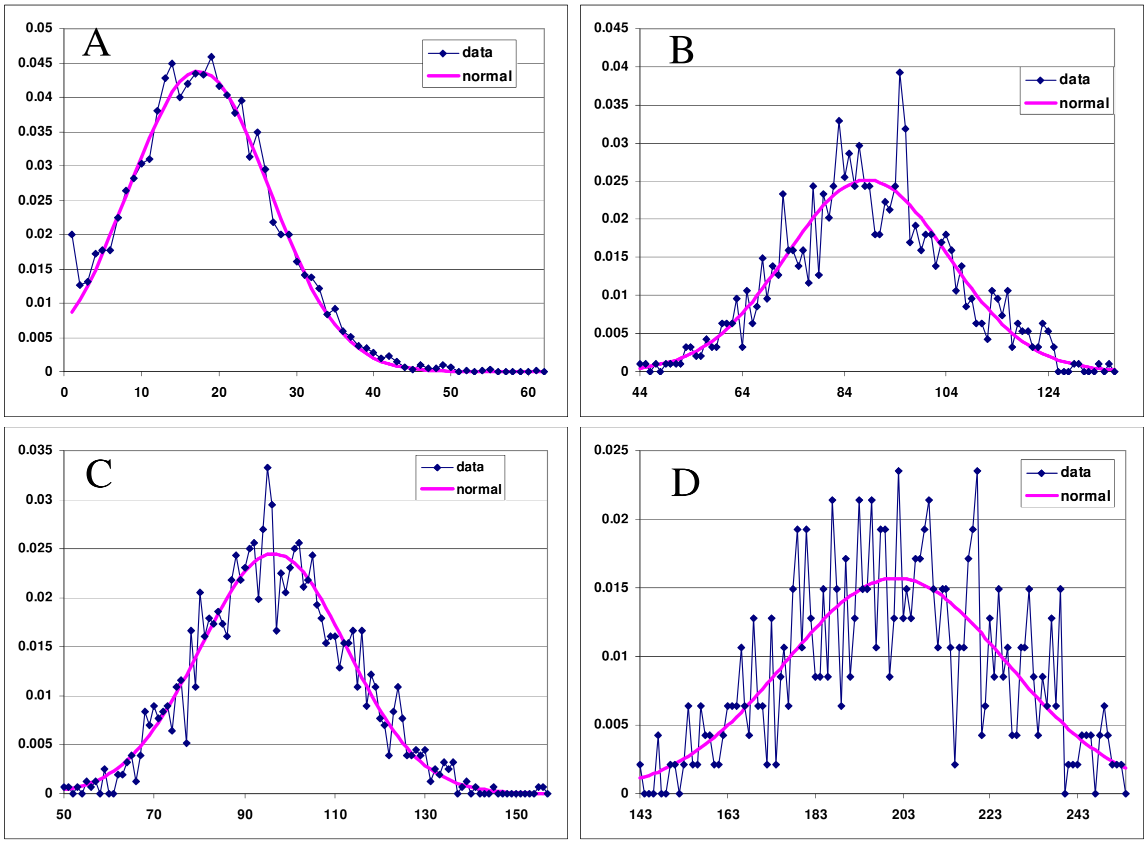


Figure 8-11: Gaussian fits to empirical intensity distributions for the three regions identified in Figure 8-10.

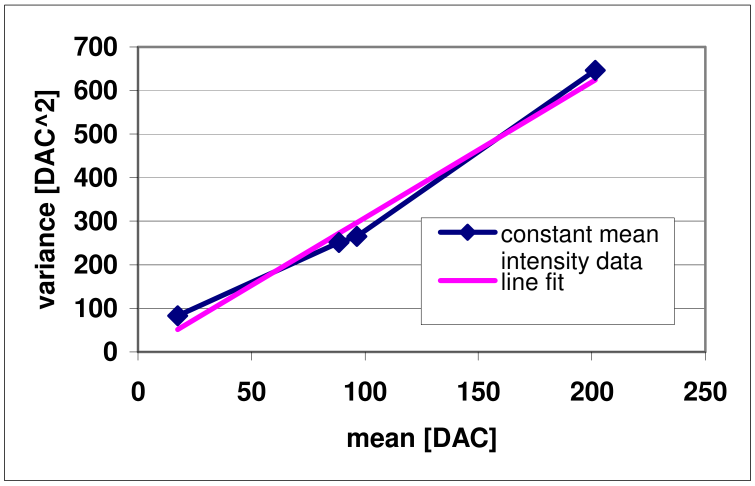


Figure 8-12: Mean and variance parameters of the Gaussian fits shown in Figure 8-11 shown in a scatter plot. I assume that the noise variance varies linearly with the mean intensity and use the line fit to these points to determine the variance from the mean intensity calculated by a simulation.

8.2.3 Calculating Log-Likelihood of an SEM Image

I compute the log-likelihood for a real SEM image, $S(x,y)$, assuming it is a noisy version of a simulated SEM image, $S_{\text{FB}}(x,y)$. $S_{\text{FB}}(x,y)$ is computed using the filter bank model described in chapter 6 from a given surface estimate *H*. The distribution for the noise at each pixel is assumed to be independent with a normal distribution: $E_{\text{SEM}}(x,y) \sim \text{N}\left(0, \sigma_{\text{FB}}^2(x,y)\right)$. The variance, $\sigma_{\text{FB}}^2(x,y)$, is computed assuming that the simulated image gives the mean value at each pixel and assuming the affine relationship between variance and mean estimated in section 8.2.2 as

$$\sigma_{\text{FB}}^2(x,y) = aS_{\text{FB}}(x,y) + b$$

The log-likelihood is computed as

$$\begin{aligned}\log p_{\text{SEM}}(S|H) &= \sum_{x=1}^{N_x}\sum_{y=1}^{N_y}\log\left(\frac{1}{\sqrt{2\pi\sigma_{\text{FB}}^2(x,y)}}e^{-\left(S(x,y)-S_{\text{FB}}(x,y)\right)^2/\left(2\sigma_{\text{FB}}^2(x,y)\right)}\right)\\ &= -\sum_{x=1}^{N_x}\sum_{y=1}^{N_y}\left[\frac{1}{2}\log\left(2\pi\sigma_{\text{FB}}^2(x,y)\right)+\frac{\left(S(x,y)-S_{\text{FB}}(x,y)\right)^2}{\left(2\sigma_{\text{FB}}^2(x,y)\right)}\right]\end{aligned}$$

9 Surface Optimization

In this chapter I describe an objective function for surface reconstruction that incorporates AFM and SEM data. I also describe how the gradient of this objective function can be efficiently computed, enabling a general gradient-based optimization method to be applied to surface reconstruction from combination AFM/SEM images.

9.1 Surface Model and Scale Space Reconstruction

Jones and Taylor introduced an approach called scale-space reconstruction in which a surface is represented by Gaussian basis functions and is constructed in a coarse to fine sequence. They also computed the SEM objective function gradient using convolution. Though the basic framework for my approach closely follows that described in [Jones94], I generalize the SEM shading function from one that depends only on slope at a point to one that depends on slope and curvature at multiple scales. I further extend the earlier work by including multiple images of different modalities (AFM and SEM). Such an optimization framework has not, to my knowledge, been used to reconstruct a surface from an AFM image by itself or to fuse an AFM image with an SEM image. The description of my algorithm closely follows the one in [Jones94] but is modified to incorporate my extensions.

The scale space representation of a surface, $H\left(x, y;\sigma\right)$, gives the height of the surface as a function of a position in the plane (*x,y*) for x=1..N_x and y=1..N_y and a scale parameter σ. The algorithm computes the surface at each of a set of discrete scales in decreasing order $\sigma_{N-1}, \sigma_{N-2}, \ldots \sigma_0$ from the largest scale to the smallest scale. At any scale σ_k ($k<N$)

the reconstruction is computed as the sum of the reconstruction at the next larger scale and an incremental update $h(x, y; \sigma_k)$ such that

$$H(x, y; \sigma_k) = \sum_{i=k..N} h(x, y; \sigma_i)$$

In the following descriptions, I abbreviate $h(x, y; \sigma_k)$ as h_k or $h_k(x, y)$ and $H(x, y; \sigma_k)$ as H_k or $H_k(x, y)$.

Building the surface up from large scale to small scale ensures that a smooth surface is constructed that explains the observed data. Although the solution found by this algorithm is not necessarily the smoothest among all surfaces that optimally explains the data, an upper bound for a lack-of-smoothness measure given in [Jones94] suggests that the solution is nearly as smooth as possible.

Each function h_k is the convolution of a coefficient image C_k with a Gaussian basis function G_k^F and is defined as

$$h_k = C_k \otimes G_k^F = \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} C_k(i, j) \cdot G_k^F(x - i, y - j)$$

$$\text{where } G_k^F(x, y) = \frac{1}{2\pi\sigma_k^2} e^{-(x^2+y^2)/(2\sigma_k^2)}$$

I define $H_k(x, y)$ in terms of the same (*x,y*) coordinates as the SEM image. For the AFM image, the surface model is sampled at the pixel coordinates for the AFM image to create a sampled height image in AFM coordinates $(x_{\text{AFM}}, y_{\text{AFM}})$ defined as

$$H_k'(x_{\text{AFM}}, y_{\text{AFM}}) = H_k(x_{\text{SEM}}, y_{\text{SEM}}) = \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} C_k(i, j) \cdot G_k^F(x_{\text{SEM}} - i, y_{\text{SEM}} - j)$$

where $(x_{\text{SEM}}, y_{\text{SEM}}) = \mathbf{T}_{A \to S}(x_{\text{AFM}}, y_{\text{AFM}})$ and $\mathbf{T}_{A \to S}$ is the transform mapping AFM coordinates to SEM coordinates as described in section 7.2.5.

In [Jones94], the sum of squared differences between the SEM intensity and the intensity predicted by the candidate surface reconstruction was used as the objective function. Assuming identical and independent Gaussian noise at each pixel in the SEM image, the surface that minimizes this objective function is equivalent to the maximum likelihood surface. In my approach, the main difference from that of [Jones94] is in the definition of a new objective function that incorporates information from both an SEM image and an AFM image. The remaining sections of this chapter describe my objective function.

9.2 Objective Function

I define the optimal surface as the one that is most likely given the observed AFM and SEM data. Applying Bayes' rule to the conditional probability of the surface reconstruction H_k given the AFM image (*A*) and SEM image (*S*) we have

$$p\left(H_k \middle| A \text{ and } S\right) = \frac{p\left(A \text{ and } S \middle| H_k\right) \cdot p\left(H_k\right)}{p\left(A \text{ and } S\right)}$$

$p\left(A \text{ and } S\right)$ is a constant given the available data and I assume that $p\left(H_k\right)$ is uniform (no surface is more probable than any other). With this assumption, $p\left(H_k \middle| A \text{ and } S\right)$ is maximized when $p\left(A \text{ and } S \middle| H_k\right)$ is maximized. Because the AFM and SEM images are independent, $p\left(A \text{ and } S \middle| H_k\right) = p_{\text{AFM}}\left(A \middle| H_k\right) p_{\text{SEM}}\left(S \middle| H_k\right)$. Taking the log of both sides we get $\log p\left(A \text{ and } S \middle| H_k\right) = \log p_{\text{AFM}}\left(A \middle| H_k\right) + \log p_{\text{SEM}}\left(S \middle| H_k\right)$ and because the log function is monotonically increasing, maximizing $\log p\left(A \text{ and } S \middle| H_k\right)$ is equivalent to maximizing $p\left(H_k \middle| A \text{ and } S\right)$. I actually minimize $-\log p\left(A \text{ and } S \middle| H_k\right)$.

For each scale σ_k the objective function that I minimize is an estimate up to a constant of the negative log-likelihood of the AFM and SEM images given the observed data. The objective function is defined as

$$f\left(H_k\right) = -\log p_{\text{AFM}}\left(A \middle| H_k\right) - \log p_{\text{SEM}}\left(S \middle| H_k\right) - constant$$

where $p_{\text{AFM}}\left(A \middle| H_k\right)$ is the probability for the observed AFM image *A* and $p_{\text{SEM}}\left(S \middle| H_k\right)$ is the probability for the observed SEM image *S* given that the true surface *H* equals the candidate reconstruction H_k. The log-likelihood is estimated up to a constant offset ($-constant$ above) that can be ignored because we are only interested in minimizing the function and not computing its actual value.

9.2.1 AFM Component

$p_{\mathrm{AFM}}\left(A \middle| H_k\right)$ is estimated assuming that $A = H_k' \oplus \hat{T} + E_{AFM}$ where $\hat{T}$ is an estimate of the AFM tip shape, $\oplus$ is discrete dilation, and E_{AFM} is a noise image described by an AR1 model. Given pixels in *A*, (x_t,y_t), where *t* indexes the pixels in chronological order of acquisition then the AR1 noise model is expressed as described in section 8.1.1 by

$$E_{\mathrm{AFM}}\left(x_t, y_t\right) = \alpha \cdot E_{\mathrm{AFM}}\left(x_{t-1}, y_{t-1}\right) + v_t$$

where α is a constant and v_t is normally distributed with mean 0 and constant variance σ^2_{AFM}. Given an AFM image *A*, an estimated surface height image in AFM coordinates, H_k', and an AFM tip image $\hat{T}$, an error image, E_{AFM}, is computed by subtracting from the AFM image the estimated surface image dilated by the tip image ($E_{\mathrm{AFM}} = A - H_k' \oplus \hat{T}$). The method for estimating the log-likelihood described in section 8.1.2 is used to calculate the AFM part of the objective function as

$$-\log p_{\mathrm{AFM}}\left(A \middle| H_k\right) = \text{constant} + \frac{e_0^2\left(1-\alpha^2\right) + \sum_{t=1}^{N_{AFM}-1}\left(e_t - \alpha e_{t-1}\right)^2}{2\sigma^2_{AFM}}$$

where $e_t = E_{\mathrm{AFM}}\left(x_t, y_t\right)$ and *t* indexes the pixels in the AFM image in chronological order.

9.2.2 SEM Component

$p_{\mathrm{SEM}}\left(S \middle| H_k\right)$ is estimated assuming that $S\left(x,y\right) = S_{\mathrm{FB}}\left(x,y\right) + E_{\mathrm{SEM}}\left(x,y\right)$ where S_{FB} is a function of H_k and represents the filter-bank-based SEM simulation described in chapter 6 and E_{SEM} is a noise image where each pixel is independent with a normal distribution: $E_{\mathrm{SEM}}\left(x,y\right) \sim \mathrm{N}\left(0, \sigma^2_{\mathrm{FB}}\left(x,y\right)\right)$ where $\sigma^2_{\mathrm{FB}}\left(x,y\right) = aS_{\mathrm{FB}}\left(x,y\right) + b$ is the line relating variance to the mean value $S_{\mathrm{FB}}\left(x,y\right)$ found using the procedure described in section 8.2.2. The SEM part of the objective function is calculated as

$$\begin{aligned}
-\log p_{\mathrm{SEM}}\left(S \middle| H_k\right) &= -\sum_{x=1}^{N_x}\sum_{y=1}^{N_y}\log\left(\frac{1}{\sqrt{2\pi\sigma^2_{\mathrm{FB}}\left(x,y\right)}}e^{-\left(S\left(x,y\right)-S_{\mathrm{FB}}\left(x,y\right)\right)^2 / \left(2\sigma^2_{\mathrm{FB}}\left(x,y\right)\right)}\right) \\
&= \sum_{x=1}^{N_x}\sum_{y=1}^{N_y}\left[\frac{1}{2}\log\left(2\pi\sigma^2_{\mathrm{FB}}\left(x,y\right)\right) + \frac{1}{2}\frac{\left(S\left(x,y\right)-S_{\mathrm{FB}}\left(x,y\right)\right)^2}{\sigma^2_{\mathrm{FB}}\left(x,y\right)}\right]
\end{aligned}$$

9.3 Objective Function Gradient

Because I use a gradient-based optimization method, it is necessary to calculate the gradient of the objective function in addition to the value of the objective function. The gradient of the objective function (∇f) is the direction in parameter space in which the objective function increases the most. It is defined as

$$(\nabla f)_{i,j} = \frac{\partial f(H_k)}{\partial C_k(i,j)} = -\frac{\partial \log p_{\text{AFM}}(A|H_k)}{\partial C_k(i,j)} - \frac{\partial \log p_{\text{SEM}}(S|H_k)}{\partial C_k(i,j)}$$

where *(i,j)* i=1..N_x, j=1..N_y index over the components of the gradient vector. Intuitively, this tells which Gaussian coefficients should go up, and which should go down (and by how much) to most quickly increase the likelihood of the surface estimate. The next two sections describe in detail how the gradient is calculated.

9.3.1 AFM Component of Log-Likelihood Gradient

Recall from section 9.2.1 that the negative log-likelihood for the AFM image given a specimen surface model is approximated by

$$-\log p_{\text{AFM}}(A|H_k) = \text{constant} + \frac{e_0^2(1-\alpha^2) + \sum_{t=1}^{N-1}(e_t - \alpha e_{t-1})^2}{2\sigma_{AFM}^2}$$

Equation 9-1

where N is the number of pixels in the AFM image, $e_t = A(x_t, y_t) - A_k(x_t, y_t)$, $A_k = H_k' \oplus \hat{T}$ (the simulated AFM image computed by dilating H_k' with the tip model $\hat{T}$) and σ_{AFM}^2 and α are parameters of the AR1 model. In this section I derive an expression for the derivative of this function with respect to each of the surface parameters. The pixels in both the AFM image (*A*) and the simulated AFM image ($A_k = H_k' \oplus \hat{T}$) are indexed in chronological order of acquisition. $A(x_t, y_t)$ represents the pixel in the AFM image at timestep *t* and $A_k(x_t, y_t)$ represents the corresponding simulated pixel. For my data, the sequence of pixels locations (x_t, y_t), *t*=0..*N*-1 is a raster scan of the image starting at (0,0).

The derivative of Equation 9-1 with respect to a surface parameter $C_k(i,j)$ (Gaussian basis function coefficient for scale k centered at pixel (i,j) in the surface estimate height image) is

$$-\frac{\partial \log p_{\mathrm{AFM}}\left(A|H_k\right)}{\partial C_k(i,j)} = \frac{1}{\sigma_{AFM}^2}\left[\left(1-\alpha^2\right)e_0\frac{\partial e_0}{\partial C_k(i,j)} + \sum_{t=1}^{N-1}\left(e_t-\alpha e_{t-1}\right)\frac{\partial\left(e_t-\alpha e_{t-1}\right)}{\partial C_k(i,j)}\right]$$
$$= \frac{1}{\sigma_{AFM}^2}\left[\left(1-\alpha^2\right)e_0\frac{\partial e_0}{\partial C_k(i,j)} + \sum_{t=1}^{N-1}\left(e_t-\alpha e_{t-1}\right)\left(\frac{\partial e_t}{\partial C_k(i,j)} - \alpha\frac{\partial e_{t-1}}{\partial C_k(i,j)}\right)\right]$$

Substituting $A(x_t,y_t)-A_k(x_t,y_t)$ for e_t and using $\frac{\partial A(x_t,y_t)}{\partial C_k(i,j)}=0$ (the AFM data doesn't change if we change the reconstructed surface) we get

$$-\frac{\partial \log p_{\mathrm{AFM}}\left(A|H_k\right)}{\partial C_k(i,j)} = \frac{1}{\sigma_{AFM}^2}\left[\begin{array}{l}\left(1-\alpha^2\right)e_0\frac{\partial\left(A(x_0,y_0)-A_k(x_0,y_0)\right)}{\partial C_k(i,j)} + \\ \sum_{t=1}^{N-1}\left(e_t-\alpha e_{t-1}\right)\left(\begin{array}{l}\frac{\partial\left(A(x_t,y_t)-A_k(x_t,y_t)\right)}{\partial C_k(i,j)} - \\ \alpha\frac{\partial\left(A(x_{t-1},y_{t-1})-A_k(x_{t-1},y_{t-1})\right)}{\partial C_k(i,j)}\end{array}\right)\end{array}\right]$$
$$= -\frac{1}{\sigma_{AFM}^2}\left[\begin{array}{l}\left(1-\alpha^2\right)e_0\frac{\partial A_k(x_0,y_0)}{\partial C_k(i,j)} + \\ \sum_{t=1}^{N-1}\left(e_t-\alpha e_{t-1}\right)\left(\frac{\partial A_k(x_t,y_t)}{\partial C_k(i,j)} - \alpha\frac{\partial A_k(x_{t-1},y_{t-1})}{\partial C_k(i,j)}\right)\end{array}\right]$$

Equation 9-2

Equation 9-2 tells us how to calculate the gradient in terms of the derivative of the dilated surface with respect to the surface parameter. Although the derivative is not necessarily continuous because of the nature of the dilation operation, it can be calculated using a modified dilation operation that keeps track of the point on the input surface that determined the height at each point in the dilated surface. In the discrete dilation operation this can be accomplished by associating with each height value in the input surface its (x,y) position. $A_k = H_k' \oplus \hat{T}$ is computed as

$$A_k(x,y) = \max_{(b_x,b_y)}\left\{H_k'\left(x-b_x, y-b_y\right) + \hat{T}\left(b_x,b_y\right)\right\}$$

I define a modified version of discrete dilation that keeps track of locations from the surface H'_k as

$$\mathbf{A}_k(x,y) = \left(A_k(x,y), x_H, y_H\right) \text{ such that}$$

$$A_k(x,y) = H'_k(x_H, y_H) + \hat{T}(x - x_H, y - y_H) = \max_{(b_x,b_y)}\left\{H'_k\left(x - b_x, y - b_y\right) + \hat{T}\left(b_x, b_y\right)\right\}$$

This means that the dilated image is augmented by an *x* and *y* channel for each pixel. Instantaneously we have that

$$A_k(x,y) = H'_k\left(x_H(x,y), y_H(x,y)\right) + \hat{T}\left(x - x_H(x,y), y - y_H(x,y)\right)$$

Equation 9-3

$\left(x_H(x,y), y_H(x,y)\right)$ can shift by a distance of multiple pixels with arbitrarily small changes in H'_k and many points on H'_k may not touch the tip and therefore won't be in $\left\{\left(x_H(x_t,y_t), y_H(x_t,y_t)\right) \middle| t = 0..N-1\right\}$. These two qualities may cause problems for local optimization methods. Those points on H'_k that don't touch the tip will not be considered for adjustment although changes to those points could lead to a reduction in the AFM error. Such a situation is illustrated in Figure 9-1.

A gradient-based optimization method may seem a poor choice given the discontinuities in the AFM gradient formula and possibility for stagnation of parts of the surface that lose contact with the simulated tip scanning but there are two factors that help to avoid these problems and enable the gradient-based approach to succeed. One is that the SEM part of the objective is better behaved and takes over where the AFM part loses influence over parts of the surface that aren't touched by the tip. Another factor is that each basis function influences the height over a region of the surface rather than at a single point and this helps to ensure that each basis function has some interaction with the tip and that jumps in the AFM part of the gradient won't be very large. This effect is most significant for optimization at the largest scales. Because the surface is reconstructed in a coarse-to-fine sequence, by the time the smaller scale structures are adjusted all the parts of the surface (relatively flat places) that the AFM tells us about are mostly determined already. The parts of the surface that do not contact the tip will be those for which there is no information from the AFM (near the bottom of cliff faces). Stagnation of parts of the surface would be a much bigger problem if one tried to just adjust the height at each pixel independently from the beginning.

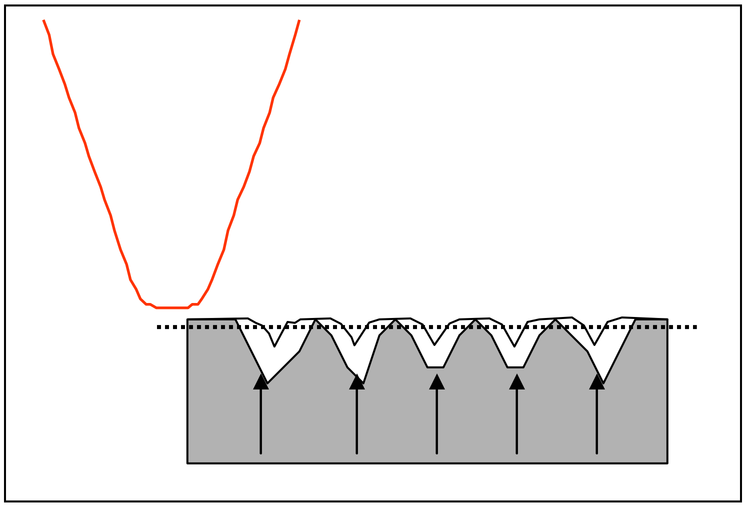


Figure 9-1: Possible problem case for local optimization of AFM component of the objective function. The gray region represents the current solution for the surface. The dotted line represents the observed AFM image, and the solid line represents the AFM image simulated by dilation with the tip shape shown in red. The parts of the surface indicated by arrows need to be raised up to help make the simulated AFM image better match the observed AFM image but because they do not touch the simulated tip as it is scanned over the surface, small changes to these parts of the surface have no effect on the simulated AFM image and local optimization may fail. For this case, we rely on SEM information to pull the valleys up.

Taking the derivative of Equation 9-3 with respect to the basis function coefficient we get

$$\frac{\partial A_k\left(x_t, y_t\right)}{\partial C_k(i,j)} = \frac{\partial H_k'\left(x_H\left(x_t, y_t\right), y_H\left(x_t, y_t\right)\right)}{\partial C_k(i,j)}$$

Recall that the surface estimate at scale σ_k in AFM coordinates is defined as

$$H_k'\left(x_{\text{AFM}}, y_{\text{AFM}}\right) = H_k\left(x_{\text{SEM}}, y_{\text{SEM}}\right) = \sum_{i=1}^{N_x}\sum_{j=1}^{N_y} C_k\left(i,j\right)\cdot G_k^F\left(x_{\text{SEM}} - i, y_{\text{SEM}} - j\right)$$

$$\text{where } \left(x_{\text{SEM}}, y_{\text{SEM}}\right) = \mathbf{T}_{A\to S}\left(x_{\text{AFM}}, y_{\text{AFM}}\right)$$

so

$$\frac{\partial H_k'\left(x_{H_k'}(x_t,y_t),y_{H_k'}(x_t,y_t)\right)}{\partial C_k(i,j)}=G_k^F\left(x_t'-i,y_t'-j\right)$$

$$=\frac{1}{2\pi\sigma_k^2}e^{-((x_t'-i)^2+(y_t'-j)^2)/2\sigma_k^2}$$

where

$$x_t'=\left(\mathbf{T}_{A\to S}\cdot\left(x_{H_k'}(x_t,y_t),y_{H_k'}(x_t,y_t)\right)\right)_x$$

$$y_t'=\left(\mathbf{T}_{A\to S}\cdot\left(x_{H_k'}(x_t,y_t),y_{H_k'}(x_t,y_t)\right)\right)_y$$

Putting all this together, the gradient of the negative log-likelihood is computed as

$$-\frac{\partial\log p_{\text{AFM}}\left(A|H_k\right)}{\partial C_k(i,j)}=-\frac{1}{\sigma_{\text{AFM}}^2}\left(1-\alpha^2\right)e_0\frac{1}{2\pi\sigma_k^2}e^{-((x_0'-i)^2+(y_0'-j)^2)/2\sigma_k^2}$$

$$-\frac{1}{\sigma_{\text{AFM}}^2}\sum_{t=1}^{N-1}\left[\left(e_t-\alpha e_{t-1}\right)\left(\begin{array}{l}\frac{1}{2\pi\sigma_k^2}e^{-((x_t'-i)^2+(y_t'-j)^2)/2\sigma_k^2}\\-\alpha\frac{1}{2\pi\sigma_k^2}e^{-((x_{t-1}'-i)^2+(y_{t-1}'-j)^2)/2\sigma_k^2}\end{array}\right)\right]$$

9.3.2 SEM Component of Log-Likelihood Gradient

This section describes an efficient method for computing the gradient of the SEM component of the objective function. This method relies on the fact that the facet model coefficients introduced in section 6.4 can be computed by a convolution as described in Appendix D.2 and [Haralick81]. I first give a mathematical derivation of a formula for the gradient. After this I describe the algorithm used to compute this formula. Throughout these descriptions I use $K_{m,n}^F$ to represent the convolution kernel used to compute a facet model coefficient at each pixel and $K_{m,n}(x,y)$ to represent the actual coefficient computed at (*x*,*y*) by the convolution. G_k^F represents the normalized 2-dimensional Gaussian kernel centered at 0 with width the same as that used for the surface basis functions at scale *k* (the superscript *F* in G_k^F is to avoid confusion with the gradient magnitude image G_n representing the gradient magnitude estimated for neighborhood size *n*).

9.3.2.1 Derivation of SEM Objective Function Gradient

The derivative of the SEM objective function (see section 9.2.2) with respect to a Gaussian basis function coefficient is

$$\frac{\partial\left(-\log p_{\text{SEM}}\left(S\middle|H_k\right)\right)}{\partial C_k(i,j)}=\frac{\partial\left(\sum_{x=1}^{N_x}\sum_{y=1}^{N_y}\left[\frac{1}{2}\log\left(2\pi\sigma_{\text{FB}}^2(x,y)\right)+\frac{1}{2}\frac{\left(S(x,y)-S_{\text{FB}}(x,y)\right)^2}{\sigma_{\text{FB}}^2(x,y)}\right]\right)}{\partial C_k(i,j)}$$

For clarity I split the log-likelihood into terms and compute the derivative for each term as follows

$$-\log p_{\text{SEM}}\left(S\middle|H_{\sigma_k}\right)=\frac{1}{2}\sum_{x,y}[R+Q]$$

$$R=\log\left(2\pi\sigma_{\text{FB}}^2(x,y)\right),\ Q=\frac{\left(S(x,y)-S_{\text{FB}}(x,y)\right)^2}{\left(\sigma_{\text{FB}}^2(x,y)\right)}$$

Substituting $\sigma_{\text{FB}}^2(x,y)=aS_{\text{FB}}(x,y)+b$ (see section 9.2.2), the derivatives of R and Q are

$$\begin{aligned}\frac{\partial R}{\partial C_k(i,j)}&=\frac{\partial\log\left(2\pi\sigma_{\text{FB}}^2(x,y)\right)}{\partial C_k(i,j)}=\cancel{\frac{\partial\log(2\pi)}{\partial C_k(i,j)}}+\frac{\partial\log\left(\sigma_{\text{FB}}^2(x,y)\right)}{\partial C_k(i,j)}\\&=\frac{1}{\sigma_{\text{FB}}^2(x,y)}\frac{\partial\sigma_{\text{FB}}^2(x,y)}{\partial C_k(i,j)}\\&=\frac{1}{aS_{\text{FB}}(x,y)+b}\frac{\partial\left(aS_{\text{FB}}(x,y)+b\right)}{\partial C_k(i,j)}\quad\text{substituting } aS_{\text{FB}}(x,y)+b \text{ for } \sigma_{\text{FB}}^2(x,y)\\&=\frac{a}{aS_{\text{FB}}(x,y)+b}\frac{\partial S_{\text{FB}}(x,y)}{\partial C_k(i,j)}\end{aligned}$$

$$\frac{\partial Q}{\partial C_k(i,j)} =$$

$$\left(S(x,y)-S_{\mathrm{FB}}(x,y)\right)^2 \frac{\partial\left(\left(\sigma_{\mathrm{FB}}^2(x,y)\right)^{-1}\right)}{\partial C_k(i,j)} + \frac{1}{\sigma_{\mathrm{FB}}^2(x,y)} \frac{\partial\left(\left(S(x,y)-S_{\mathrm{FB}}(x,y)\right)^2\right)}{\partial C_k(i,j)}$$

$$= \frac{-\left(S(x,y)-S_{\mathrm{FB}}(x,y)\right)^2}{\left(\sigma_{\mathrm{FB}}^2(x,y)\right)^2} \frac{\partial \sigma_{\mathrm{FB}}^2(x,y)}{\partial C_k(i,j)} + \frac{-2\left(S(x,y)-S_{\mathrm{FB}}(x,y)\right)}{\sigma_{\mathrm{FB}}^2(x,y)} \frac{\partial S_{\mathrm{FB}}(x,y)}{\partial C_k(i,j)}$$

(using chain rule)

$$= \frac{-\left(S(x,y)-S_{\mathrm{FB}}(x,y)\right)^2}{\left(aS_{\mathrm{FB}}(x,y)+b\right)^2} \frac{a\partial S_{\mathrm{FB}}(x,y)}{\partial C_k(i,j)} + \frac{-2\left(S(x,y)-S_{\mathrm{FB}}(x,y)\right)}{\left(aS_{\mathrm{FB}}(x,y)+b\right)} \frac{\partial S_{\mathrm{FB}}(x,y)}{\partial C_k(i,j)}$$

(substituting for $\sigma_{\mathrm{FB}}^2(x,y)$)

$$= \frac{-a\left(S(x,y)-S_{\mathrm{FB}}(x,y)\right)^2}{\left(aS_{\mathrm{FB}}(x,y)+b\right)^2} \frac{\partial S_{\mathrm{FB}}(x,y)}{\partial C_k(i,j)} +$$

$$\frac{-2\left(S(x,y)-S_{\mathrm{FB}}(x,y)\right)}{\left(aS_{\mathrm{FB}}(x,y)+b\right)} \frac{\left(aS_{\mathrm{FB}}(x,y)+b\right)}{\left(aS_{\mathrm{FB}}(x,y)+b\right)} \frac{\partial S_{\mathrm{FB}}(x,y)}{\partial C_k(i,j)} \qquad \text{(multiplying by 1)}$$

$$= \frac{-a\left(\left(S(x,y)\right)^2 - 2S(x,y)S_{\mathrm{FB}}(x,y) + \left(S_{\mathrm{FB}}(x,y)\right)^2\right)}{\left(aS_{\mathrm{FB}}(x,y)+b\right)^2} \frac{\partial S_{\mathrm{FB}}(x,y)}{\partial C_k(i,j)} +$$

$$\frac{-2a\left(S_{\mathrm{FB}}(x,y)S(x,y) - \left(S_{\mathrm{FB}}(x,y)\right)^2\right) - 2b\left(S(x,y)-S_{\mathrm{FB}}(x,y)\right)}{\left(aS_{\mathrm{FB}}(x,y)+b\right)^2} \frac{\partial S_{\mathrm{FB}}(x,y)}{\partial C_k(i,j)}$$

(evaluating products)

$$= \frac{a\left(\left(S_{\mathrm{FB}}(x,y)\right)^2 - \left(S(x,y)\right)^2\right) + 2b\left(S_{\mathrm{FB}}(x,y)-S(x,y)\right)}{\left(aS_{\mathrm{FB}}(x,y)+b\right)^2} \frac{\partial S_{\mathrm{FB}}(x,y)}{\partial C_k(i,j)} \qquad \text{(simplification)}$$

Summing all the terms together gives the derivative of the negative log-likelihood in terms of the derivative of each pixel in the simulated image with respect to the surface basis function coefficient:

$$\frac{\partial\left(-\log p_{\mathrm{SEM}}\left(S\middle|H_k\right)\right)}{\partial C_k(i,j)} = \frac{1}{2}\sum_{x,y}\left[\left(\frac{a}{aS_{\mathrm{FB}}(x,y)+b}+\frac{a\left(\left(S_{\mathrm{FB}}(x,y)\right)^2-\left(S(x,y)\right)^2\right)+2b\left(S_{\mathrm{FB}}(x,y)-S(x,y)\right)}{\left(aS_{\mathrm{FB}}(x,y)+b\right)^2}\right)\frac{\partial S_{\mathrm{FB}}(x,y)}{\partial C_k(i,j)}\right]$$

Equation 9-4

Using $S_{\mathrm{FB}}(x,y)=d+\sum_{n=1..N_{\mathrm{FB}}} a_n G_n(x,y)+b_n\kappa_{+,n}(x,y)+c_n\kappa_{-,n}(x,y)$, (recall from section 6.7.2 that d, a_n, b_n, c_n (n=1..N_{FB}) are coefficients of the filter bank outputs for the filter bank SEM model) the derivative of the simulated SEM image in terms of the derivatives of the filter bank outputs is

$$\frac{\partial S_{\mathrm{FB}}(x,y)}{\partial C_k(i,j)}=\sum_{n=1..N_{\mathrm{FB}}} a_n\frac{\partial G_n(x,y)}{\partial C_k(i,j)}+b_n\frac{\partial \kappa_{+,n}(x,y)}{\partial C_k(i,j)}+c_n\frac{\partial \kappa_{-,n}(x,y)}{\partial C_k(i,j)}$$

Equation 9-5

Recall from section 6.5 that the output of the gradient and principal curvature filters are defined in terms of the facet model coefficients as

$$G_n(x,y)=\sqrt{K_{2,n}^2(x,y)+K_{3,n}^2(x,y)}$$
$$\kappa_{+,n}(x,y)=\left(K_{6,n}(x,y)+K_{4,n}(x,y)+\sqrt{\left(K_{6,n}(x,y)-K_{4,n}(x,y)\right)^2+K_{5,n}^2(x,y)}\right)$$
$$\kappa_{-,n}(x,y)=\left(K_{6,n}(x,y)+K_{4,n}(x,y)-\sqrt{\left(K_{6,n}(x,y)-K_{4,n}(x,y)\right)^2+K_{5,n}^2(x,y)}\right)$$

Equation 9-6

Using the chain rule, the derivatives of the filter bank outputs can be written in terms of the derivatives of the facet model coefficients as

$$\frac{\partial G_n(x,y)}{\partial C_k(i,j)} = \frac{\partial G_n(x,y)}{\partial K_{2,n}(x,y)} \frac{\partial K_{2,n}(x,y)}{\partial C_k(i,j)} + \frac{\partial G_n(x,y)}{\partial K_{3,n}(x,y)} \frac{\partial K_{3,n}(x,y)}{\partial C_k(i,j)}$$

$$\begin{aligned}\frac{\partial \kappa_{\pm,n}(x,y)}{\partial C_k(i,j)} &= \frac{\partial \kappa_{\pm,n}(x,y)}{\partial K_{4,n}(x,y)} \frac{\partial K_{4,n}(x,y)}{\partial C_k(i,j)} + \\ &\quad \frac{\partial \kappa_{\pm,n}(x,y)}{\partial K_{5,n}(x,y)} \frac{\partial K_{5,n}(x,y)}{\partial C_k(i,j)} + \frac{\partial \kappa_{\pm,n}(x,y)}{\partial K_{6,n}(x,y)} \frac{\partial K_{6,n}(x,y)}{\partial C_k(i,j)}\end{aligned}$$

Equation 9-7

From Equation 9-6, the derivatives of the gradient magnitude and principal curvatures with respect to the facet model coefficients are

$$\frac{\partial G_n(x,y)}{\partial K_{2,n}(x,y)} = \frac{K_{2,n}(x,y)}{\sqrt{K_{2,n}^2(x,y) + K_{3,n}^2(x,y)}} \qquad \frac{\partial G_n(x,y)}{\partial K_{3,n}(x,y)} = \frac{K_{3,n}(x,y)}{\sqrt{K_{2,n}^2(x,y) + K_{3,n}^2(x,y)}}$$

$$\frac{\partial \kappa_{\pm,n}(x,y)}{\partial K_{4,n}(x,y)} = \left(1 \pm \frac{K_{4,n}(x,y) - K_{6,n}(x,y)}{\sqrt{\left(K_{6,n}(x,y) - K_{4,n}(x,y)\right)^2 + K_{5,n}^2(x,y)}}\right)$$

$$\frac{\partial \kappa_{\pm,n}(x,y)}{\partial K_{5,n}(x,y)} = \frac{\pm K_{5,n}(x,y)}{\sqrt{\left(K_{6,n}(x,y) - K_{4,n}(x,y)\right)^2 + K_{5,n}^2(x,y)}}$$

$$\frac{\partial \kappa_{\pm,n}(x,y)}{\partial K_{6,n}(x,y)} = \left(1 \pm \frac{K_{6,n}(x,y) - K_{4,n}(x,y)}{\sqrt{\left(K_{6,n}(x,y) - K_{4,n}(x,y)\right)^2 + K_{5,n}^2(x,y)}}\right)$$

Because the surface is defined (see section 9.1) in terms of a convolution of the basis function coefficient image with a Gaussian function at scale *k* (G_k) we get the following result for each of the required facet model coefficient derivatives (*m*=2,3,4,5,6)

$$\begin{aligned}\frac{\partial K_{m,n}(x,y)}{\partial C_k(i,j)} &= \frac{\partial\left(C_k \otimes G_k^F \otimes K_{m,n}^F\right)(x,y)}{\partial C_k(i,j)} \\ &= \frac{\partial\left(\sum_{i=1}^{N_x}\sum_{j=1}^{N_y} C_k(i,j)\cdot\left(G_k^F \otimes K_{m,n}^F\right)(x-i,y-j)\right)}{\partial C_k(i,j)} \\ &= \left(G_k^F \otimes K_{m,n}^F\right)(x-i,y-j)\end{aligned}$$

Equation 9-8

($C_k \otimes G_k^F \otimes K_{m,n}^F$ is the convolution of the surface basis function coefficient image with the Gaussian kernel followed by convolution with the cubic facet model coefficient kernel for term *m* of the cubic polynomial and neighborhood *n*.)

For computing the SEM objective function efficiently using convolution, it will be useful to write Equation 9-8 in terms of an image indexed by $(i-x, j-y)$ rather than $(x-i, y-j)$. To transform the equation in this way we must use the symmetry in the Gaussian basis functions and the facet model coefficient filters (G_k^F and $K_{m,n}$):

$$G_k^F(x,y) = G_k^F(-x,y) = G_k^F(x,-y)$$

$$\begin{aligned}
K_{2,n}(x,y) &= K_{2,n}(-x,y) = -K_{2,n}(x,-y) \\
K_{3,n}(x,y) &= K_{2,n}(x,-y) = -K_{2,n}(-x,y) \\
K_{4,n}(x,y) &= K_{4,n}(-x,y) = K_{4,n}(x,-y) \\
K_{5,n}(x,y) &= -K_{5,n}(x,-y) = -K_{5,n}(-x,y) \\
K_{6,n}(x,y) &= K_{6,n}(-x,y) = K_{6,n}(x,-y)
\end{aligned}$$

Using these equations and the definition of convolution we have

$$\begin{aligned}
\frac{\partial K_{2,n}(x,y)}{\partial C_k(i,j)} &= \left(G_k^F \otimes K_{2,n}^F\right)(x-i, y-j) = -\left(G_k^F \otimes K_{2,n}^F\right)(i-x, j-y) \\
\frac{\partial K_{3,n}(x,y)}{\partial C_k(i,j)} &= \left(G_k^F \otimes K_{3,n}^F\right)(x-i, y-j) = -\left(G_k^F \otimes K_{3,n}^F\right)(i-x, j-y) \\
\frac{\partial K_{4,n}(x,y)}{\partial C_k(i,j)} &= \left(G_k^F \otimes K_{4,n}^F\right)(x-i, y-j) = \left(G_k^F \otimes K_{4,n}^F\right)(i-x, j-y) \\
\frac{\partial K_{5,n}(x,y)}{\partial C_k(i,j)} &= \left(G_k^F \otimes K_{5,n}^F\right)(x-i, y-j) = \left(G_k^F \otimes K_{5,n}^F\right)(i-x, j-y) \\
\frac{\partial K_{6,n}(x,y)}{\partial C_k(i,j)} &= \left(G_k^F \otimes K_{6,n}^F\right)(x-i, y-j) = \left(G_k^F \otimes K_{6,n}^F\right)(i-x, j-y)
\end{aligned}$$

Equation 9-9

Using Equation 9-9 to substitute for $\frac{\partial K_{m,n}(x,y)}{\partial C_k(i,j)}$ in Equation 9-7 we get

$$\frac{\partial G_n(x,y)}{\partial C_k(i,j)} = \frac{-K_{2,n}(x,y)\left(G_k^F \otimes K_{2,n}^F\right)(i-x,j-y) - K_{3,n}(x,y)\left(G_k^F \otimes K_{3,n}^F\right)(i-x,j-y)}{\sqrt{K_{2,n}^2(x,y)+K_{3,n}^2(x,y)}}$$

$$\begin{aligned}\frac{\partial \kappa_{\pm,n}(x,y)}{\partial C_k(i,j)} = &\left(1 \pm \frac{K_{4,n}(x,y)-K_{6,n}(x,y)}{\sqrt{\left(K_{6,n}(x,y)-K_{4,n}(x,y)\right)^2+K_{5,n}^2(x,y)}}\right)\left(G_k^F \otimes K_{4,n}^F\right)(i-x,j-y) + \\ &\left(\frac{\pm K_{5,n}(x,y)}{\sqrt{\left(K_{6,n}(x,y)-K_{4,n}(x,y)\right)^2+K_{5,n}^2(x,y)}}\right)\left(G_k^F \otimes K_{5,n}^F\right)(i-x,j-y) + \\ &\left(1 \pm \frac{K_{6,n}(x,y)-K_{4,n}(x,y)}{\sqrt{\left(K_{6,n}(x,y)-K_{4,n}(x,y)\right)^2+K_{5,n}^2(x,y)}}\right)\left(G_k^F \otimes K_{6,n}^F\right)(i-x,j-y)\end{aligned}$$

Equation 9-10

Using Equation 9-10 to substitute for $\frac{\partial G_n(x,y)}{\partial C_k(i,j)}$ and $\frac{\partial \kappa_{\pm,n}(x,y)}{\partial C_k(i,j)}$ in Equation 9-5 we get

$$
\begin{aligned}
\frac{\partial S_{\mathrm{FB}}(x,y)}{\partial C_k(i,j)} = {} & \sum_{n=1..N_{\mathrm{FB}}} a_n \frac{-K_{2,n}(x,y)\left(G_k^F \otimes K_{2,n}^F\right)(i-x,j-y)}{\sqrt{K_{2,n}^2(x,y)+K_{3,n}^2(x,y)}} + \\
& \sum_{n=1..N_{\mathrm{FB}}} a_n \frac{-K_{3,n}(x,y)\left(G_k^F \otimes K_{3,n}^F\right)(i-x,j-y)}{\sqrt{K_{2,n}^2(x,y)+K_{3,n}^2(x,y)}} + \\
& \sum_{n=1..N_{\mathrm{FB}}} b_n \left(1+\frac{K_{4,n}-K_{6,n}}{\sqrt{\left(K_{6,n}-K_{4,n}\right)^2+K_{5,n}^2}}\right)(x,y)\left(G_k^F \otimes K_{4,n}^F\right)(i-x,j-y) + \\
& \sum_{n=1..N_{\mathrm{FB}}} b_n \left(\frac{+K_{5,n}}{\sqrt{\left(K_{6,n}-K_{4,n}\right)^2+K_{5,n}^2}}\right)(x,y)\left(G_k^F \otimes K_{5,n}^F\right)(i-x,j-y) + \\
& \sum_{n=1..N_{\mathrm{FB}}} b_n \left(1+\frac{K_{6,n}-K_{4,n}}{\sqrt{\left(K_{6,n}-K_{4,n}\right)^2+K_{5,n}^2}}\right)(x,y)\left(G_k^F \otimes K_{6,n}^F\right)(i-x,j-y) + \\
& \sum_{n=1..N_{\mathrm{FB}}} c_n \left(1-\frac{K_{4,n}-K_{6,n}}{\sqrt{\left(K_{6,n}-K_{4,n}\right)^2+K_{5,n}^2}}\right)(x,y)\left(G_k^F \otimes K_{4,n}^F\right)(i-x,j-y) + \\
& \sum_{n=1..N_{\mathrm{FB}}} c_n \left(\frac{-K_{5,n}}{\sqrt{\left(K_{6,n}-K_{4,n}\right)^2+K_{5,n}^2}}\right)(x,y)\left(G_k^F \otimes K_{5,n}^F\right)(i-x,j-y) + \\
& \sum_{n=1..N_{\mathrm{FB}}} c_n \left(1-\frac{K_{6,n}-K_{4,n}}{\sqrt{\left(K_{6,n}-K_{4,n}\right)^2+K_{5,n}^2}}\right)(x,y)\left(G_k^F \otimes K_{6,n}^F\right)(i-x,j-y)
\end{aligned}
$$

This can be simplified to

$$\frac{\partial S_{\mathrm{FB}}(x,y)}{\partial C_k(i,j)} = \sum_{n=1..N_{\mathrm{FB}}} -a_n \frac{K_{2,n}(x,y)\left(G_k^F \otimes K_{2,n}^F\right)(i-x,j-y)}{\sqrt{K_{2,n}^2(x,y)+K_{3,n}^2(x,y)}} +$$
$$\sum_{n=1..N_{\mathrm{FB}}} -a_n \frac{K_{3,n}(x,y)\left(G_k^F \otimes K_{3,n}^F\right)(i-x,j-y)}{\sqrt{K_{2,n}^2(x,y)+K_{3,n}^2(x,y)}} +$$
$$\sum_{n=1..N_{\mathrm{FB}}} \left(\left(b_n+c_n\right) + \frac{\left(b_n-c_n\right)\left(K_{4,n}-K_{6,n}\right)}{\sqrt{\left(K_{6,n}-K_{4,n}\right)^2+K_{5,n}^2}} \right)(x,y)\left(G_k^F \otimes K_{4,n}^F\right)(i-x,j-y) +$$
$$\sum_{n=1..N_{\mathrm{FB}}} \left(\frac{K_{5,n}\left(b_n-c_n\right)}{\sqrt{\left(K_{6,n}-K_{4,n}\right)^2+K_{5,n}^2}} \right)(x,y)\left(G_k^F \otimes K_{5,n}^F\right)(i-x,j-y) +$$
$$\sum_{n=1..N_{\mathrm{FB}}} \left(\left(b_n+c_n\right) + \frac{\left(b_n-c_n\right)\left(K_{6,n}-K_{4,n}\right)}{\sqrt{\left(K_{6,n}-K_{4,n}\right)^2+K_{5,n}^2}} \right)(x,y)\left(G_k^F \otimes K_{6,n}^F\right)(i-x,j-y)$$

Equation 9-11

For convenience I define an image *D* as

$$D(x,y) = \frac{1}{2}\left(\begin{array}{l} \dfrac{a\left(\left(S_{\mathrm{FB}}(x,y)\right)^2-\left(S(x,y)\right)^2\right)+2b\left(S_{\mathrm{FB}}(x,y)-S(x,y)\right)}{\left(aS_{\mathrm{FB}}(x,y)+b\right)^2} \\ +\dfrac{a}{aS_{\mathrm{FB}}(x,y)+b} \end{array} \right)$$

Equation 9-12

D is defined mainly for the purpose of simplification but intuitively it represents a measure of the difference between the simulated and real SEM images, weighting the difference more where the noise is expected to be lower. Using Equation 9-11 to substitute for $\dfrac{\partial S_{\mathrm{FB}}(x,y)}{\partial C(i,j)}$ in Equation 9-4 we get

$$\frac{\partial\left(-\log p_{\mathrm{SEM}}\left(S\left|H_k\right.\right)\right)}{\partial C_k\left(i,j\right)}=$$

$$-\sum_{n=1..N_{\mathrm{FB}}}\sum_{x,y}D\left(x,y\right)a_n\frac{K_{2,n}\left(x,y\right)\left(G_k^F\otimes K_{2,n}^F\right)\left(i-x,j-y\right)}{\sqrt{K_{2,n}^2\left(x,y\right)+K_{3,n}^2\left(x,y\right)}}+$$

$$-\sum_{n=1..N_{\mathrm{FB}}}\sum_{x,y}D\left(x,y\right)a_n\frac{K_{3,n}\left(x,y\right)\left(G_k^F\otimes K_{3,n}^F\right)\left(i-x,j-y\right)}{\sqrt{K_{2,n}^2\left(x,y\right)+K_{3,n}^2\left(x,y\right)}}+$$

$$\sum_{n=1..N_{\mathrm{FB}}}\sum_{x,y}D\left(x,y\right)\left(\left(b_n+c_n\right)+\frac{\left(b_n-c_n\right)\left(K_{4,n}-K_{6,n}\right)}{\sqrt{\left(K_{6,n}-K_{4,n}\right)^2+K_{5,n}^2}}\right)\left(x,y\right)\left(G_k^F\otimes K_{4,n}^F\right)\left(i-x,j-y\right)+$$

$$\sum_{n=1..N_{\mathrm{FB}}}\sum_{x,y}D\left(x,y\right)\left(\frac{K_{5,n}\left(b_n-c_n\right)}{\sqrt{\left(K_{6,n}-K_{4,n}\right)^2+K_{5,n}^2}}\right)\left(x,y\right)\left(G_k^F\otimes K_{5,n}^F\right)\left(i-x,j-y\right)+$$

$$\sum_{n=1..N_{\mathrm{FB}}}\sum_{x,y}D\left(x,y\right)\left(\left(b_n+c_n\right)+\frac{\left(b_n-c_n\right)\left(K_{6,n}-K_{4,n}\right)}{\sqrt{\left(K_{6,n}-K_{4,n}\right)^2+K_{5,n}^2}}\right)\left(x,y\right)\left(G_k^F\otimes K_{6,n}^F\right)\left(i-x,j-y\right)$$

Equation 9-13

Rewriting the summations over (*x,y*) as convolutions and factoring out the convolution with G_k^F shows that the gradient can be computed efficiently using O(N_{FB}) convolutions:

$$\frac{\partial\left(-\log p_{\mathrm{SEM}}\left(S \mid H_k\right)\right)}{\partial C_k\left(i,j\right)} =$$

$$\left[\begin{array}{l}
-\sum_{n=1..N_{\mathrm{FB}}}\left[D(x,y)\,a_n \frac{K_{2,n}(x,y)}{\sqrt{K_{2,n}^2(x,y)+K_{3,n}^2(x,y)}}\right]\otimes K_{2,n}^F + \\
-\sum_{n=1..N_{\mathrm{FB}}}\left[D(x,y)\,a_n \frac{K_{3,n}(x,y)}{\sqrt{K_{2,n}^2(x,y)+K_{3,n}^2(x,y)}}\right]\otimes K_{3,n}^F + \\
\sum_{n=1..N_{\mathrm{FB}}}\left[D(x,y)\left((b_n+c_n)+\frac{(b_n-c_n)(K_{4,n}-K_{6,n})}{\sqrt{(K_{6,n}-K_{4,n})^2+K_{5,n}^2}}\right)(x,y)\right]\otimes K_{4,n}^F + \\
\sum_{n=1..N_{\mathrm{FB}}}\left[D(x,y)\left(\frac{K_{5,n}(b_n-c_n)}{\sqrt{(K_{6,n}-K_{4,n})^2+K_{5,n}^2}}\right)(x,y)\right]\otimes K_{5,n}^F + \\
\sum_{n=1..N_{\mathrm{FB}}}\left[D(x,y)\left((b_n+c_n)+\frac{(b_n-c_n)(K_{6,n}-K_{4,n})}{\sqrt{(K_{6,n}-K_{4,n})^2+K_{5,n}^2}}\right)(x,y)\right]\otimes K_{6,n}^F
\end{array}\right]\otimes G_k^F$$

Equation 9-14

9.3.2.2 Algorithm for Computing the Gradient

In summary, the steps for computing the gradient are

1. convolve the height estimate (itself a convolution between the coefficient image and a Gaussian kernel) $H_k = C_k \otimes G_k^F$ with each of $5N_{FB}$ facet model coefficient kernels to get images $K_{2,n}(x,y), K_{3,n}(x,y), K_{4,n}(x,y), K_{5,n}(x,y), K_{6,n}(x,y)$ for n=1..N_{FB}
2. using Equation 9-6, combine the images from step 1 into the simulated SEM image $S_{\mathrm{FB}}(x,y) = d + \sum_{n=1..N_{\mathrm{FB}}} a_n G_n(x,y) + b_n \kappa_{+,n}(x,y) + c_n \kappa_{-,n}(x,y)$
3. compute $D(x,y)$ from the simulated and real SEM images (Equation 9-12)

4. compute the $5N_{FB}$ images (n=1..N_{FB}), (Keep in mind that all arithmetic here is done pixelwise. Quantities such as $K_{3,n}(x,y)$ represents a pixel in the image $K_{3,n}$ so computing $(K_{3,n}(x,y))^2$ requires squaring each pixel in the image $K_{3,n}$):

$$C_{2,n}(x,y) = D(x,y)\,a_n \frac{K_{2,n}(x,y)}{\sqrt{(K_{2,n}(x,y))^2 + (K_{3,n}(x,y))^2}}$$

$$C_{3,n}(x,y) = D(x,y)\,a_n \frac{K_{3,n}(x,y)}{\sqrt{(K_{2,n}(x,y))^2 + (K_{3,n}(x,y))^2}}$$

$$C_{4,n}(x,y) = D(x,y)\left((b_n + c_n) + \frac{(b_n - c_n)(K_{4,n}(x,y) - K_{6,n}(x,y))}{\sqrt{(K_{6,n}(x,y) - K_{4,n}(x,y))^2 + (K_{5,n}(x,y))^2}}\right)$$

$$C_{5,n}(x,y) = D(x,y)\left(\frac{K_{5,n}(x,y)(b_n - c_n)}{\sqrt{(K_{6,n}(x,y) - K_{4,n}(x,y))^2 + (K_{5,n}(x,y))^2}}\right)$$

$$C_{6,n}(x,y) = D(x,y)\left((b_n + c_n) + \frac{(b_n - c_n)(K_{6,n}(x,y) - K_{4,n}(x,y))}{\sqrt{(K_{6,n}(x,y) - K_{4,n}(x,y))^2 + (K_{5,n}(x,y))^2}}\right)$$

5. convolve each of these $5N_{FB}$ with a corresponding convolution kernel and sum the resulting images as

$$B = \sum_{n=1..N_{FB}} \left(-C_{2,n} \otimes K^F_{2,n} - C_{3,n} \otimes K^F_{3,n} + C_{4,n} \otimes K^F_{4,n} + C_{5,n} \otimes K^F_{5,n} + C_{6,n} \otimes K^F_{6,n}\right)$$

where $C_{m,n} \otimes K^F_{m,n}$ represents the convolution of $C_{m,n}$ with the same facet model coefficient kernel used to compute $K_{m,n}(x,y)$.

6. compute the gradient with one more convolution of B with the Gaussian basis function

$$\frac{\partial(-\log P)}{\partial C_k(x,y)} = \left(B \otimes G^F_k\right)(x,y)$$

The evaluation of the gradient typically requires less than 10 times the time required to evaluate the simulated SEM image using the filter bank model.

9.4 Conjugate Gradient Method

To minimize $f\left(H_k\left(x,y\right)\right) = -\log p_{\text{AFM}}\left(A\middle|H_k\right) - \log p_{\text{SEM}}\left(S\middle|H_k\right)$ I use a program called CG+ that implements the non-linear Polak-Ribière conjugate gradient method [Liu00][Gilbert92][Shewchuk94]. I converted the code from FORTRAN to C and added a more flexible C++ interface to make it more convenient to output images and other optimization state during an optimization. There are also options in the code to use the Fletcher-Reeves and positive Polak-Ribière variants of the method but I did not observe a significant difference in speed between the different methods.

The conjugate gradient method requires an initial starting point in the parameter space. I initialize the surface basis function coefficients at all but the largest scale to 0. For the largest scale I initialize the surface from the AFM image eroded by the reconstructed tip shape: $H'_A = A \ominus \hat{T}$. After computing H'_A, I resample it onto the SEM image grid using bilinear interpolation and the transformation matrix $\mathbf{T}_{S\to A}$ to get an image H_A. Next I Gaussian blur this image using the widest Gaussian basis function so that the image used for initialization closely matches the level of detail that can be represented by the surface model. The coefficients are computed from the blurred resampled eroded image ($H_{A,blur}$) using the Van-Cittert iterative deconvolution algorithm [Katsaggelos89]. The deconvolution algorithm sets the first estimate for the coefficient image to the initialization image $C_k^0 = H_{A,blur}$. For each iteration of the deconvolution I compute a residual image $R_i = H_{A,blur} - \left(C_k^i \otimes G_k\right)$ and compute the update to the coefficient image as $C_k^{i+1} = C_k^i + g \cdot R_i$, where g is a somewhat arbitrary relaxation parameter. Results showed that the algorithm was stable and converged in about a minute or less for g=0.1 for an image of size 456x172.

Once a starting point ($C_k(i,j) = x_0$) has been set, the conjugate gradient method initializes the first search direction (d_0) and residual vector (r_0) to the negative gradient vector (computed at x_0):

$$d_0 = r_0 = -\frac{\partial f}{\partial C_k(i,j)}$$

At the beginning of each iteration of the conjugate gradient method, a line-search algorithm is used to find a scalar α_t that minimizes $f(x_t + \alpha_t d_t)$. The iteration is completed by computing the new point in the parameter space, a new residual vector, and a new search direction as

$$\begin{aligned} x_{t+1} &= x_t + \alpha_t d_t \\ r_{t+1} &= -\frac{\partial f}{\partial C_k(i,j)} \\ \beta_{t+1} &= \frac{r_{t+1}^T (r_{t+1} - r_t)}{r_t^T r_t} \\ d_{t+1} &= r_{t+1} + \beta_{t+1} d_t \end{aligned}$$

where the gradient $\frac{\partial f}{\partial C_k(i,j)}$ is computed at $C_k = x_{t+1}$.

The conjugate gradient method is terminated when the magnitude of all components of the gradient vector fall below a threshold: $\max_{i,j}\left|\frac{\partial f}{C_k(i,j)}\right| < \varepsilon \cdot (1+f)$. ε was set to 1e-6.

The line search algorithm used in CG+ is Moré's method [Moré94]. This line search method seeks an α_t that satisfies the conditions:

$$f(x_t + \alpha_t d_t) \le f(x_t) + \mu\alpha_t \left(\frac{\partial f(x_t + \alpha d_t)}{\partial \alpha} \text{ at } \alpha = 0 \right) \text{ and}$$

$$\left(\left| \frac{\partial f(x_t + \alpha d_t)}{\partial \alpha} \right| \text{ at } \alpha = \alpha_t \right) \le \eta \left(\left| \frac{\partial f(x_t + \alpha d_t)}{\partial \alpha} \right| \text{ at } \alpha = 0 \right)$$

for some μ and η in the interval (0,1). The first condition specifies a sufficient decrease in the objective function value but allows for arbitrarily small values of α_t so the method may not converge. The second condition ensures that α_t is not too small and that it is near a local minimizer of $f(x_t + \alpha d_t)$. I refer the reader to [Moré94] for details of the line search algorithm.

In chapter 10 I describe the application of this algorithm to various examples. The time required to complete the surface reconstruction for these examples varied from 2-3.5 hours as described in the captions for Figure 10-8, Figure 10-13, Figure 10-18 and Figure 10-23.

10 Validation

10.1 Selection and Preprocessing of Test Inputs

The reconstruction algorithm was validated using both real and synthetic pairs of AFM and SEM images. The real images were acquired of a specimen called NIST0523, provided by John Dagata at NIST. This specimen was constructed of silicon by an AFM dip-pen lithography technique described in [Chien02]. AFM images were acquired using a Veeco/Topometrix Explorer^{TM} AFM with a 100 micron scanner and a Nanosensors^{TM} FM (force modulation) tip. SEM images were acquired using a Hitachi S4700 Field Emission SEM. SEM images were acquired first because the SEM, with its larger scan region, is much better than the AFM at providing an overview of the specimen which helps to identify useful landmarks for revisiting a specific region of the surface (Figure 10-1). From the SEM images two sites were selected for testing. The first site, referred to as NIST0523_thick, has linear structures that are about 100 nm high, 300 nm wide and 3500-5500 nm long. The second site, referred to as NIST0523_thin, has linear structures that are about 100 nm high and 50-60 nm wide. The locations of these sites are shown in Figure 10-1.

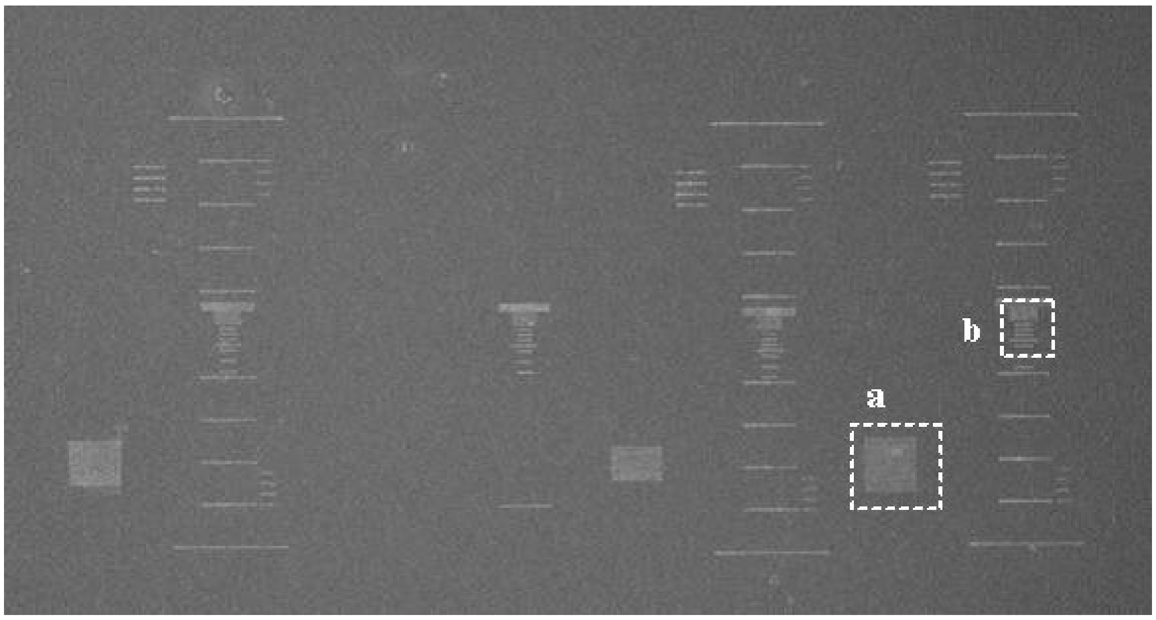


Figure 10-1: Overview SEM image of the NIST0523 specimen showing the NIST0523_thin region (a) and the NIST0523_thick region (b).

At a 1kV accelerating voltage there was noticeable distortion of the image coordinates in the SEM image due to stray magnetic fields making straight lines appear wavy. This distortion was reduced by scanning at lower magnification but this reduced the signal to noise ratio. An image was also taken at 3kV that helped to reduce the effects of interference but this image was accidentally taken with the lowest intensities saturated so I decided to use the low magnification image for testing. These images are shown in Figure 10-2.

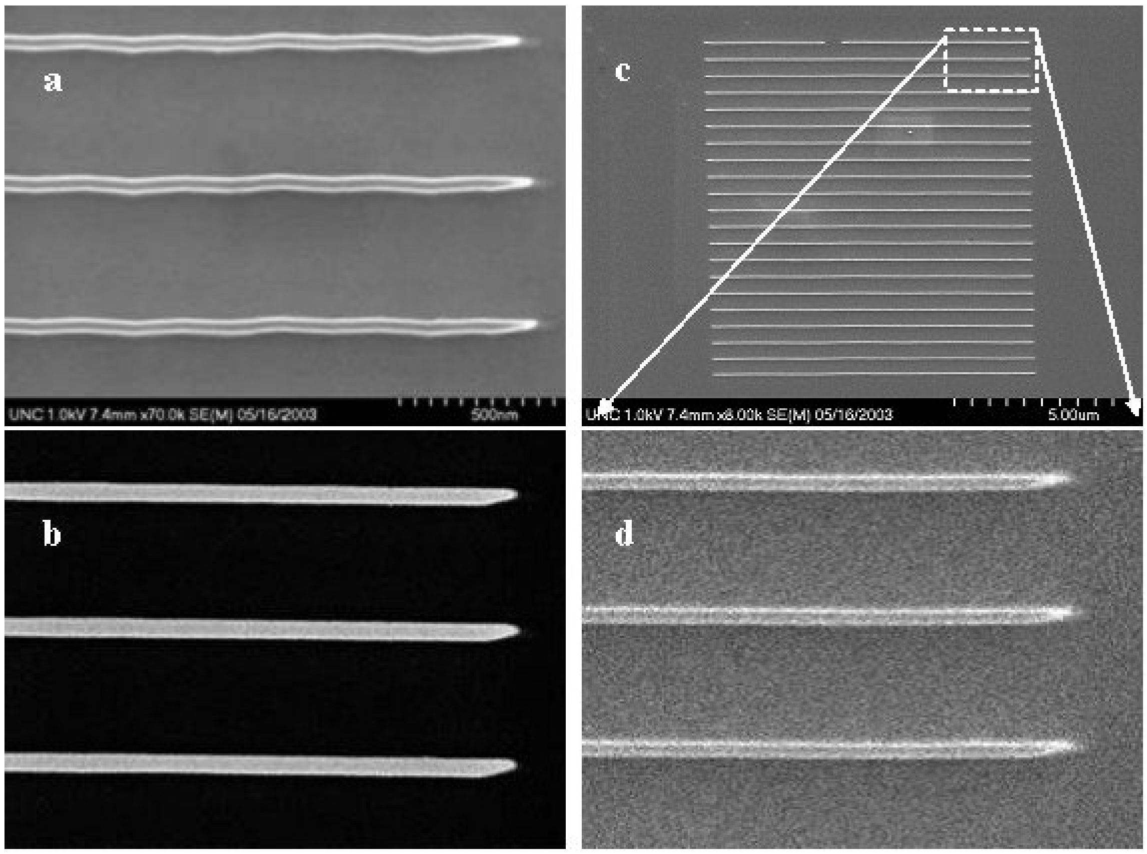

a
c
b
d
UNC 1.0kV 7.4mm x70.0k SE(M) 05/16/2003
500nm
UNC 1.0kV 7.4mm x8.00k SE(M) 05/16/2003
5.00um

Figure 10-2: SEM images of the 50nm wide lines in the NIST0523_thin example: (a) 2560x1920 pixel high magnification image at 1kV that showed waviness due to electromagnetic interference, (b) image at 3kV showing reduced waviness but also saturation at the low end of the intensity range, (c) 2560x1920 pixel low magnification image at 1kV, (d) 320x240 pixel crop of low magnification image showing reduced interference effects

For each of the two real data tests, two AFM images and one SEM image were used. One AFM image was taken with a relatively new tip. The second AFM image was taken with the same tip after it was purposely dulled by scanning in contact mode. All AFM images were acquired in non-contact mode.

The region of the two AFM images and SEM image chosen for the NIST0523_thin test is indicated by dashed squares in Figure 9-3. The white particles in other portions of the image are debris caused by the tip-dulling procedure and are not normally present in AFM images. Unfortunately, during the process of dulling the AFM tip, the tip apparently became contaminated with various particles that were later deposited on the specimen during imaging. This would have introduced errors in using the sharp tip AFM image as a ground truth reference so to avoid this I selected a relatively clean region of the dull tip AFM image where the particles would not introduce errors in the validation procedure. The particles persisted even when the specimen was cleaned by a plasma cleaner so they are likely to be pieces of silicon that were broken off of the surface during contact mode scanning. The relation between the AFM and SEM regions input to the algorithm is shown in Figure 10-4.

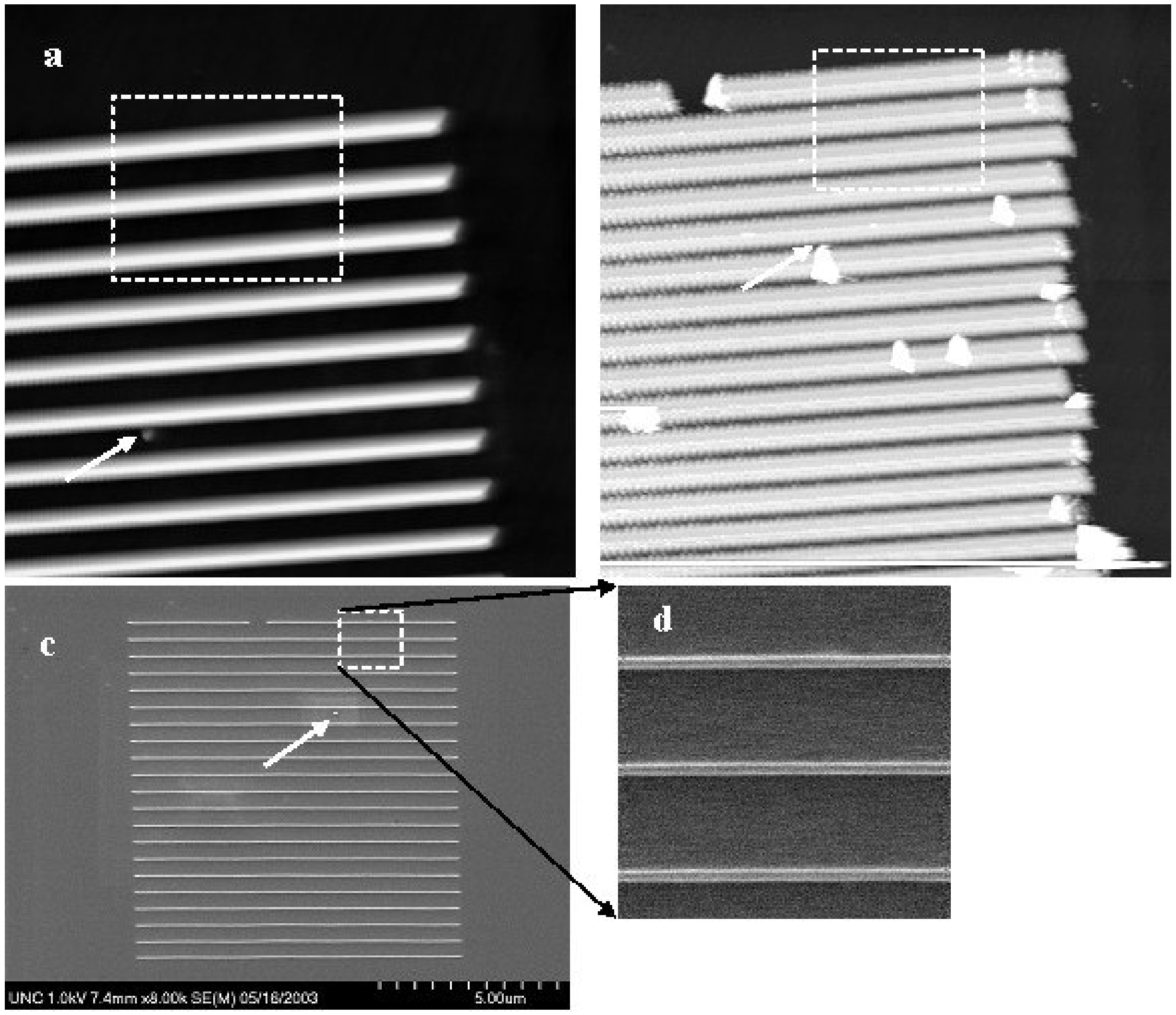


Figure 10-3: Original images and crop rectangles used for the NIST0523_thin example:(a) sharp tip AFM image, (b) dull tip AFM image, (c) SEM image, (d) cropped region of SEM image used for reconstruction and representing the region of the surface to be reconstructed. The white arrows indicate the location of a particle that is present in all the full size images but the dull tip AFM image had many more particles that were probably deposited by the AFM tip during imaging.

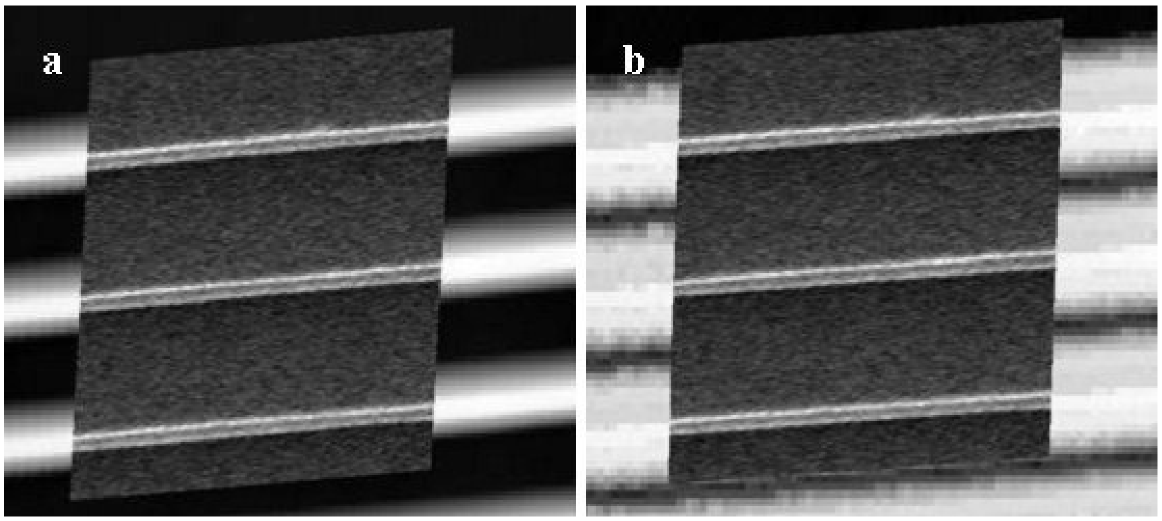


Figure 10-4: (a) Sharp tip AFM image with SEM image resampled and overlaid (b) Dull tip AFM image with SEM image resampled and overlaid. These images show the region for the SEM image (same as that for the reconstructed part of the surface) in relation to the AFM images.

The NIST0523_thick AFM images showed some variation in the drift rate so these images were cropped to ensure that the image distortions would best match the assumption of an affine transformation while keeping enough features in the image to constrain the transformation between the SEM and AFM images. The cropped region for each of the original images is shown in Figure 10-5.

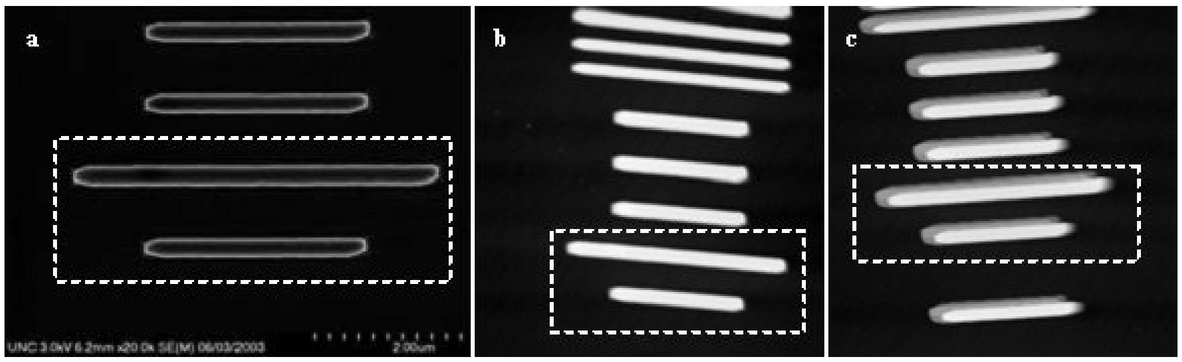


Figure 10-5: Original images and crop rectangles used for the NIST0523_thick example: (a) SEM image, (b) sharp tip AFM image, (c) dull tip AFM image

The AFM tip was reconstructed for each of the four AFM images using the blind tip reconstruction method described in section 4.2. In the tip volume plot used to determine the best threshold there was a single large jump indicating the optimal threshold value for the sharp tip AFM images. For the dull tip AFM images the best threshold value was not so obvious. About 3-4 of the largest jumps in volume were identified each corresponding to a threshold value and tip reconstruction. To determine which of these tips were consistent with the AFM image I opened (eroded and then redilated) the AFM image using each tip and calculated the difference between the opened result and the original AFM image. I selected the broadest tip that approximately satisfied the idempotency condition of the opening operation (Equation C-6) up to the expected noise in the AFM image.

For the NIST0523_thin example, the cropped AFM image did not contain sufficient information to erode all sides of the tip because the pattern was nearly one-dimensional. This poses no problem for the tip reconstruction algorithm but limits the reconstructed tip image to a profile of the tip onto a plane perpendicular to the lines on the specimen surface. Any tip that has the same profile is equally consistent with the AFM image and the reconstructed tip will tend to be a surface of extrusion along the direction of the lines, extending as far as possible within the image allocated for storing the tip image (Figure 10-12). The same tip profile was also used for the synthetic data (Figure 10-22). This limitation was not present in the NIST0523_thick specimen, so a full tip reconstruction was performed.

Both the sharp tip AFM images and dull tip AFM images were eroded by their associated tip estimates and then resampled at the same locations as the SEM image samples for comparison to the reconstructed surfaces. The AFM/SEM-based surface reconstruction was compared to the sharp tip AFM-based surface reconstruction by comparing heights in the space of the reconstructed surface. The difference in height was computed as the sum of squared differences over all pixels. The same measure was used to compare the reconstructed surface with the true surface for the synthetic data tests.

10.2 Construction of Synthetic Test Images

Two sets of synthetic surfaces similar to the real ones were constructed by thresholding either AFM or SEM images. The resulting binary image was then scaled to match the height range of the real surface. Gaussian blurring or grayscale dilation was then used to modify the edge shape. The synthetic surfaces were not based on analytic functions but were constructed of triangles by tessellating regularly sampled height values.

Because the real AFM images had relatively low resolution compared with the scale of structures in the surface topography, special care was required to make similar synthetic AFM images and avoid discretization errors. The discretized dilation operation used to construct the synthetic AFM image was performed on a grid with 3 times the target resolution to reduce errors in approximating a continuous dilation. This involved resampling the synthetic surface into an AFM grid that had been upsampled by a factor of 3 relative to the target AFM image, dilating the upsampled surface by a tip that was also upsampled by a factor of 3, and then downsampling to the resolution of the final synthetic AFM image minus noise. While the discretized dilation by itself should not introduce errors, the combination of resampling into AFM coordinates and discretized dilation would introduce errors because the sample points in AFM coordinates do not align with the vertices of the triangular facets. This procedure is illustrated in Figure 10-6. Correlated noise was added to the AFM image using an autoregressive model. The parameters for the noise were chosen to be the same as those extracted from real AFM images.

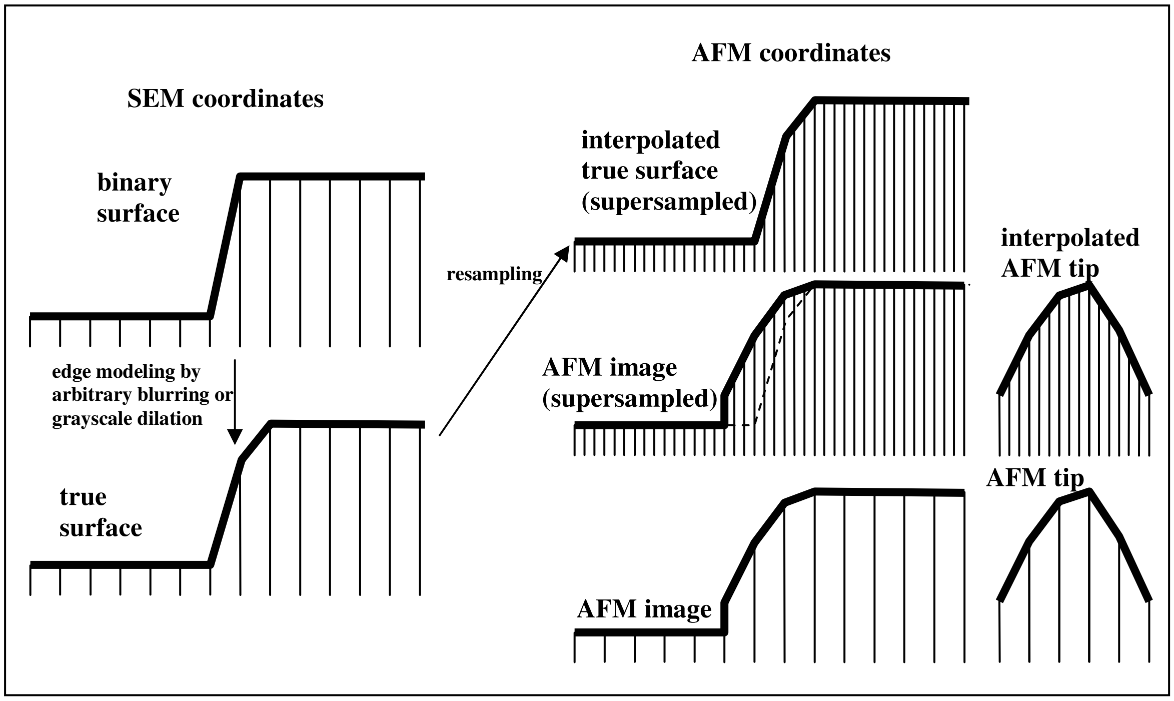


Figure 10-6: Procedure for modeling synthetic test surfaces and AFM images.

Synthetic SEM images were computed using Monte Carlo simulation. The noise in the synthetic SEM images was solely the noise inherent in the Monte Carlo simulation. The SEM to AFM coordinate transformation and AFM tip used for the reconstruction were computed in the same way as for the real data, relying on the registration procedure described in chapter 7 and the blind tip reconstruction algorithm.

10.3 Real Input Tests

This section describes the input AFM and SEM images for the real data tests and the resulting reconstructed surfaces. The reconstructions using combined dull tip AFM and SEM images and dull tip AFM images alone are compared to a reconstruction using a sharp tip AFM image alone. The true surface is not known but the sharp tip AFM image is assumed to provide a significantly closer approximation to the true surface than the dull tip AFM image and is used to estimate the difference from the true surface.

10.3.1 NIST0523 Thick

10.3.1.1 Inputs

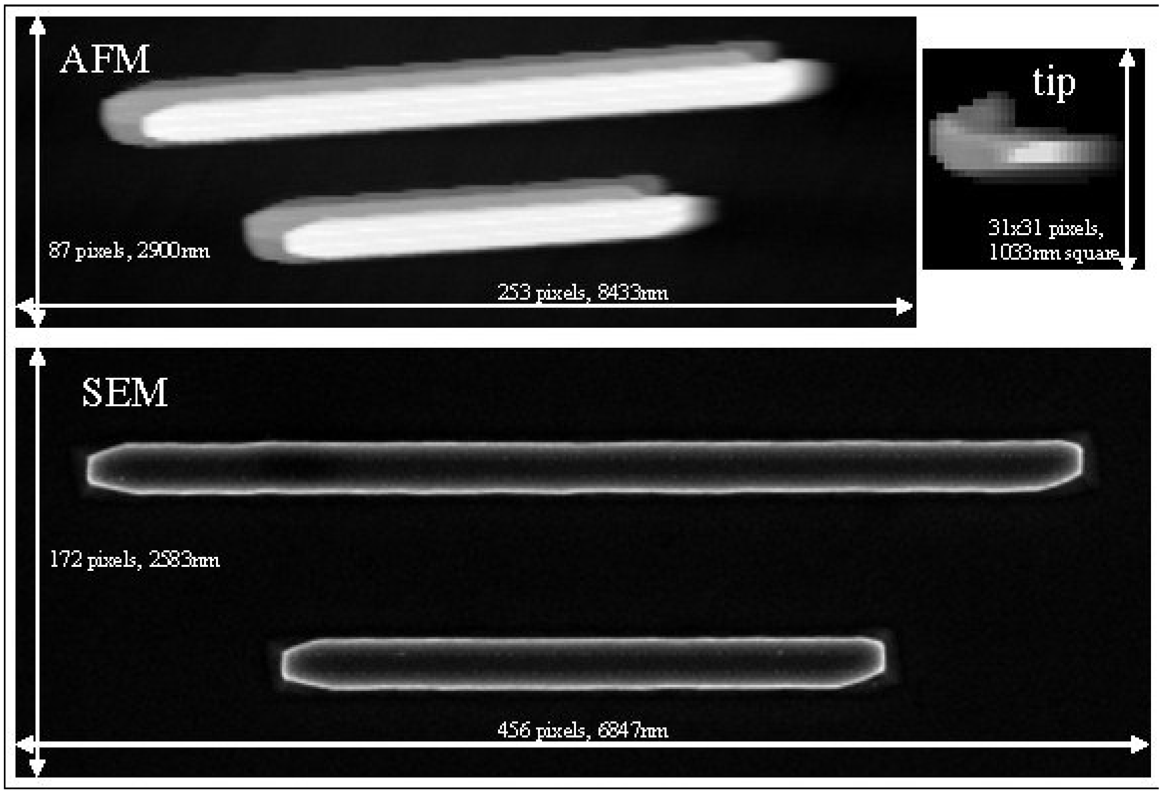


Figure 10-7: Inputs for the NIST0523_thick specimen.

10.3.1.2 Outputs

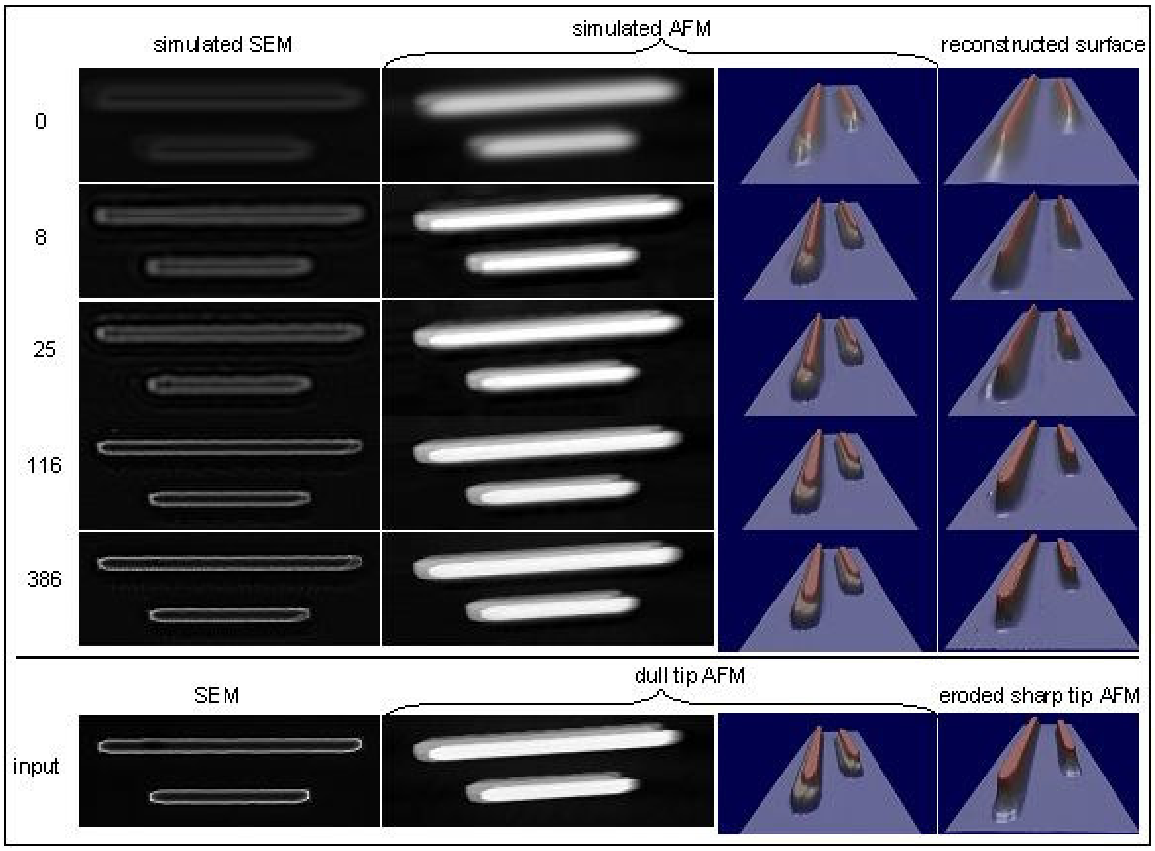


Figure 10-8: Selection of images from the reconstruction for NIST0523_thick. The numbers on the left indicate the total number of conjugate gradient iterations. Excluding the bottom row, the columns from left to right are the simulated SEM image, the simulated AFM image, a 3D visualization of the simulated AFM image, and a 3D visualization of the reconstructed surface. The images in the bottom row from left to right are the input SEM image, input AFM image, 3D visualization of the input AFM image, and a 3D visualization of the reconstruction from an AFM image acquired using a sharper tip than the one used for the combination AFM-SEM reconstruction. This reconstruction required about 3.5 hours.

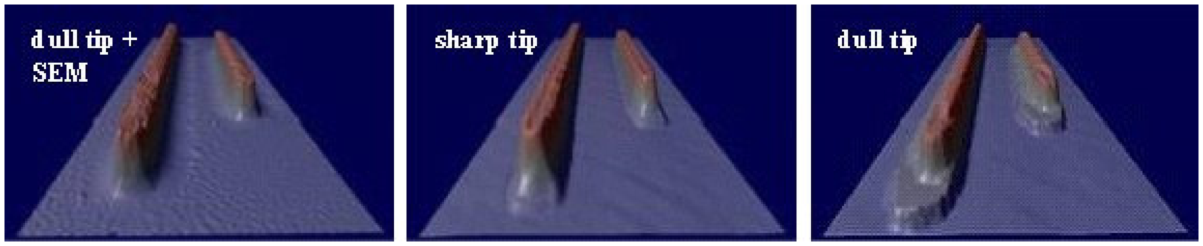


Figure 10-9: Comparison of the surface reconstructed by optimization of the likelihood computed from the dull tip AFM image and the SEM image (left), the surface found by grayscale erosion of an AFM image acquired using a relatively sharp tip (center), and the surface found by grayscale erosion of the dull tip AFM image (right). The similarity of the left two images and their difference from the third displays the effectiveness of the combined technique.

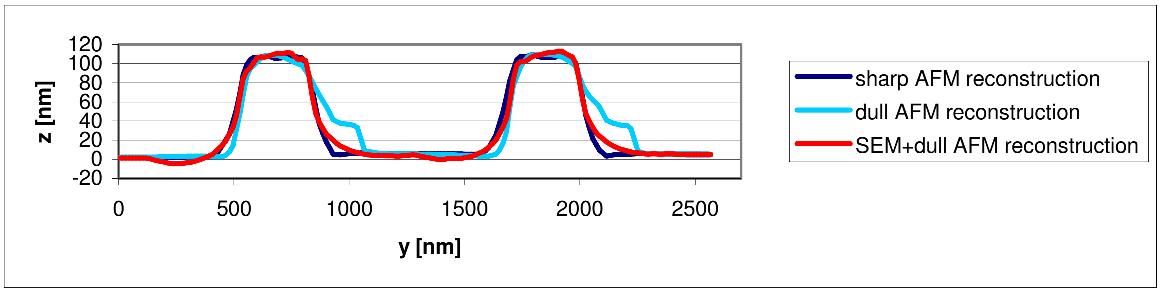


Figure 10-10: Comparison of cross-sections of the reconstructed surfaces shown in Figure 10-9.

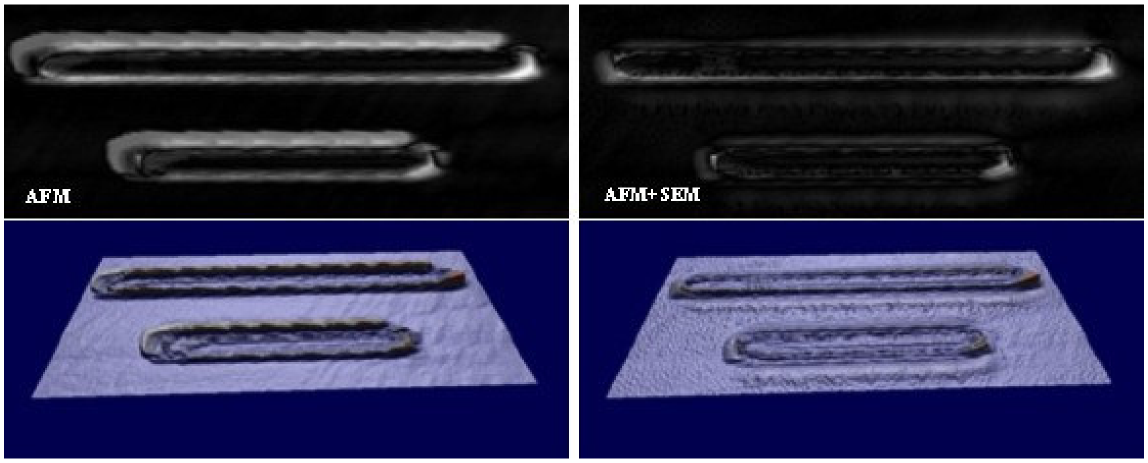


Figure 10-11: Absolute value of the height error judged by the difference from the height estimated by eroding the AFM image acquired with a sharp tip. The error for the reconstruction using only the dull tip AFM image is shown on the left (sum of squares = $1.52\text{x}10^7$) and the error for the reconstruction using both the dull tip AFM image and SEM image is on the right (sum of squares = $3.19\text{x}10^6$). The grayscale mapping to intensity values is identical for both images.

10.3.2 NIST0523 Thin

10.3.2.1 Inputs

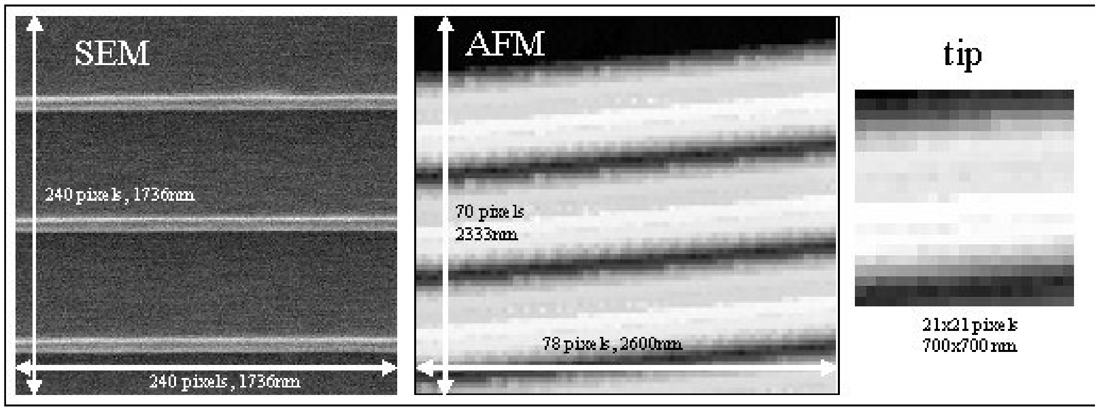


Figure 10-12: Inputs for the NIST0523_thin specimen.

10.3.2.2 Outputs

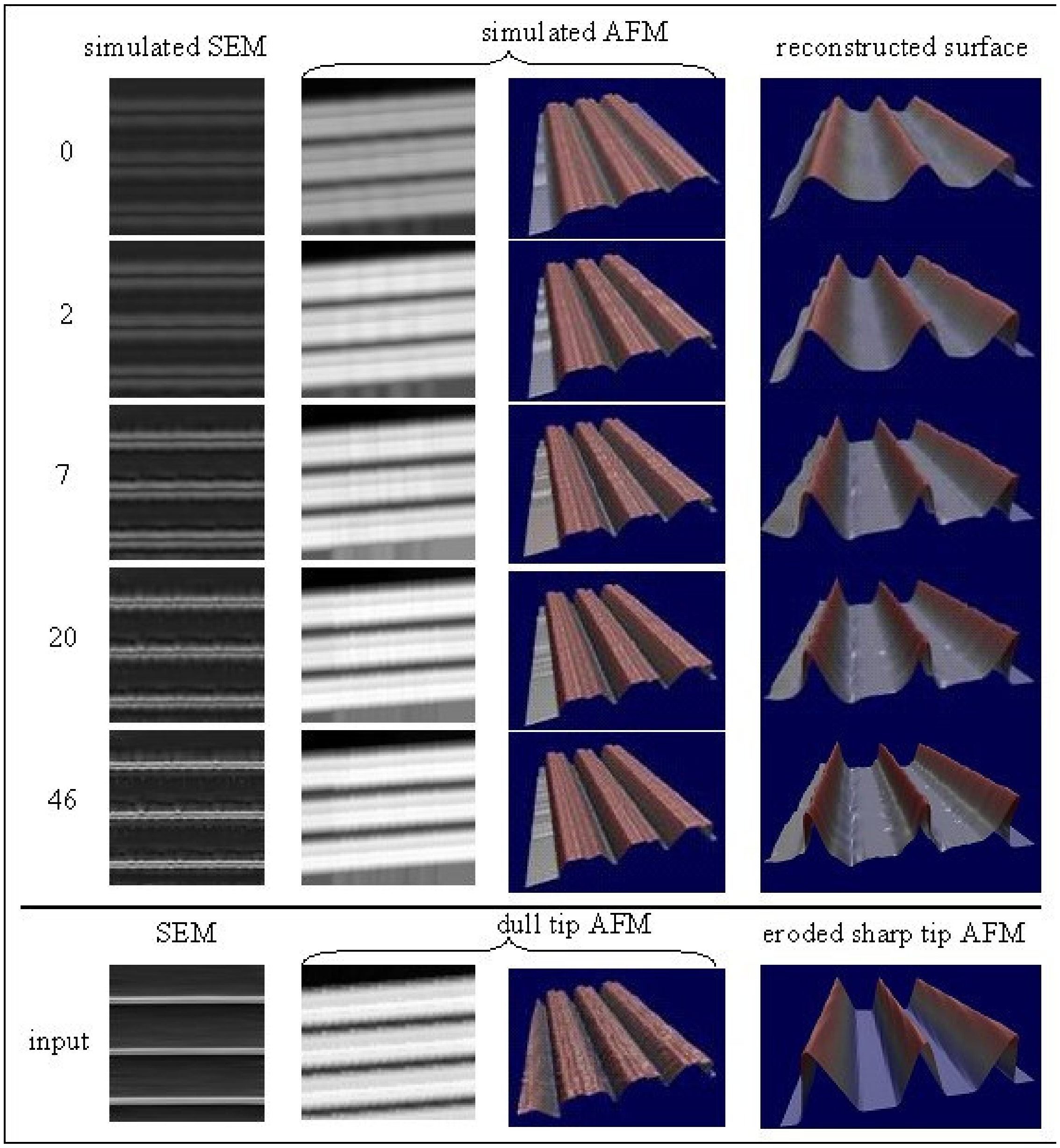


Figure 10-13: Selection of images from the reconstruction for NIST0523_thin. The numbers on the left indicate the total number of conjugate gradient iterations. Excluding the bottom row, the columns from left to right are the simulated SEM image, the simulated AFM image, a 3D visualization of the simulated AFM image, and a 3D visualization of the reconstructed surface. The images in the bottom row from left to right are the input SEM image, input AFM image, 3D visualization of the input AFM image, and a 3D visualization of the reconstruction from an AFM image acquired using a sharper tip than the one used for the combination AFM-SEM reconstruction. The simulated AFM is missing the ridge on the left side of the dull tip AFM surface because this is due to a part of the specimen surface outside the region covered by the surface reconstruction. This reconstruction required 2 hours.

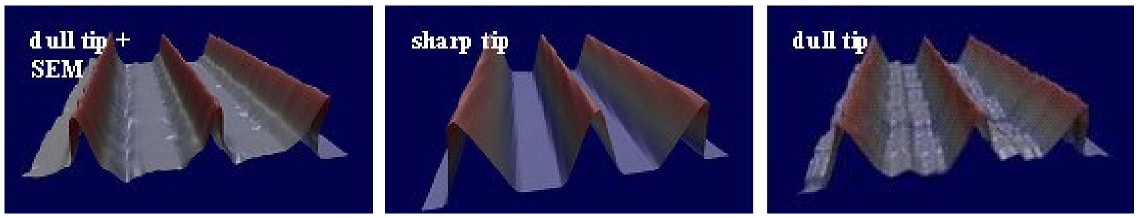


Figure 10-14: Comparison of the surface reconstructed by optimization of the likelihood computed from the AFM image (acquired with a broken tip) and the SEM image (left), the surface found by grayscale erosion of the AFM image acquired using a relatively sharp tip (center), and the surface found by grayscale erosion of the AFM image acquired using a broken tip (right). Although the dull tip is much wider at the apex, the sides of the dull tip have the same slope as the sides of the sharp tip and this makes the reconstruction from the dull tip near the tops of the ridges very close to the reconstruction from the sharp tip. The sharp tip reconstruction does not provide a significantly better reference for the true surface than the dull tip reconstruction so the performance of the reconstruction algorithm is inconclusive in this case.

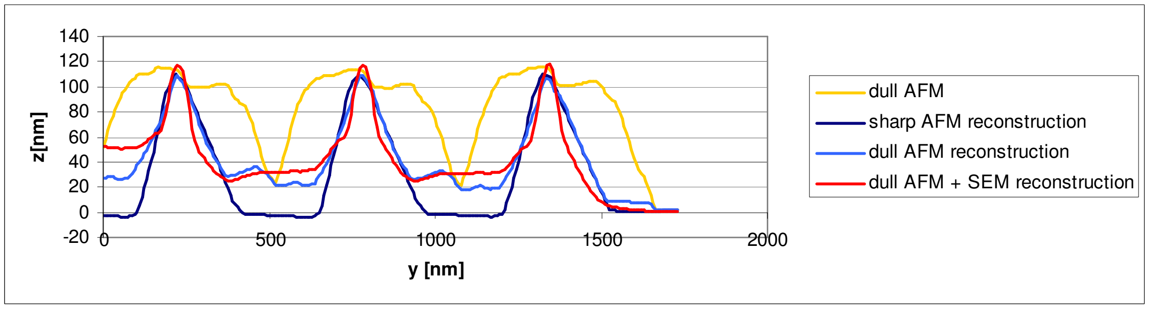


Figure 10-15: Comparison of cross-sections of the surfaces shown in Figure 10-14. The profile from the original dull AFM image is also shown.

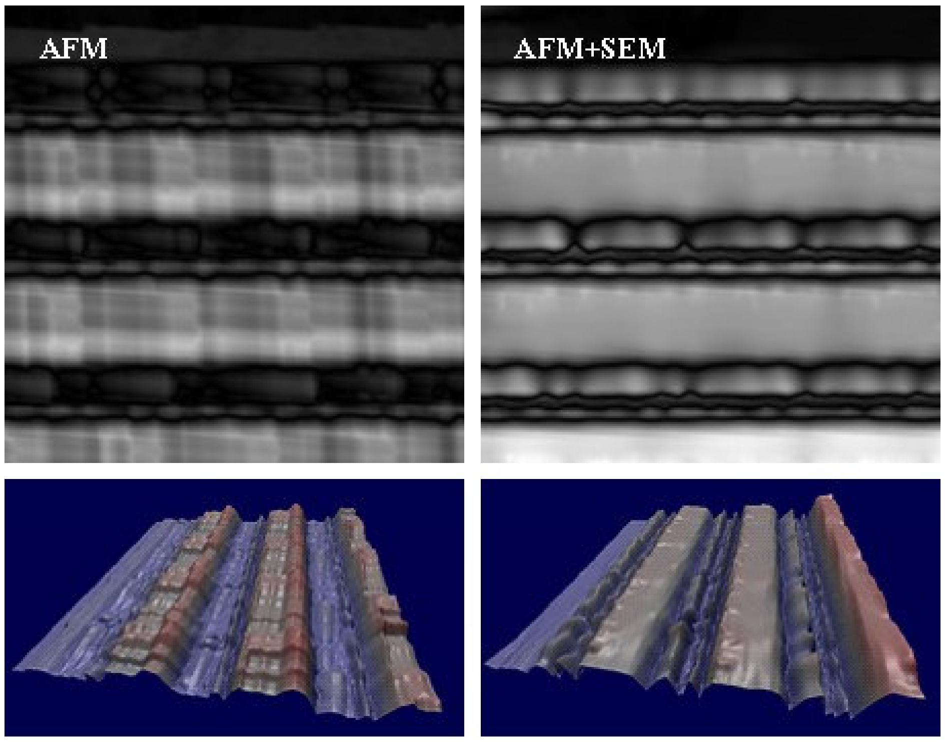


Figure 10-16: Absolute value of the height error judged by the difference from the height estimated by eroding the AFM image acquired with a sharp tip. The error for the reconstruction using only the dull tip AFM image is shown on the left (sum of squares = 2.27x10^7)and the error for the reconstruction using both the dull tip AFM image and SEM image is on the right (sum of squares = 3.95x10^7). The grayscale mapping to intensity values is identical for both images.

10.4 Synthetic Input Tests

This section describes the input AFM and SEM images for the synthetic data tests and the resulting reconstructed surfaces. The reconstructions using combined AFM and SEM images and AFM images alone are compared the ground truth.

10.4.1 Mimic of the NIST0523 Thick Example

10.4.1.1 Inputs

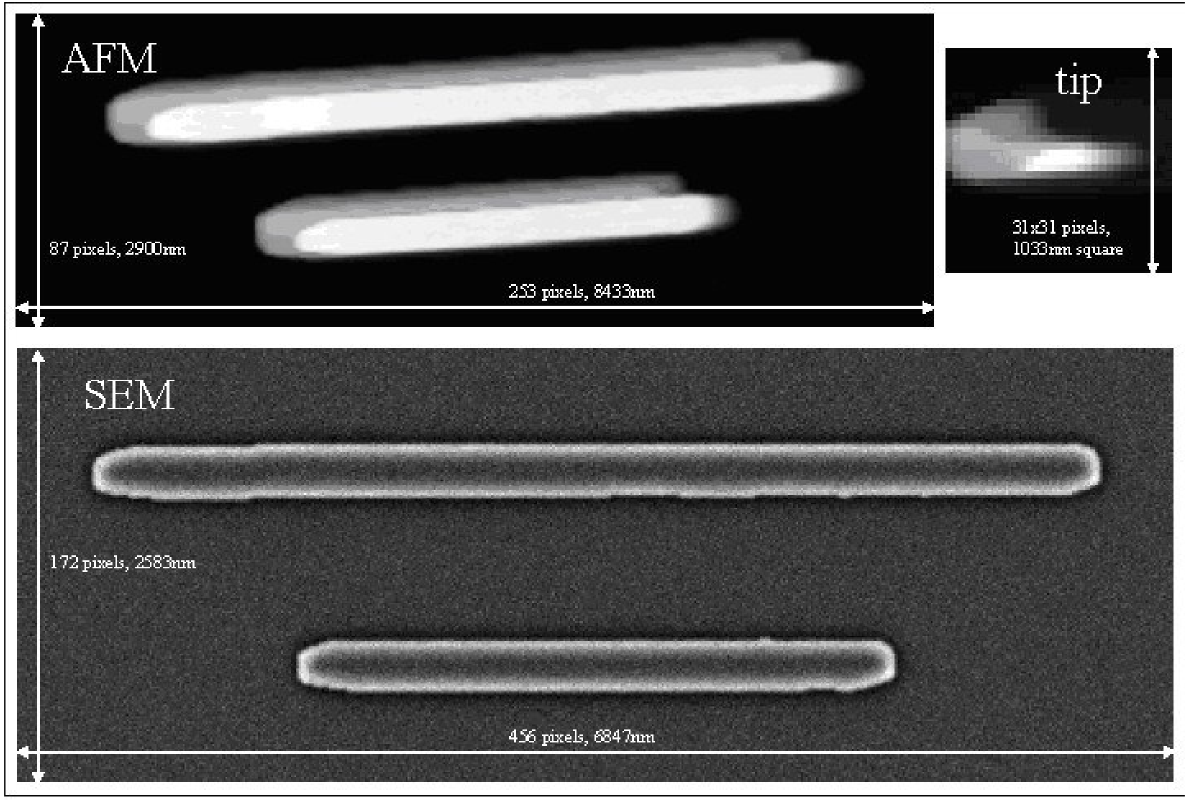


Figure 10-17: Inputs for the NIST0523_thick-based synthetic data test. The AFM tip was estimated by blind tip reconstruction from the AFM image.

10.4.1.2 Outputs

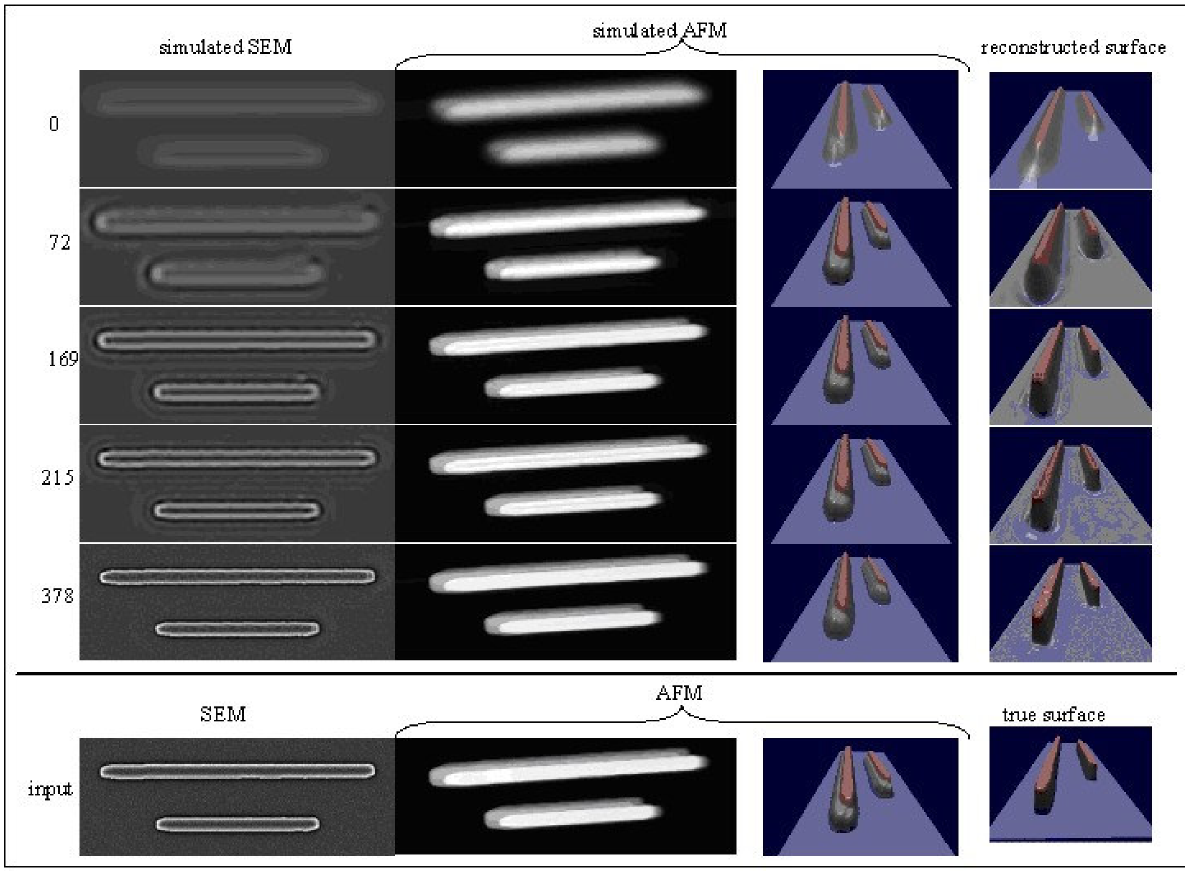


Figure 10-18: Selection of images from the reconstruction for NIST0523_thick-based synthetic example. The numbers on the left indicate the total number of conjugate gradient iterations. Excluding the bottom row, the columns from left to right are the simulated SEM image, the simulated AFM image, a 3D visualization of the simulated AFM image, and a 3D visualization of the reconstructed surface. The images in the bottom row from left to right are the input SEM image, input AFM image, 3D visualization of the input AFM image, and a 3D visualization of the "true" synthetic surface. This reconstruction required about 2.5 hours.

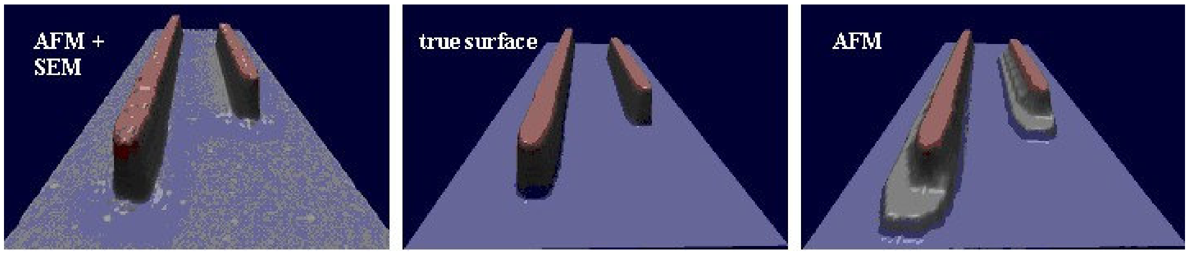


Figure 10-19: Comparison of the surface reconstructed by optimization of the likelihood computed from the AFM image and the SEM image (left), the true surface (center), and the surface found by grayscale erosion of the AFM image(right).

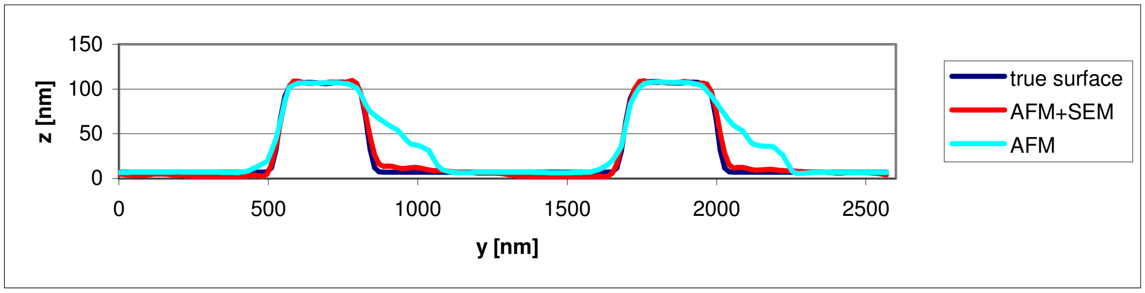


Figure 10-20: Comparison of cross-sections for the surfaces shown in Figure 10-19.

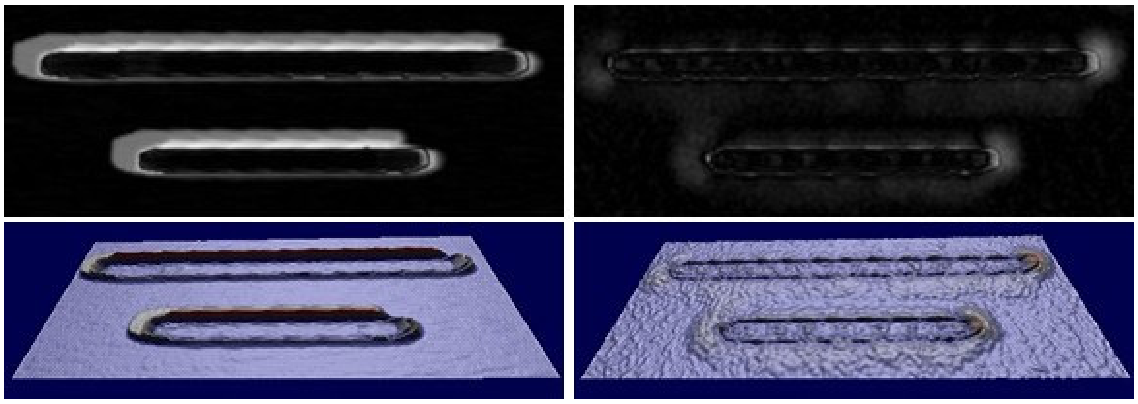


Figure 10-21: Absolute value of the difference in height between reconstructed surfaces and the ground truth surface. The error for the reconstruction using only the AFM image is shown on the left (sum of squares = 1.68x10^7) and the error for the reconstruction using both the AFM image and SEM image is on the right (sum of squares = 1.924x10^6). Each error image is shown as a grayscale image (using a common mapping from error value to grayscale value).

10.4.2 Mimic of the NIST0523 Thin Example

10.4.2.1 Inputs

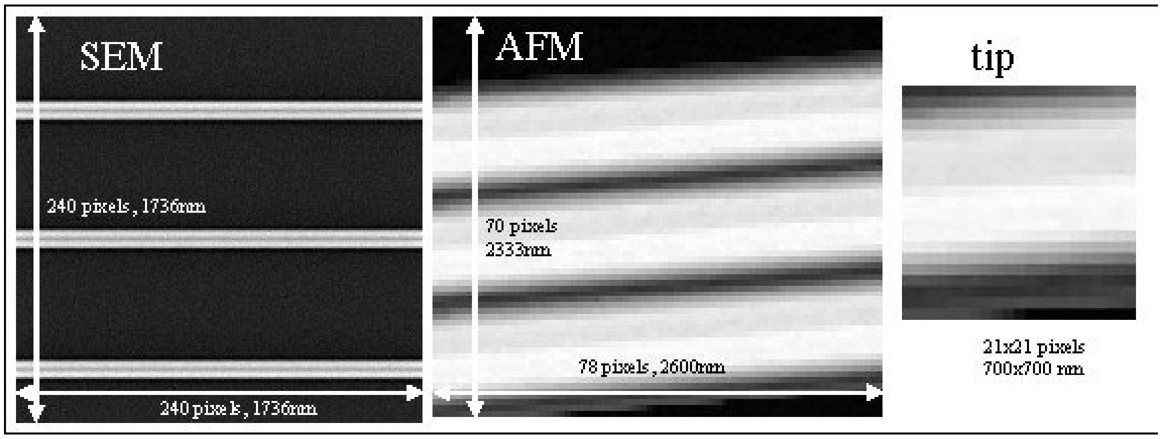


Figure 10-22: Inputs for the NIST0523_thin-based synthetic example. The AFM tip shape is the same as that found from the AFM image for the NIST052_thin example.

10.4.2.2 Outputs

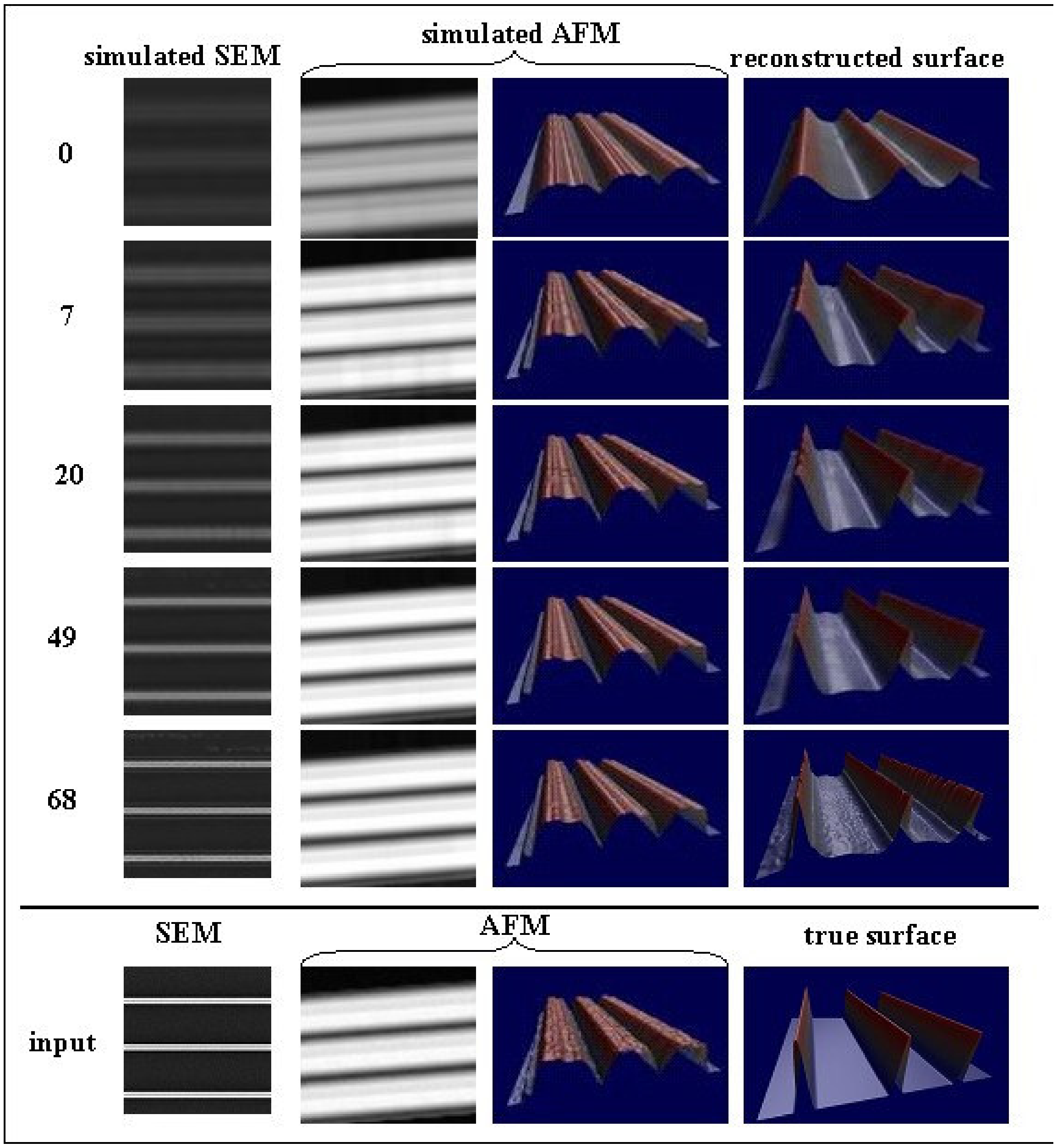


Figure 10-23: Selection of images from the reconstruction for NIST0523_thin-based synthetic. The numbers on the left indicate the total number of conjugate gradient iterations. Excluding the bottom row, the columns from left to right are the simulated SEM image, the simulated AFM image, a 3D visualization of the simulated AFM image, and a 3D visualization of the reconstructed surface. The images in the bottom row from left to right are the input SEM image, input AFM image, 3D visualization of the input AFM image, and a 3D visualization of the true surface. This reconstruction required about 3.5 hours.

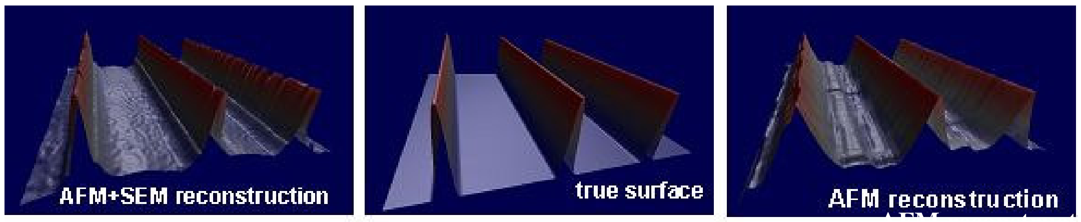


Figure 10-24: Comparison of the surface reconstructed by optimization of the likelihood computed from the AFM image and the SEM image (left), the true surface (center), and the surface found by grayscale erosion of the AFM image (right).

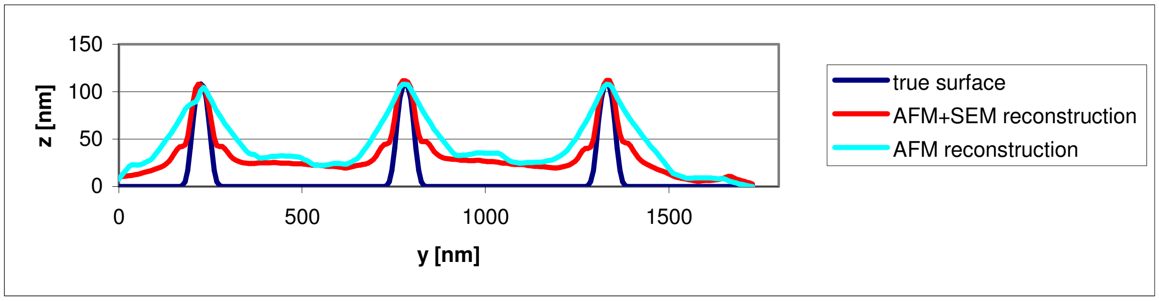


Figure 10-25: Comparison of cross-sections for the surfaces shown in Figure 10-24.

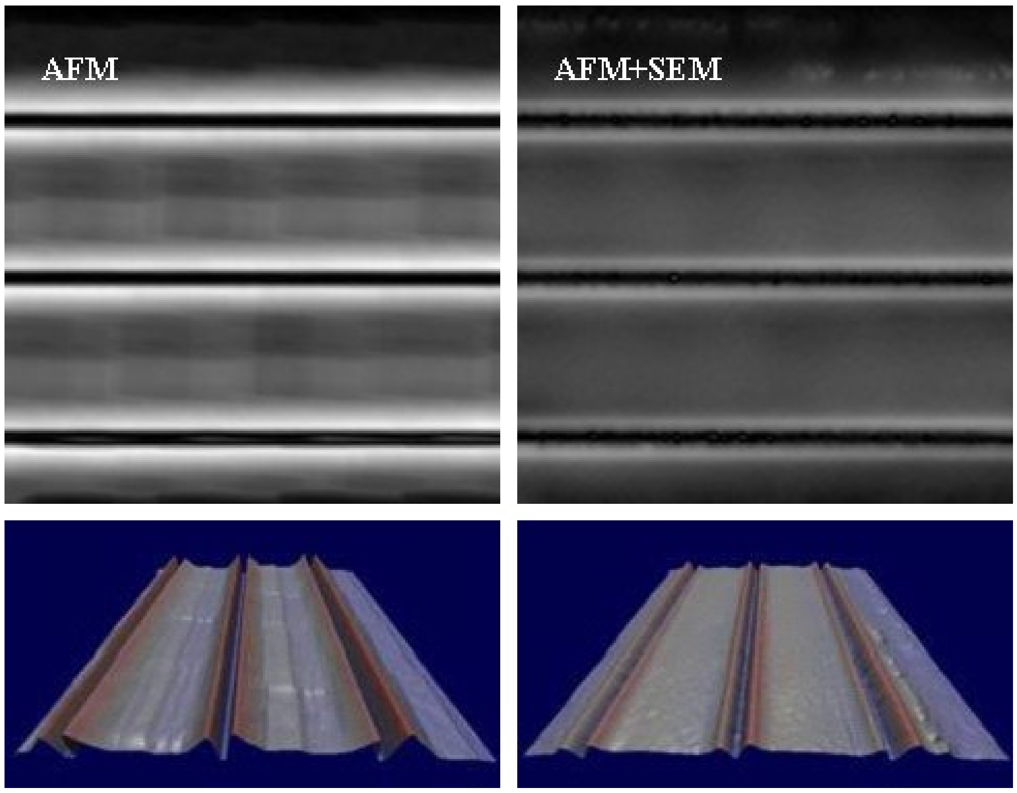


Figure 10-26: Absolute value of the difference in height between reconstructed surfaces and the ground truth surface. The error for the reconstruction using only the AFM image is shown on the left (sum of squares = 9.07x10^7) and the error for the reconstruction using both the AFM image and SEM image is on the right (sum of squares = 3.6x10^7). Each error image is shown as a grayscale image (using a common mapping from error value to grayscale value) and as a 3D visualization.

10.4.3 Synthetic Data Test 3

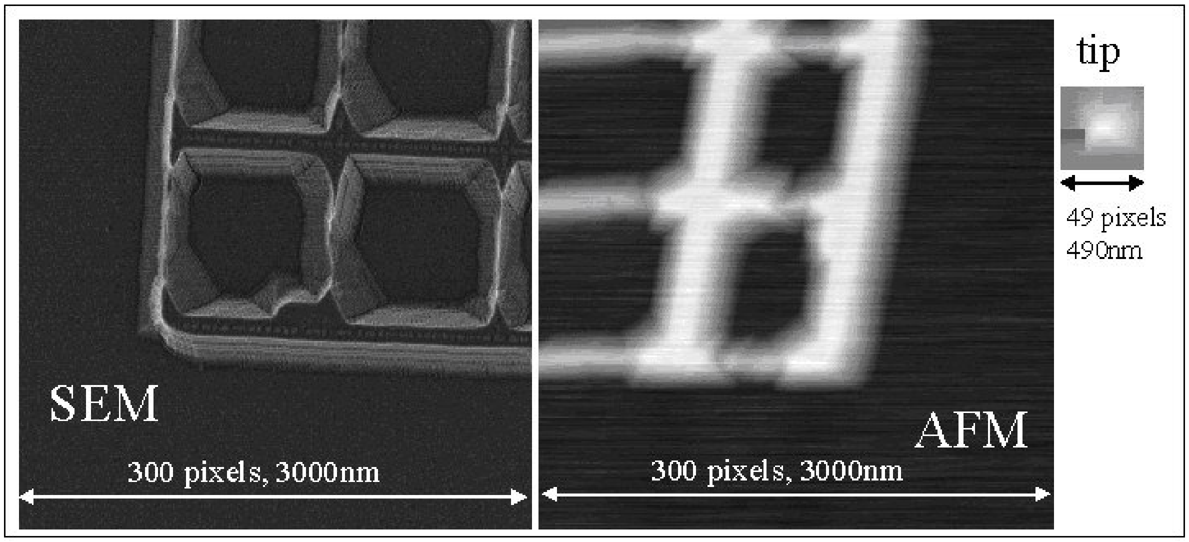


Figure 10-27: Inputs for synthetic test 3. The AFM tip was estimated by blind tip reconstruction from the AFM image.

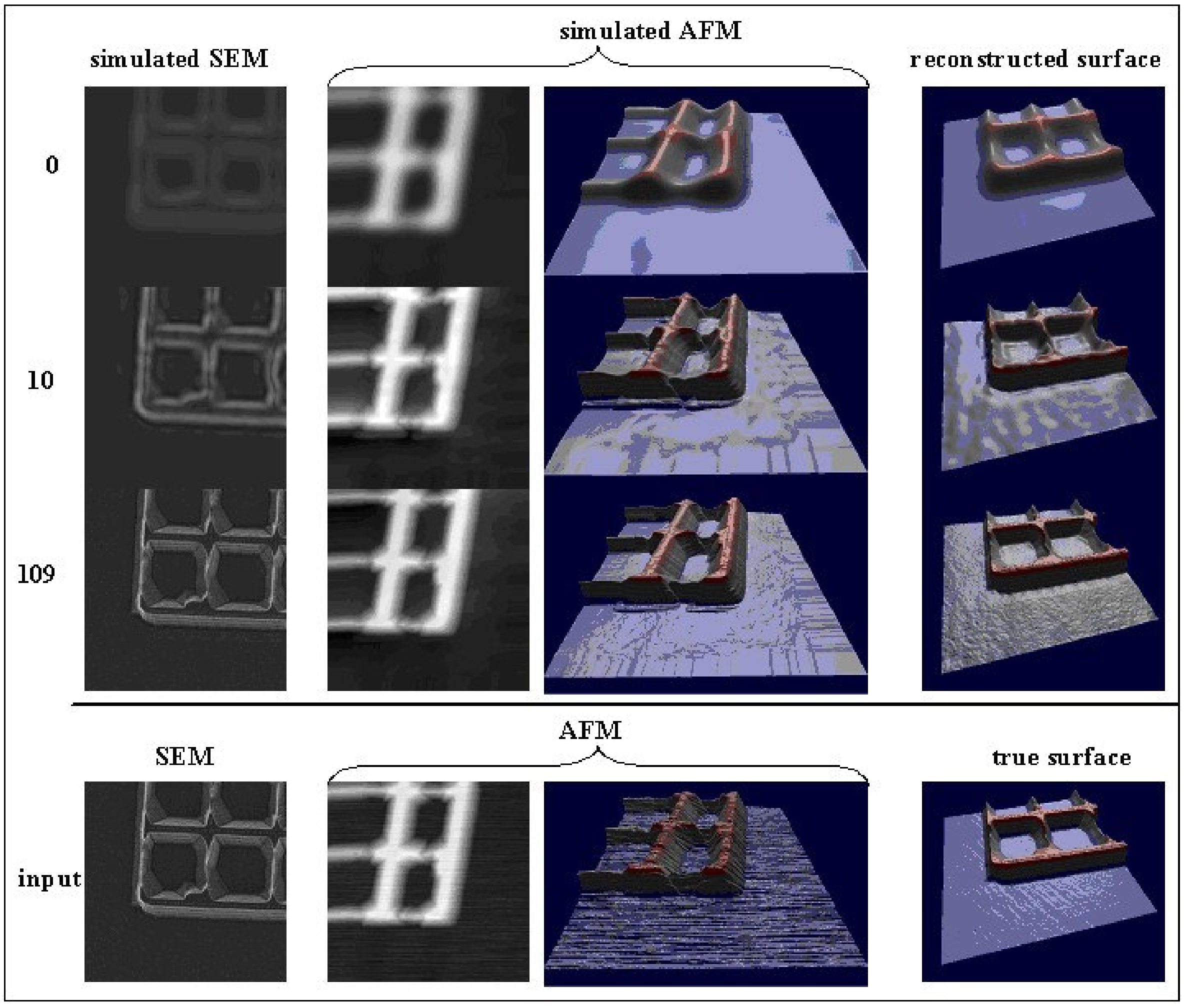


Figure 10-28: Selection of images from the reconstruction for synthetic test 3. The numbers on the left indicate the total number of conjugate gradient iterations. Excluding the bottom row, the columns from left to right are the simulated SEM image, the simulated AFM image, a 3D visualization of the simulated AFM image, and a 3D visualization of the reconstructed surface. The images in the bottom row from left to right are the input SEM image, input AFM image, 3D visualization of the input AFM image, and a 3D visualization of the "true" synthetic surface.

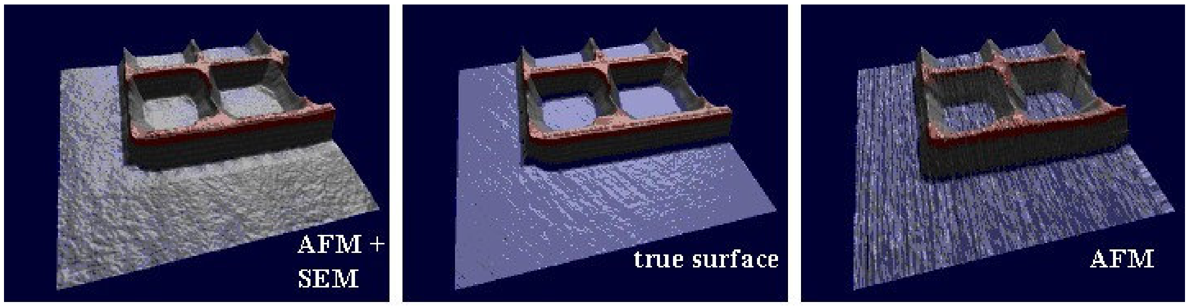


Figure 10-29: Comparison of the surface reconstructed by optimization of the likelihood computed from the AFM image and the SEM image (left), the true surface (center), and the surface found by grayscale erosion of the AFM image(right).

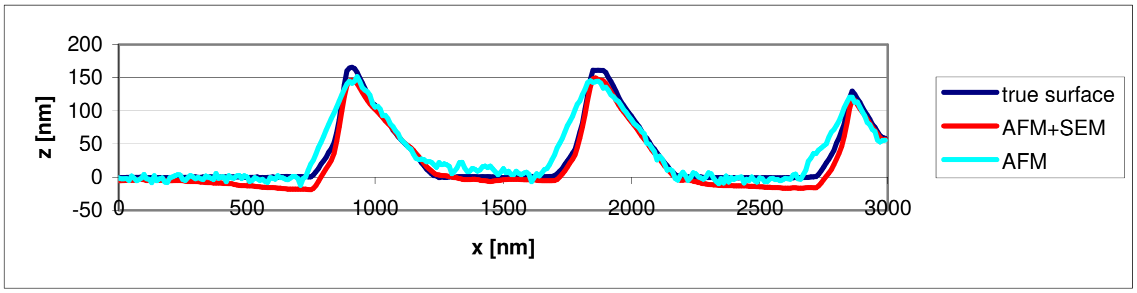


Figure 10-30: Comparison of cross-sections for the surfaces shown in Figure 10-29.

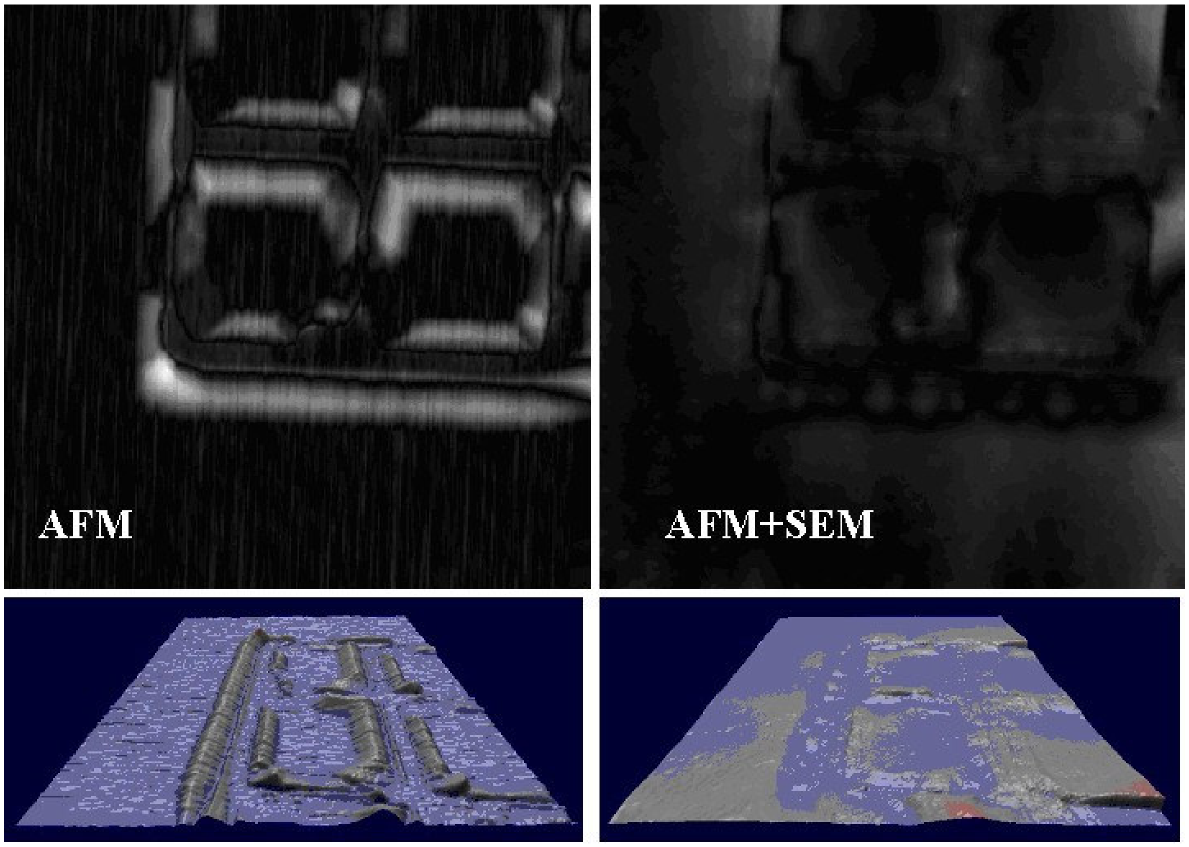


Figure 10-31: Absolute value of the difference in height between reconstructed surfaces and the ground truth surface. The error for the reconstruction using only the AFM image is shown on the left and the error for the reconstruction using both the AFM image and SEM image is on the right. Each error image is shown as a grayscale image (using a common mapping from error value to grayscale value) and as a 3D visualization.

10.4.4 Synthetic Data Test 4

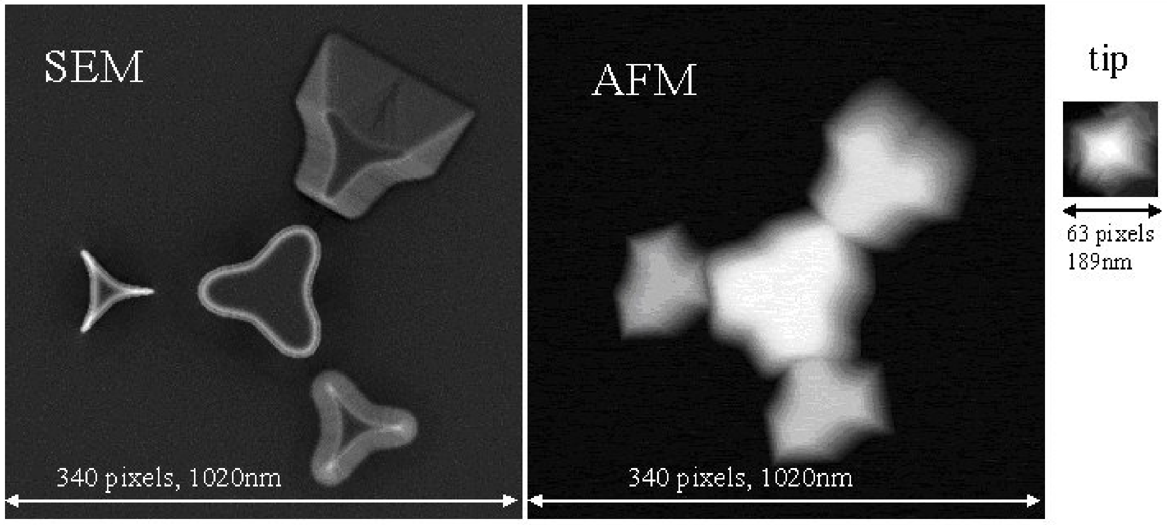


Figure 10-32: Inputs for synthetic test 4. The AFM tip was estimated by blind tip reconstruction from the AFM image.

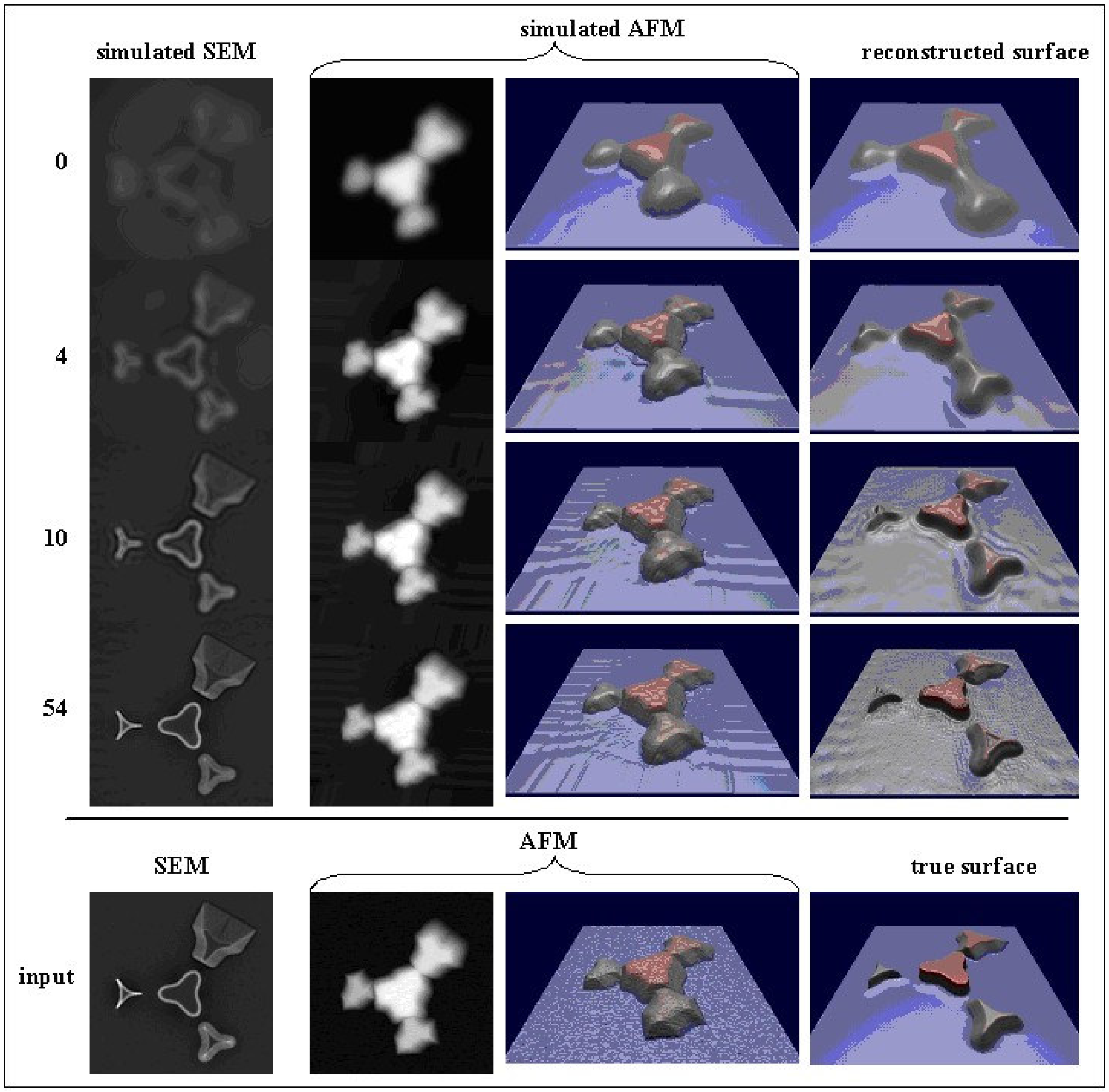


Figure 10-33: Selection of images from the reconstruction for synthetic test 4. The numbers on the left indicate the total number of conjugate gradient iterations. Excluding the bottom row, the columns from left to right are the simulated SEM image, the simulated AFM image, a 3D visualization of the simulated AFM image, and a 3D visualization of the reconstructed surface. The images in the bottom row from left to right are the input SEM image, input AFM image, 3D visualization of the input AFM image, and a 3D visualization of the "true" synthetic surface.

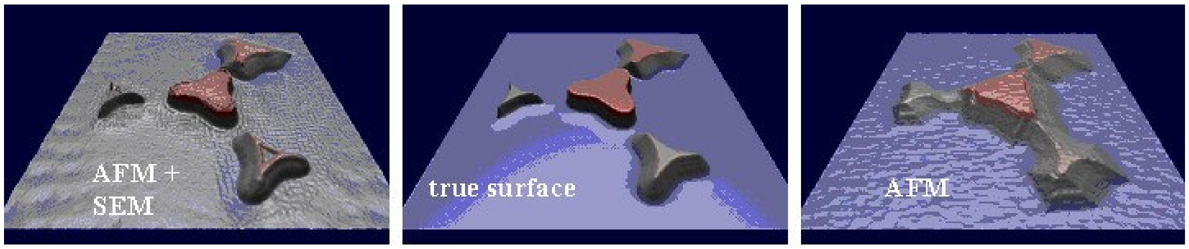


Figure 10-34: Comparison of the surface reconstructed by optimization of the likelihood computed from the AFM image and the SEM image (left), the true surface (center), and the surface found by grayscale erosion of the AFM image(right).

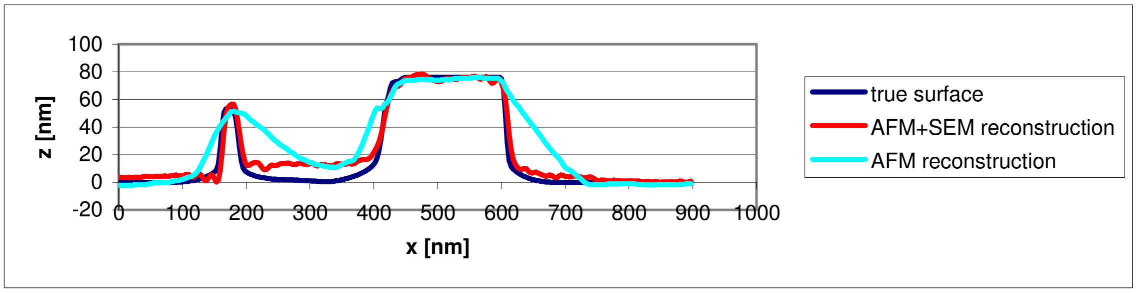


Figure 10-35: Comparison of cross-sections for the surfaces shown in Figure 10-34.

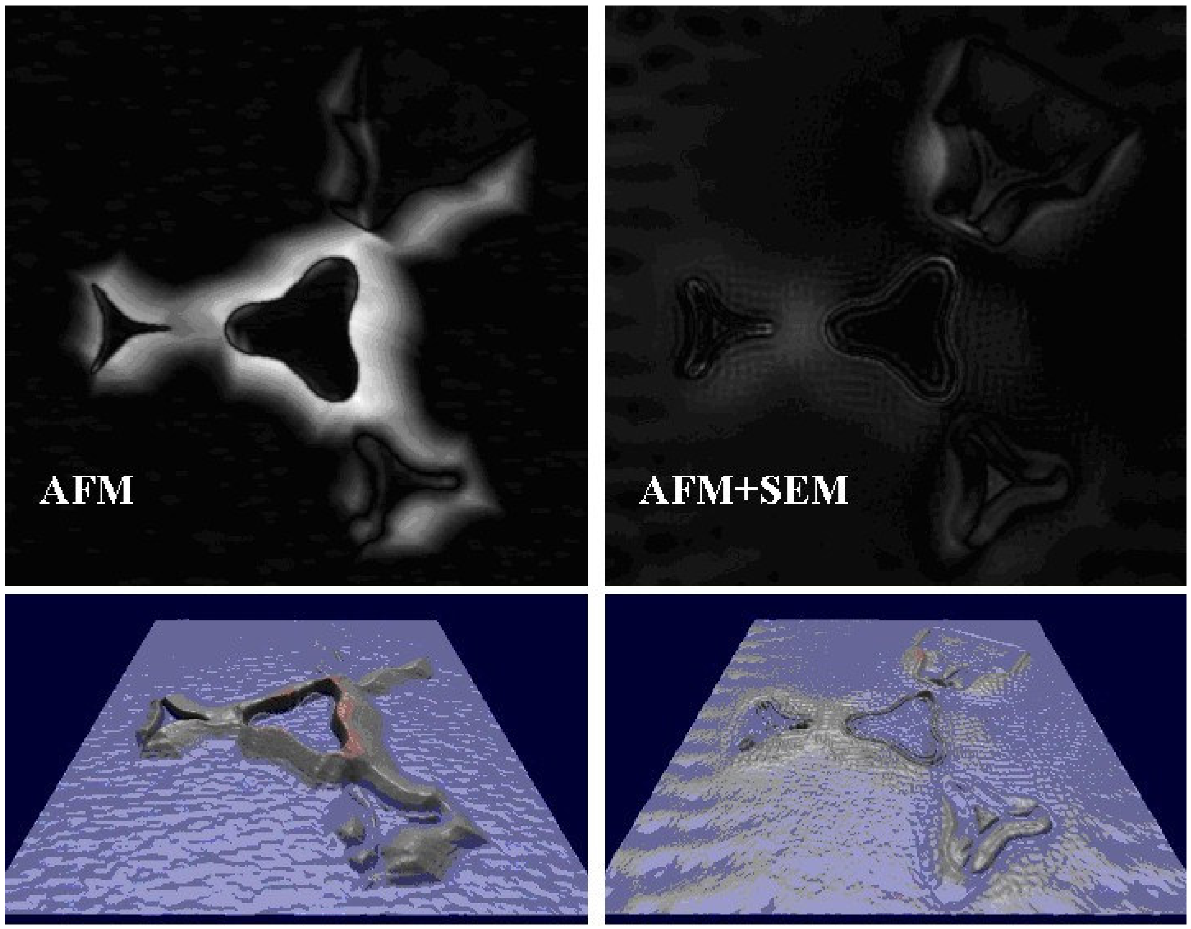


Figure 10-36: Absolute value of the difference in height between reconstructed surfaces and the ground truth surface. The error for the reconstruction using only the AFM image is shown on the left and the error for the reconstruction using both the AFM image and SEM image is on the right. Each error image is shown as a grayscale image (using a common mapping from error value to grayscale value) and as a 3D visualization.

10.4.5 Synthetic Data Test 5

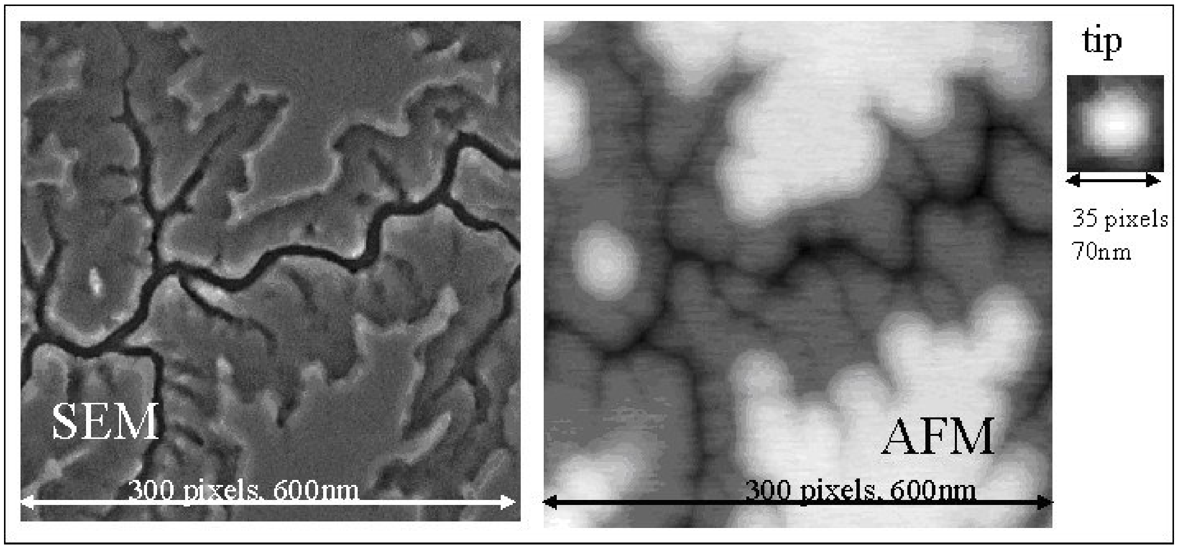


Figure 10-37: Inputs for synthetic test 5. The AFM tip was estimated by blind tip reconstruction from the AFM image. The surface used to generate these inputs is based on terrain data of the Grand Canyon from the U.S. Geological Survey (USGS) processed by Chad McCabe and available from http://www.cc.gatech.edu/projects/large_models/gcanyon.html.

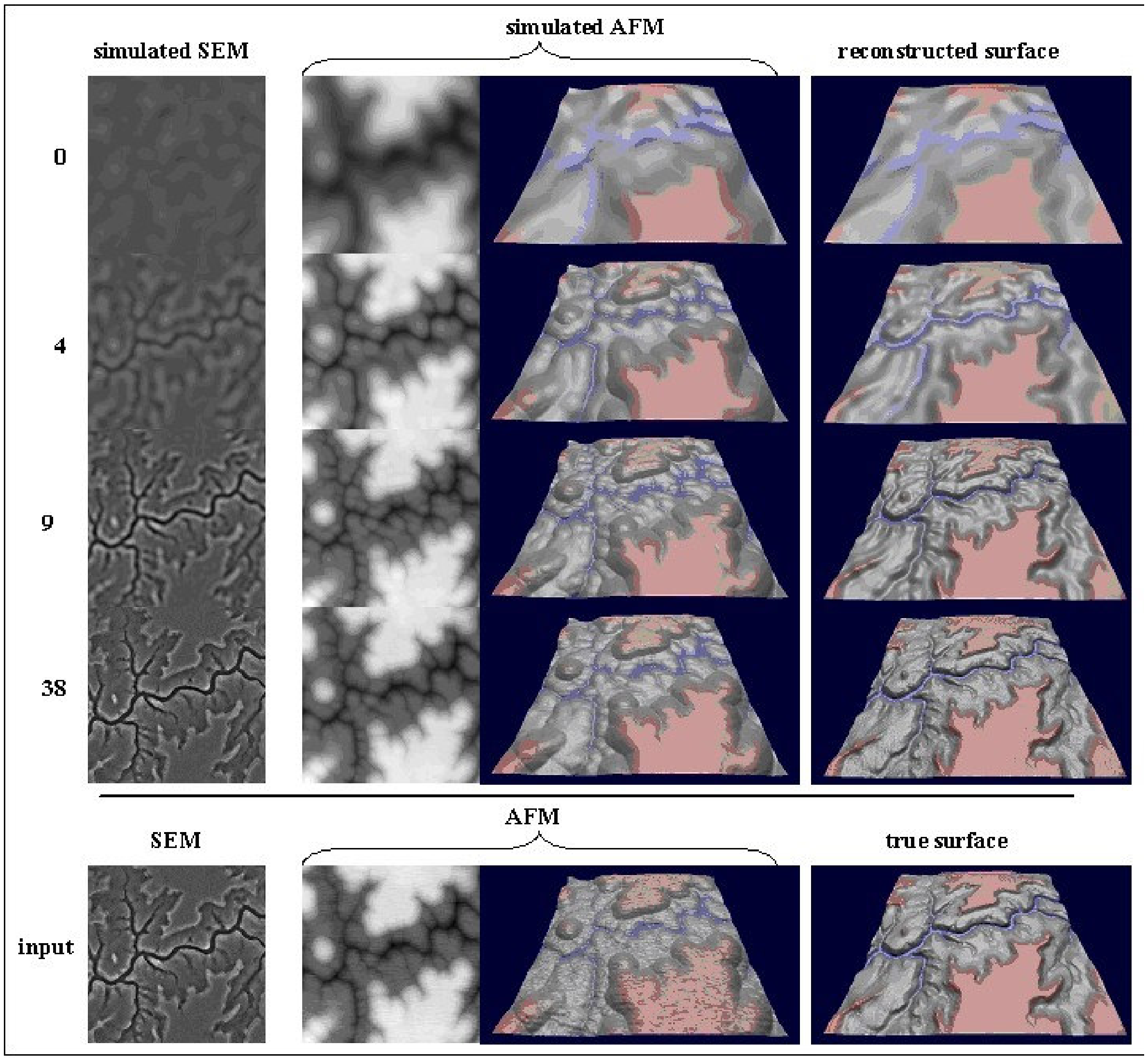


Figure 10-38: Selection of images from the reconstruction for synthetic test 5. The numbers on the left indicate the total number of conjugate gradient iterations. Excluding the bottom row, the columns from left to right are the simulated SEM image, the simulated AFM image, a 3D visualization of the simulated AFM image, and a 3D visualization of the reconstructed surface. The images in the bottom row from left to right are the input SEM image, input AFM image, 3D visualization of the input AFM image, and a 3D visualization of the "true" synthetic surface.

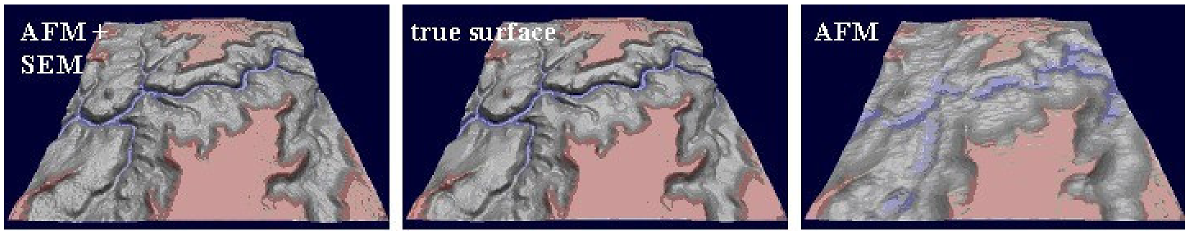


Figure 10-39: Comparison of the surface reconstructed by optimization of the likelihood computed from the AFM image and the SEM image (left), the true surface (center), and the surface found by grayscale erosion of the AFM image(right).

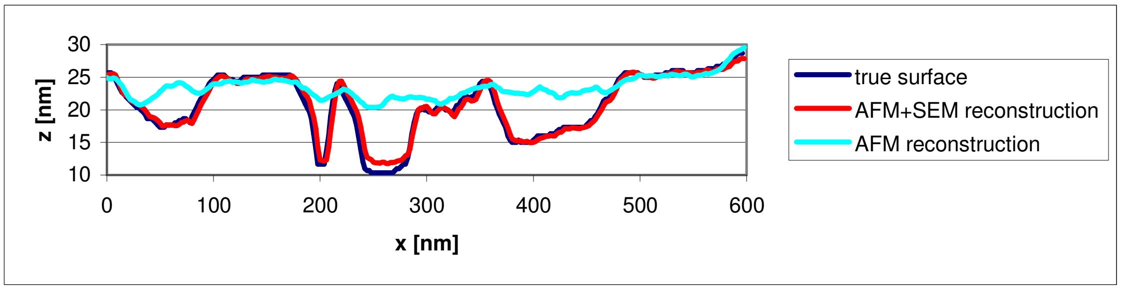


Figure 10-40: Comparison of cross-sections for the surfaces shown in Figure 10-39.

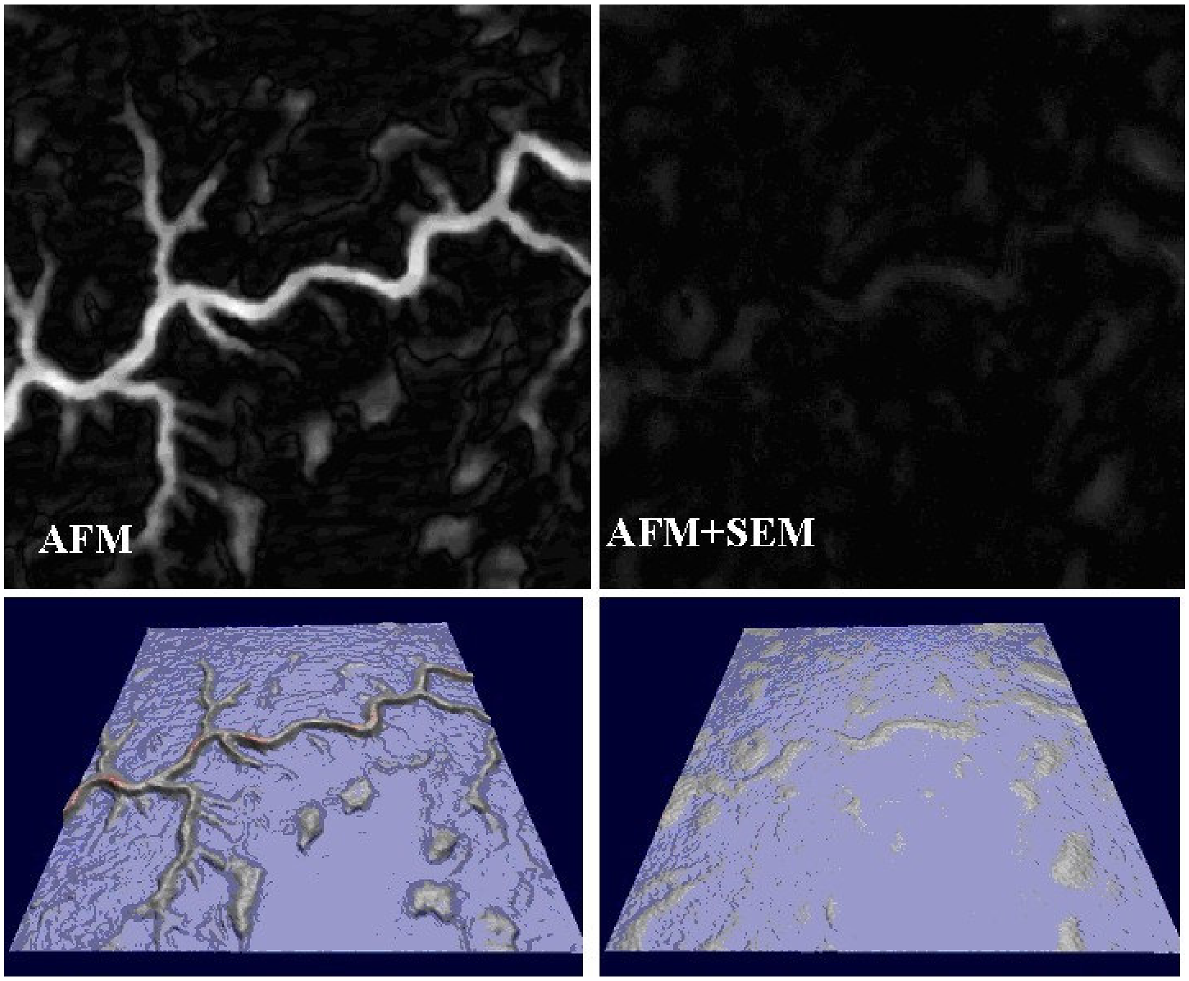


Figure 10-41: Absolute value of the difference in height between reconstructed surfaces and the ground truth surface. The error for the reconstruction using only the AFM image is shown on the left and the error for the reconstruction using both the AFM image and SEM image is on the right. Each error image is shown as a grayscale image (using a common mapping from error value to grayscale value) and as a 3D visualization.

10.5 Summary of Results

The estimated and actual errors for the reconstructions along with the time required for the AFM/SEM-based reconstruction on an 1.3GHz Intel Pentium M processor are listed in Table 10-1.

test data set	AFM reconstruction error	AFM/SEM reconstruction error	AFM/SEM run time
real data test 1	15.7e6	3.19e6	3.5 hours
real data test 2	22.7e6	39.5e6	2 hours
synthetic data test 1	16.8e6	1.92e6	4 hours
synthetic data test 2	90. 7e6	36.0e6	6 hours
synthetic data test 3	22.5e6	9.32e6	6 hours
synthetic data test 4	13.8e6	1.83e6	4 hours
synthetic data test 5	0.536e6	0.0261e6	3.5 hours

Table 10-1: Comparison of estimated and actual errors. For the two real data tests the error value is the sum of squared differences between the reconstruction and the AFM-based reconstruction using a sharper AFM tip. For the synthetic data tests the error is the sum of squared differences between the reconstruction and the true surface.

10.6 Discussion of Results

All five of the tests with synthetic data (where ground truth is known) indicated that the SEM/AFM reconstruction had less than 50% of the error of the AFM-only reconstruction. The real-world NIST0523_thick data set using my SEM/AFM method had 20% of the error of an AFM-only reconstruction. None of these surfaces were amenable to SEM-only reconstruction using existing techniques.

The second NIST0523_thin data set indicated more error in the SEM/AFM reconstruction than the AFM-only reconstruction. I believe that this represents a failure of the test procedure rather than an indication of the algorithm's performance. The major problem here is that inaccuracy in the sharp tip AFM-based reconstruction makes the estimate of the difference between the reconstruction and the true surface innacurate. If the sharp tip and dull tip AFM images are too close, as the reconstruction gets closer to the true surface, the estimated error will increase instead of decreasing as it should.

Another problem with the NIST0523_thin data is that the SEM image shows some asymmetry in the shading that is interpreted under the assumption of a circularly symmetrical shading model as an asymmetry in the line shapes. This assymmetry might be caused by surface charging and I am assuming such charging doesn't occur. As the SEM scans over the lines they will tend to charge positively (Monte Carlo simulations predict more electrons are ejected from such topography than are injected by the beam). In this scenario, initially a line would look very bright but as charge is removed making the line positively charged it would become more difficult for electrons to escape making the second edge darker than the first. After passing over a line there will be a strong attraction of escaping electrons towards the positively charged line and the surface will appear somewhat darker than usual. By the time the beam has reached the next line the previous line will have become less charged as secondary electrons from other parts of the surface were pulled into it and the next line will initially look bright again. Another possibility is that there is assymmetry in the detector, specimen chamber, or electric fields in the chamber and this makes it easier for electrons escaping towards the top of the image to be detected than electrons escaping towards the bottom. A vertical line of the SEM image is plotted in Figure 10-42. Diagrams for the two possible explanations are provided in Figure 10-43. Additional experiments would be necessary to determine the source of this anomaly. If the effect were due to charging then it might be eliminated by using a lower accelerating voltage or faster scan rate.

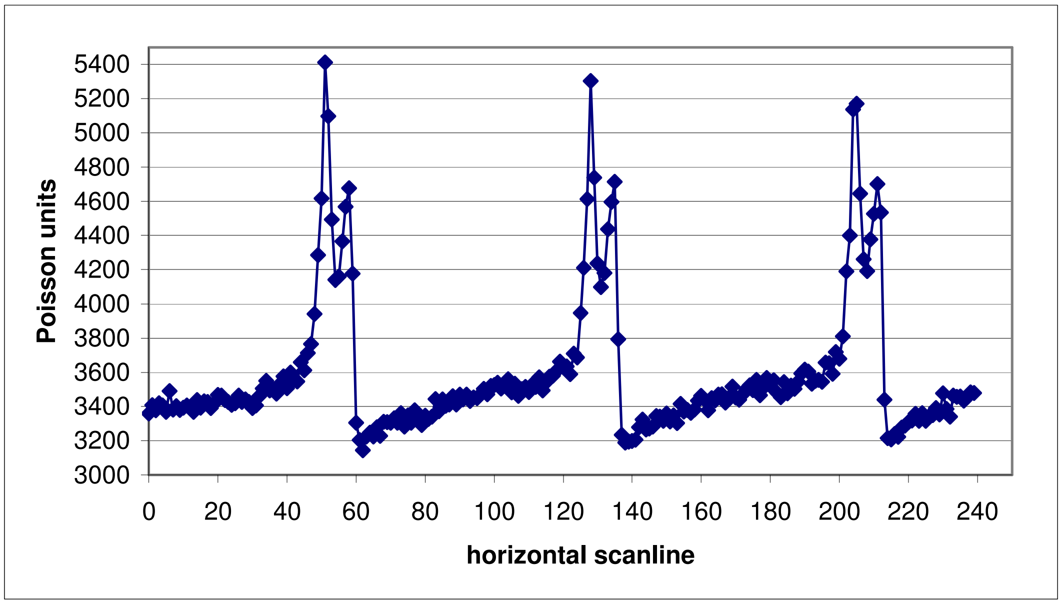


Figure 10-42: Plot of vertical line from Figure 10-3d showing that edges towards the top of the image (scanline 0) appear brighter than those towards the bottom (scanline 240).

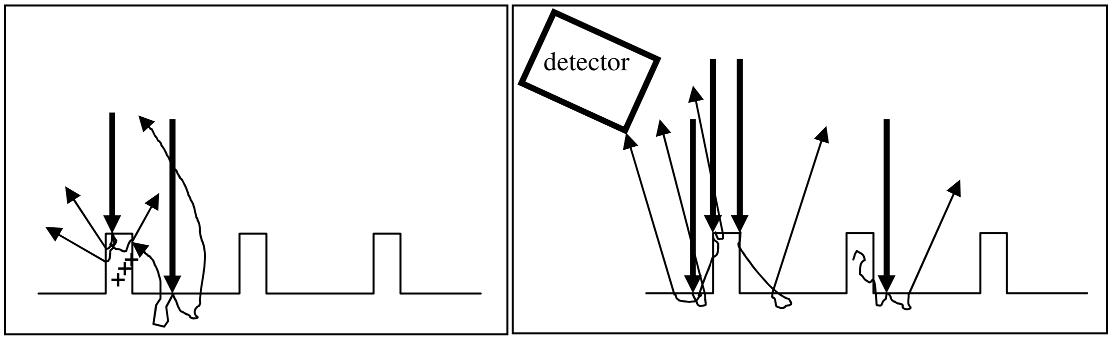


Figure 10-43: Two possible explanations for asymmetry in the shading of the NIST0523_thin SEM image. (left) Positive charging of each raised feature causes secondary electrons to be pulled back towards the surface reducing the number of electrons that are detected. (right) Offcenter position of the detector or greater sensitivity to electrons escaping to the left side of the surface in conjunction with blocking of electrons traveling to the left by the raised features causes more electrons to be detected escaping near the left side of the raised feature than the number detected escaping near the right side.

11 Results and Future Work

11.1 Results

This document describes several new contributions to multi-modality image analysis applied to SEM and AFM including an objective function that defines an optimal surface reconstruction given AFM and SEM images of the same area on the surface. This objective function is based on models for image formation (chapters 4, 5, and 6) with both correlated and independent noise (chapter 8). I have demonstrated that highly detailed physically-based models for image formation can be used to relate two very different image modalities. Though this idea is not entirely new, the physical models used are significantly more complex than those previously used.

Chapter 6 showed that in the case of specimens described by a height field and composed of a single material, filter bank synthesis is a much more efficient way to simulate SEM images than Monte Carlo simulation. Monte Carlo SEM simulation as described in chapter 4 is many orders of magnitude too slow for an iterative optimization method. Besides being much faster, filter bank synthesis also has the advantage over Monte Carlo simulation of having an analytic derivative and not being noisy. Though the existing models may be based on simulations that take too long to be practical (e.g. Monte Carlo simulation), this thesis shows that in the case of SEM images the forward model can be approximated by a simpler model that is fast and accurate enough to be part of an iterative optimization that computes a reconstruction in a few hours.

Chapter 9 described an objective function for fusing AFM and SEM data based on a filter bank model for SEM and the existing morphological dilation model for AFM. Using a surface model introduced in [Jones94] the gradient of this objective function can be computed with respect to the surface parameters. Examples in chapter 10 show how the objective function can be optimized using the conjugate gradient method to reconstruct a specimen surface. This method could be applied to either AFM or SEM images in isolation as well as to combination AFM-SEM images.

Chapter 7 showed that knowledge of the distortions present in each image modality can also be used to improve the registration of two different image types. Using erosion to reduce the AFM artifact produces a model of the underlying specimen. Applying the forward SEM simulation to this model produces an image that is of a form that can be directly aligned with the real SEM image using correlation-based methods. The inverse AFM transform and forward SEM transform quantitatively apply the relevant distortions.

The results described in chapter 10 support the thesis that my method for surface reconstruction using combination AFM/SEM images is more accurate than existing methods for reconstruction using only AFM images. The increase in accuracy gained by using the SEM data depends on the sharpness of the AFM tip and the specimen topography. The more sharp jumps in height or steep sidewalls on the surface and the more dull the AFM tip, the more room there will be for improvement so it is not possible to provide a single estimate of the increase in accuracy. For the tests I performed using data believed to be representative of real AFM and SEM images, my error metric (sum of squared differences in height) was typically reduced by at least a factor of 2. The time required for a reconstruction (about 2-5 hours for the test examples) is many orders of magnitude less than what would be required if the Monte Carlo SEM model were used in place of the filter bank model.

11.2 Limitations and Future Work

The filter bank model used to model SEM image formation is limited to images that can be expressed as a linear combination of filters measuring slope and curvature. Some modalities that can probably be handled in the same way as SEM are scanning ion microscope (SIM) images and laser scanner backscatter intensity images for material with subsurface scattering (such as marble). Filter bank synthesis is a more general approach to modeling image formation and could be useful for other modalities assuming one selects an appropriate set of basis filters and function for combining the outputs of those filters. The idea of using detailed physical models for specimen reconstruction from images could also be generalized to other modalities such as images formed by visible light, magnetic resonance imaging (MRI), x-rays and computed tomography (CT), ultrasound, radar, positron emission tomography (PET), and laser scanners.

The true surface must be representable as a height field and this means that in general, surfaces with undercuts cannot be reconstructed. The SEM can sample below the surface of a specimen because electrons penetrate some distance before escaping and being detected. Also, electrons escaping from one surface may interact with other surfaces that are hidden from the incident electron beam. Thus there may be some information in an SEM image that is representative of such hidden surfaces. Allowing hidden surfaces is likely to make the reconstruction problem severely under-constrained. However, if it is known that hidden parts of the surface are constrained such that they are predictable from the visible surface, then the actual surface might be deduced from the reconstructed visible surface. For example, the profile of undercut edges might be the same everywhere, a reasonable assumption for surfaces generated by some etching processes.

A related limitation is that the SEM image is an overhead view so the AFM and SEM image planes are approximately parallel. This is not a serious limitation because if there were a more general 3D rotation between the AFM and SEM coordinate systems, given the previous limitation, the only change that would be required in the algorithm would be a change in the way the surface is resampled into the AFM coordinate system before computing the simulated AFM image. A fully three-dimensional representation would be useful, particularly if images with different viewing directions were available.

The algorithm as implemented for this project is limited to specimens composed of a single material. This limitation simplifies the problem of modeling the SEM images because different materials will in general appear differently in the SEM and one would need to either be given or determine as part of the reconstruction the distribution of the different materials in the specimen. This algorithm could be extended to handle specimens composed of multiple materials by extending the filter bank model to include filters that respond differently to different material types. A material map could be provided by an x-ray detector or considered as a free parameter of the specimen model. For manufactured samples consisting of multiple materials it is common for the different materials to be arranged in layers where each layer is contained in a unique range of heights. In reconstructing such a sample, one could use the height as a stand-in for knowing the material. The height of the surface model would automatically translate into a different material and this would essentially switch the behavior of the SEM model. Accurate simulation of the SEM image at material boundaries could still be a challenge but this may not be critical or there may be a simple interpolation scheme that would work.

For applications involving manufactured surfaces, there is usually some prior knowledge (such as CAD data) that could be incorporated into the algorithm to improve accuracy. CAD data could also be used in place of AFM images. A simulation of the manufacturing process, including etching and deposition processes, might be used to simulate a set of surfaces that sample the space of process parameters. From the set of simulated surfaces, one could generate a library of simulated images similar to that used in [Davidson99]. Then a real image could be matched to images in the library to determine the parameters and estimate the shape of a real surface.

I assume no charging effects in the SEM images but specimen charging typically occurs for specimens composed of an insulating material. Some SEMs reduce charging effects by introducing a gas into the specimen chamber or by placing a conducting grid just above the specimen but these are not commonly available. The charging results in electric fields that modify the trajectories of electrons and can create complicated variations in intensity. In order to overcome this limitation, it would be necessary to model the charging in an efficient way. Some subtle variations in intensity due to charging can be compensated for by the calibration procedure (determining the best fit combination of simulated SE and BSE images to match the real SEM image) and [Davidson99] describes a more flexible calibration method that might be used to reduce sensitivity to charging effects but this method is also restricted to subtle charging effects. Existing methods for simulating charging effects require an impractical amount of time so, unless a fast approximation can be developed, it is difficult to see how my reconstruction approach could work in the case of severe charging.

The AFM image is assumed to not have significant feedback artifacts and the rate of drift between the scanner and surface coordinates is assumed to be constant. Feedback artifacts might be handled by taking into account the feedback signal indicating the amount of deflection of the AFM cantilever. The feedback signal would be converted into a displacement correction and added to the topography image. This scheme would also enable faster scanning which would reduce the effect of drift. The shearing transformation caused by a constant rate of drift is accurately modeled by an affine transformation but if the rate of drift varies significantly, a more general transformation will be needed. It is reasonable to assume some temporal coherence in the drift velocity vector. I would represent the drift velocity vector as a smoothly varying function of time that would transform each pixel using the integrated velocity vector according to the time at which it was acquired.

The specimen is assumed to not change in between acquiring the AFM and SEM images. In some cases, imaging with the AFM or SEM may modify a specimen, causing parts to break or depositing a contamination layer. Also, a specimen may be damaged or contaminated in storage between images taken with different tools. My algorithm assumes such things do not happen. It may be possible to detect and flag inconsistencies between the two images due to such modification but my algorithm does not try to do this. Such a feature might also be useful for defect analysis in manufacturing applications, comparing a design specification or control image with an SEM or AFM image of the surface being analyzed.

The SEM shading is assumed to be rotation invariant. This is not a strict requirement of the method but in the described implementation, the SEM intensity is assumed to be a function of rotation invariant shape characteristics: gradient magnitude and the two principal curvatures estimated over a number of local neighborhood sizes. The assumption of rotation invariance is reasonable given the lack of any information about the specific detector geometry in the SEM. A secondary electron detector approximately measures SE escaping directly from the surface in all directions and BSE escaping in all directions that have generated additional SE through interactions with the walls of the specimen chamber. The detector may be more sensitive to electrons escaping in a particular direction due to asymmetry in the detector geometry, specimen chamber or electric field geometry but this information was not readily available. If a specimen with a known shape were available, it could be used to characterize the asymmetry in the SEM shading. Instead of training the filter bank model from a Monte Carlo simulation, the model could be trained from an actual SEM image of the known shape.

Determining the height of the surface from an SEM image alone is expected to be an under-constrained problem. Even in the simpler case of shape-from-shading it has been proven that the height of certain critical points (local minima and maxima) on the surface must be known in order to sufficiently constrain the problem. In my case, the AFM must provide the sufficient height constraints, touching the surface at certain critical points that properly constrain the reconstruction. I have not determined what would be a sufficient set of constraints but it is reasonable to expect similar requirements to those for shape-from-shading. The AFM tip will usually touch most local maxima of the surface but can often be prevented from touching local minima if there is a hole that is too narrow for the tip to touch the bottom before it hits the sides. While the resulting reconstruction will be better than either the SEM or AFM-based reconstruction alone, if the AFM tip doesn't touch the surface near the bottom of a pit or groove, there is likely to be significant error in the height in the corresponding part of the reconstruction.

Errors in registering the AFM and SEM images, in estimating the tip shape or in estimating SEM brightness/contrast settings will introduce errors in the surface reconstruction. Sarang Joshi has suggested that the transformation between AFM and SEM images and the surface reconstruction be optimized in a more integrated fashion to increase the accuracy. Because the AFM tip shape model has an effect on the optimal transformation, the accuracy might be increased by reconstructing the AFM tip shape in parallel with determining the transformation.

In my opinion, the most important obstacle to making this approach practical is that SEMs are not designed to make quantitative measurements. This is reflected in the fact that while the SEM detector amplifier settings may be manually set to adjust brightness and contrast, the actual values are not output by the control software making it very difficult to convert images from an SEM into meaningful physical units for later analysis. Images taken at different times typically will usually have different amplifier settings but without knowing these parameters it is difficult to compare them. Also, characterization of the detector sensitivity to electrons escaping in different directions with different energies is not typically available making it difficult to relate the output of a Monte Carlo simulation to a real image. This is only a technical problem and one could probably solve it by modifying an SEM to read out the amplifier settings and by doing many careful measurements of calibration specimens to characterize the detector. It would be nice to develop a calibration procedure that would work on multiple SEMs enabling quantitative analysis of image contrast and comparison of images across different tools.

Appendix A AFM tip-specimen interactions

The interaction potential for a pair of atoms is approximated by the Lennard-Jones potential ([Sarid94], p 189):

$$w(r) = 4w_0\left(\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right)$$

where r is the distance between the atoms and w_0 and σ are chosen to fit the physical properties of the material. The $\frac{1}{r^{12}}$ term represents a repulsive force which dominates at short distances due to overlapping electron orbits and the Pauli exclusion principle. The $\frac{1}{r^{6}}$ term comes from attractive Van der Waals forces in which the charge distribution of one atom creates a dipole moment in another atom. The minimum energy is $-w_0$ which occurs at a distance of approximately 1.12σ. A graph of the potential is shown in Figure A-1.

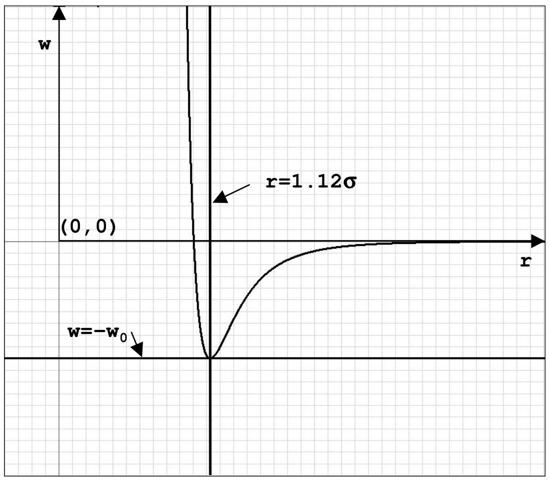


Figure A-1: Lennard-Jones potential

The actual potential describing the tip-specimen interaction can be estimated by summing the Lennard-Jones potential over all atom-atom pairs. For a spherical tip, a flat plane as shown in Figure A-2, and an intermolecular potential modeled as C/r^n the integration results in a total potential of the form (Sarid, p 192):

$$w(z) = \frac{4\pi^2 \rho_1 \rho_2 C R}{(n-2)(n-3)(n-4)(n-5)} \cdot \frac{1}{z^{n-5}}$$

Equation A-1

where ρ_1 and ρ_2 are the densities of the tip and plane, R is the radius of the tip, and z is the tip-plane separation distance.

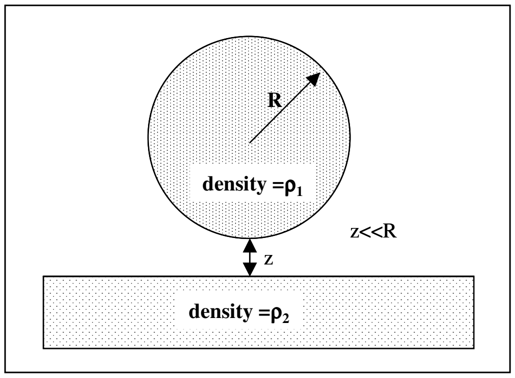


Figure A-2: Interaction between sphere and plane can be computed by integrating the Lennard-Jones potential over all atom-atom pairs

We can separate the Lennard-Jones potential into two terms with the form C/r^n :

$$w_6(r) = \frac{-4w_0\sigma^6}{r^6} = \frac{C_6}{r^6} \qquad w_{12}(r) = \frac{4w_0\sigma^{12}}{r^{12}} = \frac{C_{12}}{r^{12}}$$

Integrating these two terms using Equation A-1 we get a total potential of the form ([Sarid94], p 211):

$$w(z) = -\frac{2\pi^2 w_0 \rho_1 \rho_2 \sigma^6 R}{3z}\left[1 - \frac{1}{210} \cdot \frac{\sigma^6}{z^6}\right]$$

When imaging with the AFM in contact mode, the interaction between the tip and surface is in the repulsive regime ($\partial w/\partial z < 0$ or $z < \sqrt[6]{\sigma/30}$) where the potential increases very rapidly with decreasing distance. In true non-contact mode, the interaction would ideally stay in the attractive regime ($\partial w/\partial z > 0$). However, in tapping mode or when using non-contact mode in air, the tip will typically jump between the attractive and repulsive regimes as it oscillates and most of the reduction in oscillation amplitude as the tip is lowered towards the surface is due to repulsive forces. A water layer wets the surface when it is exposed to air and the surface tension of the water helps to pull the tip into contact with the surface. Even when the AFM is operated in non-contact mode in ultra high vacuum to avoid the interference of a water layer on the surface, it is considered likely that when the tip is nearest to the surface there is some interaction in the repulsive regime ([Magonov96] p. 39). Numerical simulations done by Sokolov et al support the hypothesis that repulsive force interaction is necessary to achieve the atomic resolution observed with non-contact mode [Sokolov97].

Appendix B Defining a Surface at the Atomic Scale

In this section I describe how the surfaces implied by my models of AFM and SEM image formation are related to actual surface structure at the atomic scale. In modeling AFM and SEM image formation, I assume that the specimen is described by a single surface with a homogenous material on one side and vacuum on the other side. This approximation should not be confused with the jellium approximation used in physics to model surfaces [Brack93] [Zangwill88]. In the jellium model, positive (nuclear) charge is uniformly distributed within a volume but electron density is allowed to vary continuously. In the MONSEL model (the one I use) and all other SEM simulation codes that I am aware of, both the atomic nuclei and electrons in the specimen are assumed to be uniformly distributed within the same bounding surface.

The parameters of this model are adjusted to maximize the likelihood that scans of this surface would produce the observed AFM and SEM data. However, when an actual material is scanned, there is an underlying true surface that is more accurately described by a variable electron density and a set of positions for discrete atomic nuclei. Furthermore, the computed surface corresponding to the AFM data may be different from that corresponding to the SEM data. I show that this difference is much smaller than the expected error in the reconstruction due to measurement errors. To do this, I will consider a very simple example of an atomically flat specimen surface and use the height, measured in the z dimension, of the centers of the uppermost layer of atomic nuclei as a reference point with z=0. It doesn't really matter which part of the atomic nuclei we use as a reference because a nucleus is for practical purposes a point with a diameter about 4e-6 nm (about 10^{-5} times the interatomic distance). The computed surface is assumed to be of the form z=z_0 (z_0 is a constant).

The forward model for SEM is solved using a Monte Carlo model of electron scattering. In general, the sensitivity of an incident or escaping electron to the position of the surface is a function of the mean free path of the electron in the material and the angle of incidence or escape relative to the surface normal. Here I consider only the case of incident electrons entering the specimen parallel to the surface normal and focus on the effect of assuming a constant mean free path for the material. The mean free path is inversely proportional to the material density, if all other material parameters and incident electron energy are constant. The distribution for the distance traversed by the electron between consecutive scattering events is given by an exponential distribution,

$$p(x) = \begin{Bmatrix} \frac{1}{\mu} e^{-x/\mu} & x \geq 0 \\ 0 & x < 0 \end{Bmatrix},$$

where μ is the mean free path. For example, in silicon the mean free path of an electron with energy 1keV is approximately 0.6 nm and decreases gradually to a minimum of about 0.1 nm as energy is lost. Calculations assuming the jellium model predict a thin layer near the material surface with thickness approximately equal to the bond length in the material over which the electron density transitions from its maximum in the material to a small fraction of the maximum (on the vacuum side) [Brack93][Zangwill88]. This layer is about 0.235 nm thick for silicon. Calculations for graphite show that the lower boundary of the transition layer is well within the bond length of the top layer of atomic nuclei [Ciraci90]. Thus, the position of the optimal computed surface given a hypothetical noiseless SEM image of an actual surface is somewhere between z=0 and $z=\delta$ where δ is the interatomic distance. The offset is in the z direction in this case because the surface is flat and perpendicular to the electron beam that is parallel to the z-axis. In general the offset is in a direction normal to the surface. For a surface that is not flat, an offset normal to the surface would effectively dilate or contract topographic features in the SEM image. Actually, the situation is a bit more complicated. Consider an electron traveling very close and parallel to the surface. The farther it travels, the more likely it is to interact with the surface. The effect would be to make the surface appear to dilate more the farther it is from the source of electrons. In any case, the apparent position of the surface for the SEM assuming the homogenous material model should not deviate much more than the bond length from the surface passing through the nuclei of the outermost atoms. It is difficult to be more precise without having a more detailed model of electron scattering that takes into account discrete atoms and varying electron density. Some work has been done in this area [Garcia-Lekue03] but I have not found any research on computing the final signal measured by the

SEM using such a detailed model of the surface. For surfaces that are not flat, I conclude that a surface reconstructed from SEM data would be a slightly dilated version of a surface passing through the outermost layer of atomic nuclei in the specimen but that the difference would be of the same order of magnitude as the bond length for the specimen material, which is 0.235nm for the silicon surfaces being studied.

When using the AFM to acquire images with atomic resolution, it is safe to assume that the AFM is being operated in a vacuum to avoid any surface water layer that would distort the surface and obscure the atomic structure. For the AFM, I assume that the computed surface is traced by the point on the tip that is closest to the specimen surface when the probe is in contact with the specimen and the force between the probe and specimen is constant with a value of 0. In fact, the AFM images I use are taken in a mode in which the AFM tip is oscillating at the resonance frequency of the cantilever and using feedback to control the fraction of free-space oscillation that the cantilever travels when it is "at the surface" but I assume this will mainly introduce an offset in z and not any significant geometric distortion (beyond that caused by the tip shape itself). Ciraci et al. have estimated the force on an AFM probe as a function of distance from a graphite surface taking into account the forces from discrete atoms [Ciraci90]. In their calculations, a single aluminum atom was used in place of the AFM probe and just the top layer of carbon atoms in the specimen surface was considered. The calculations showed that the force between the aluminum atom and the surface was 0 at about 0.2-0.25 nm above the nuclei of the carbon atoms depending on the lateral position of the aluminum atom relative to the graphite lattice (see Figure B-1). The distance between atoms in the graphite lattice is about 0.14 nm so as was the case for the SEM, the height of the computed surface (relative to the top layer of atomic nuclei) for AFM is about the same order of magnitude as the interatomic distance (see Figure B-2).

In this dissertation the goal is use SEM data to correct errors in the AFM-based surface reconstruction that are on the order of tens to hundreds of nanometers. I ignore the fact that there may be a difference of ~0.3 nm between the surfaces implied by the AFM and SEM data because this difference is much smaller than the maximum difference I would expect between the true surface and any surface reconstructed using the data and algorithm I use or any other previously developed algorithm. Other limiting factors such as noise, feedback artifacts, image resolution, and unknown distortions in the scanner due to drift or electromagnetic interference are expected to introduce errors much greater than 0.3 nm, at least with the instruments I am using.

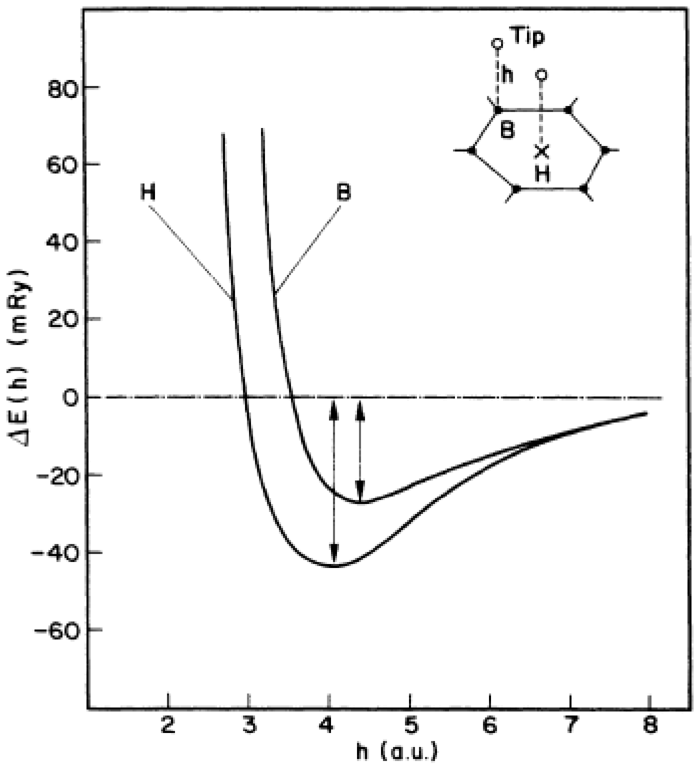


Figure B-1: Potential energy for the interaction between an aluminum tip and a graphite monolayer as a function of distance between the tip and graphite atomic nuclei. (Reprinted figure with permission from Ciraci, Baratoff and Batra, Physical Review B, vol 41, pg 2766, 1990. Copyright 1990 by the American Physical Society) [Ciraci90]

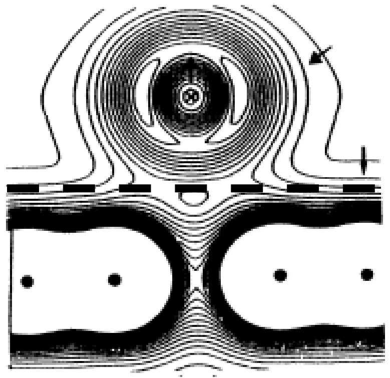


Figure B-2: Contours of constant charge density for an aluminum atom positioned about 0.3 nanometers above a graphite monolayer. The dashed line indicates the approximate effective position of the surface for the homogenous charge model assumed for the SEM. (Reprinted figure with permission from Ciraci, Baratoff and Batra, Physical Review B, vol 41, pg 2770, 1990. Copyright 1990 by the American Physical Society) [Ciraci90]

Appendix C Mathematical Morphology and Analysis of AFM Images

One of the most important tools for understanding the relationship between the shape of a surface and its AFM image is a branch of mathematics known as mathematical morphology. In this section I describe the basic theory of mathematical morphology using notation and definitions from [Haralick87] and then its application to interpreting AFM images.

C.1 Mathematical Morphology

The basic operations for mathematical morphology were first defined by Hermann Minkowski in 1903. He defined set-addition and set-subtraction operations for sets of points. These operations illustrated in Figure C-1, are known as the Minkowski sum (or dilation) and the Minkowski difference (or erosion). Given two sets of points $A \subseteq \mathbb{R}^3$ and $B \subseteq \mathbb{R}^3$, the dilation (symbolized by $\oplus$) is defined by

$$A \oplus B = \bigcup_{b \in B} A + b = \bigcup_{b \in B} \{a + b \mid a \in A\}$$

and Minkowski subtraction or erosion (symbolized by the operator $\ominus$) is defined by

$$A \ominus B = \bigcap_{b \in B} A - b = \bigcap_{b \in B} \{a - b \mid a \in A\}$$

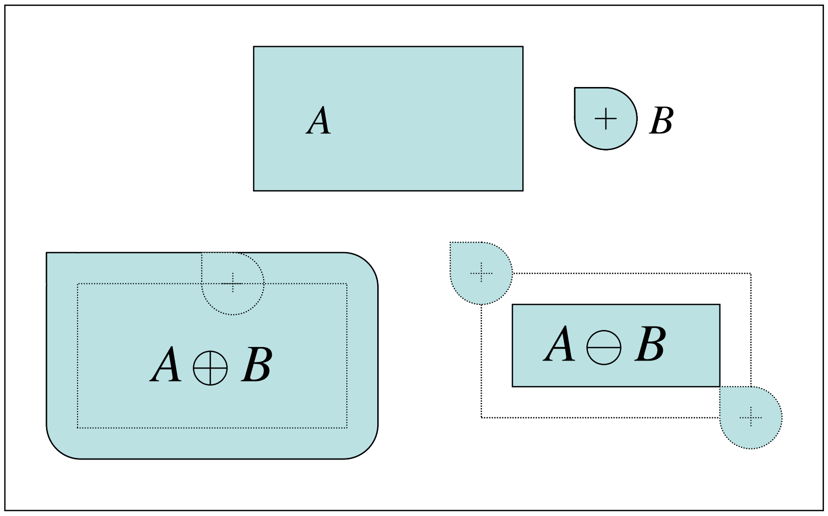


Figure C-1: Example of erosion and dilation between two sets *A* and *B*.

"Grayscale mathematical morphology" involves the Minkowski addition and subtraction operations applied to point sets defined in terms of the umbra ("shadow") of a surface. The umbra of a function *f*(*x,y*) is defined by

$$U[f] = \{(x, y, z) | z \leq f(x, y)\}$$ (see also Figure C-2)

The reverse of the operation for taking the umbra is the surface or top of a set *A* defined by

$$T[A](x, y) = \max\{z | (x, y, z) \in A\}$$

Equation C-1

We can represent the surfaces of the sample and probe by the functions *s*(*x,y*) and *p*(*x,y*). The AFM image of the sample taken with the probe is approximated by the gray-scale dilation of *s*(*x,y*) by *p*(*x,y*). The gray-scale dilation of *s*(*x,y*) by *p*(*x,y*) is the Minkowski addition of the umbra of s(*x,y*) to the umbra of *p*(*x,y*) and is defined by

$$(s \oplus p)(x, y) = T\left[U[s] \oplus U[p]\right](x, y)$$

The gray-scale erosion of a surface *f*(*x,y*) by *p*(*x,y*) involves the Minkowski subtraction of the umbra of *p*(*x,y*) from the umbra of *f*(*x,y*) and is defined by

$$(f \ominus p)(x, y) = T\left[U[f] \ominus U[p]\right](x, y)$$

When we take the top of a translation of *A* the relation to the top of the original *A* is

$$
\begin{aligned}
T[A+b](x,y) &= \max\{z \mid (x,y,z)\in A+b\} \\
&= \max\{z \mid (x-b_x, y-b_y, z-b_z)\in A\} \\
&= \max\{z \mid (x-b_x, y-b_y, z)\in A\} + b_z \\
&= T[A](x-b_x, y-b_y) + b_z
\end{aligned}
$$

Equation C-2

When we take the top of the union of two sets *A* and *B* the relation to the tops of *A* and *B* is

$$
\begin{aligned}
T[A\cup B](x,y) &= \max\{z \mid (x,y,z)\in A\cup B\} \\
&= \max\{\max\{z \mid (x,y,z)\in A\}, \max\{z \mid (x,y,z)\in B\}\} \\
&= \max\{T[A](x,y), T[B](x,y)\}
\end{aligned}
$$

Equation C-3

When we take the top of the intersection of two sets *A* and *B* the relation to the tops of *A* and *B* is

$$
\begin{aligned}
T[A\cap B](x,y) &= \max\{z \mid (x,y,z)\in A\cap B\} \\
&= \min\{\max\{z \mid (x,y,z)\in A\}, \max\{z \mid (x,y,z)\in B\}\} \\
&= \min\{T[A](x,y), T[B](x,y)\}
\end{aligned}
$$

Equation C-4

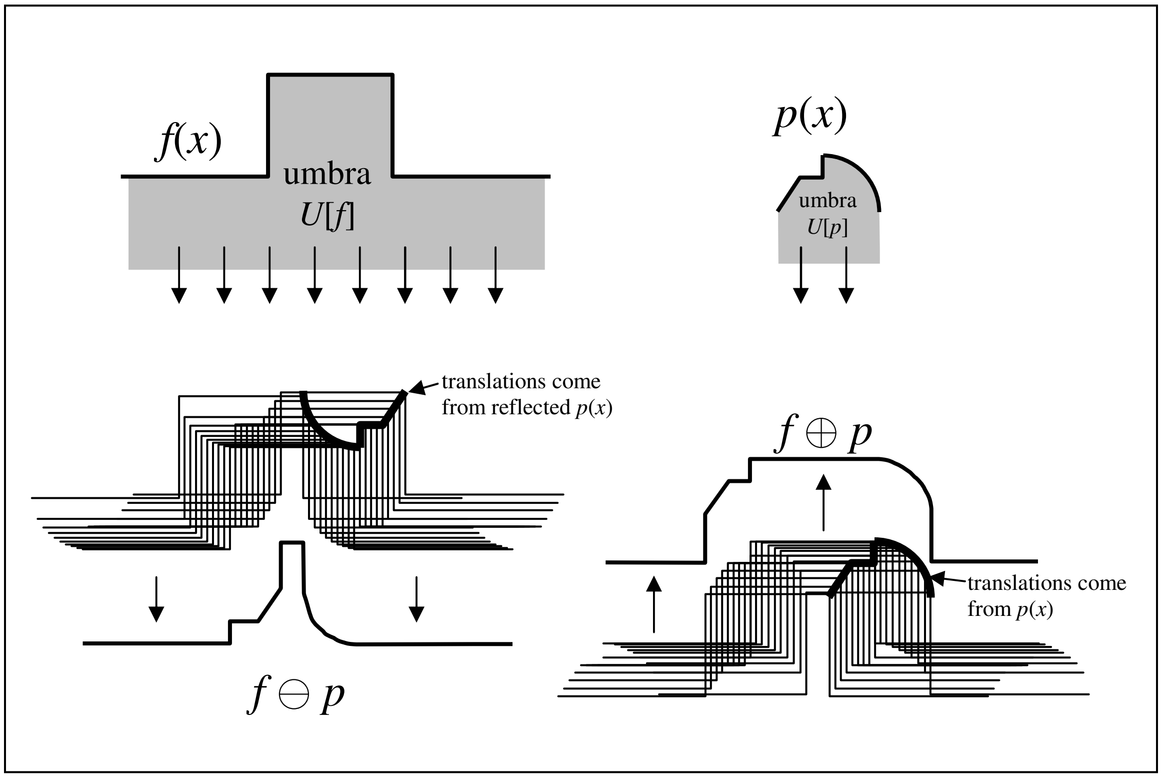
Given two surfaces *f*(*x,y*) and *p*(*x,y*), the gray-scale dilation of *f* by *p* is essentially a union of translated versions of *U*[*f*] where the translations are the points in *U*[*p*]. Using Equation C-1, Equation C-2, and Equation C-3 we can derive an expression for the resulting surface *h*(x,y) in terms of the surfaces *f*(*x,y*) and *p*(*x,y*).

$$
\begin{aligned}
h(x,y) = (f\oplus p)(x,y) &= T\left[\bigcup_{b\in U[p]} (U[f]+b)\right](x,y) \\
&= \max_{b\in U[p]} \{T[U[f]+b](x,y)\} \\
&= \max_{b\in U[p]} \{T[U[f]](x-b_x, y-b_y) + b_z\} \\
&= \max_{(b_x,b_y)} \{f(x-b_x, y-b_y) + p(b_x,b_y)\}
\end{aligned}
$$

Similarly, the erosion of *f* by *p* is an intersection of translated versions of *U*[*f*] where the translations are reflected position vectors from *U*[*p*]. Using Equation C-1, Equation C-2, and Equation C-4 we can derive the following expression for the eroded surface *h*(*x,y*) in terms of *f*(*x,y*) and *p*(*x,y*).

$$
\begin{aligned}
h(x,y) = (f \ominus p)(x,y) &= T\left[\bigcap_{d \in U[p]} \left(U[f]-b\right)\right](x,y) \\
&= \min_{b \in U[p]} \left\{ T\left[U[f]-b\right](x,y) \right\} \\
&= \min_{b \in U[p]} \left\{ T\left[U[f]\right]\left(x+b_x, y+b_y\right) - b_z \right\} \\
&= \min_{(b_x,b_y)} \left\{ f\left(x+b_x, y+b_y\right) - p\left(b_x,b_y\right) \right\}
\end{aligned}
$$

Examples of the grayscale erosion and dilation operations are illustrated in Figure C-2.



$f(x)$
umbra
$U[f]$
$p(x)$
umbra
$U[p]$
translations come from reflected $p(x)$
$f \oplus p$
translations come from $p(x)$
$f \ominus p$

Figure C-2: Example of grayscale erosion and dilation operations applied to the surfaces defined by *f*(*x*) and *p*(*x*). The erosion is the top of the set which is the intersection of translations of the set of points in the umbra of *f*(*x*) by reflected position vectors from the umbra of *p*(*x*). The dilation is the top of the set which is the union of translations of the set of points in the umbra of *f*(*x*) by position vectors from the umbra of *p*(*x*).

C.2 Application of Mathematical Morphology to AFM Image Analysis

The application of mathematical morphology to image analysis has a long history [Haralick87] but the special relevance to analysis of AFM images has become widely recognized relatively recently. Ignoring noise and imperfections in the feedback system, [Pingali92] showed that an AFM image, *h*(*x,y*), acquired in contact mode is a dilation of the specimen surface (*s*) by the reflected probe surface (*p*^).

$$h(x,y) = [s \oplus p\,\hat{}\,](x,y)$$

Also, in their model for non-contact mode AFM where there is assumed to be a fixed distance, *d*, between the tip and surface, the acquired image ($h_n(x,y)$) is equal to a two stage dilation of the specimen surface (*s*) first by a hemisphere (*g*) of radius *d*, and second by the reflected probe surface (*p*^) [Pingali92].

$$h_n(x,y) = [s \oplus g \oplus p\,\hat{}\,](x,y)$$

A reconstruction, *r*(*x,y*), of the specimen surface can be computed from a contact mode AFM image, *h*(*x,y*), by eroding it with the probe *p*(*x,y*) [Pingali92].

$$r(x,y) = [h \ominus p](x,y)$$

C.2.1 Finding Points on the Surface

Not all points in *r* will match the true specimen surface *s* if the probe does not contact all parts of the surface. Even if the probe does touch all parts of the surface, in general there is only a subset of the reconstructed surface that one can know for certain matched the true surface. Pingali and Jain developed a certainty map algorithm that computes a set of points where *r* is certain to match *s*. The points in *r* that are not flagged as certain by this algorithm could match *s* but this is something the AFM image cannot tell us. For *r*(*x,y*) to be certain, there must be some point (x_1,y_1) in the image *h* such that the point (*x,y*) on the surface reconstruction is alone in contacting the probe surface *p* translated to (x_1,y_1). Formally, the condition for a point (*x,y*) to be certain (i.e. *r*(*x,y*) = *s*(*x,y*)) is

$$\begin{aligned}&r(x,y) = s(x,y) \text{ if}\\ &\exists\ (x_1,y_1) \text{ such that } h(x_1,y_1) = r(x,y) + p(x_1 - x, y_1 - y)\\ &\text{and } \forall\ (x_2,y_2) \neq (x,y) \text{ we have } h(x_1,y_1) \neq r(x_2,y_2) + p(x_1 - x_2, y_1 - y_2)\end{aligned}$$

A similar condition was described for the certainty of a point in a non-contact AFM image [Pingali92].

C.2.2 Estimating Probe Shape

The algorithms described by Pingali and Jain require the shape of the probe p(x,y). Probe shape can either be measured using a special tip characterizer sample with a known shape or estimated using a blind tip reconstruction procedure. A characterizer usually has sharp spikes or mushroom-shaped projections on its surface. By scanning the tip characterizer and eroding the resulting image with the known shape of the characterizer, the tip is reconstructed.

One of the difficulties in using a special tip characterizer surface for tip measurement is that the shape of the tip will change from one image to the next or even while acquiring an image (particularly when imaging in contact mode) [Schneir96]. Using an arbitrary image to estimate the tip has the potential to find a more accurate model of the tip shape for the time that the image was acquired. The blind tip reconstruction method was discovered independently by Villarrubia [Villarrubia94] and Williams [Williams96]. The basic idea in blind tip reconstruction is that sharp spikes or corners on the specimen surface produce features in the AFM image that allow one to bound the shape of the AFM tip.

Blind tip reconstruction starts with an initial estimate for the tip shape, which is an upper bound, or blunt approximation of the actual shape of the tip. The estimate is iteratively refined based on features in the AFM image. The goal is to determine an upper bound on the tip surface that is consistent with the AFM image. One way to measure consistency is by eroding and then dilating the image with an estimate of the tip shape. This operation, called opening, is denoted by the operator $\circ$ and is defined as

$$A \circ B = (A \ominus B) \oplus B$$

A related operation called closing, is denoted by the operator $\bullet$ and is defined as

$$A \bullet B = (A \oplus B) \ominus B$$

Equation C-5: closing operation

An important property for the analysis of AFM images is

$$\left(A \oplus B\right) \circ B = A \ \oplus B$$

Equation C-6: idempotency of opening operation when applied to a dilation

This property tells us that if an image was formed by a dilation of some surface by a probe surface, then when we open the image with the same probe surface we should get the same image.

Another way to look at the opening operation is that it computes the space swept out by the probe as it is scanned over the underside of the AFM image. The condition in Equation

C-6 is equivalent to saying that the probe surface must touch every point in the surface without sticking above the surface (as illustrated in Figure C-3).

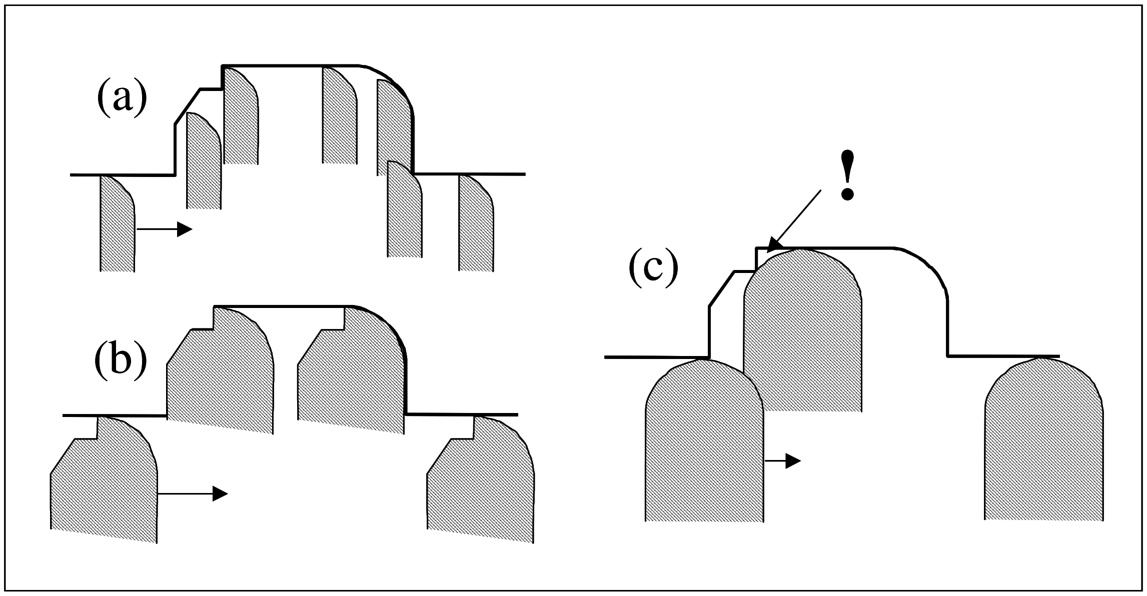


Figure C-3: A probe that is consistent with a given dilation must be able to touch every part of the surface when scanned from below. (a) shows a probe that is consistent but is not the upper bound on consistent probes. (b) shows the optimal probe that would be found by the blind tip reconstruction method. Its surface bounds that of all consistent probes. (c) shows how an inconsistent probe cannot touch all parts of the dilated surface.

For blind tip estimation, in principle one can consider every point in the AFM image (although some points may tell us much more about the tip shape than others). After choosing an initial estimate for the tip shape, P_0, one must pick a point, x, in the AFM image to use for refining the tip estimate. The next step, illustrated in Figure C-4, is to take the union of the umbrae of all translated versions of the image surface such that x lies on the tip surface and the origin of the tip lies under the image surface. The subset of points in P_0 that satisfy this criterion when used as translation vectors for the surface is referred to as P_0'. Local maxima in the image are special because in that case only the origin of the tip is in P_0'. As a result, local maxima usually provide the most information about tip shape. Because of the special property of local maxima, they can actually be used to easily pick an initial tip estimate as

$$P_0 = \bigcap_{x \in M} (I - x)$$

where M is the set of local maxima in the image and I is the umbra of the AFM image.

Villarrubia claims that the initial estimate calculated in this way is often a good estimate of the actual tip shape and is less sensitive to noise than the estimate after multiple iterations [Villarrubia97].

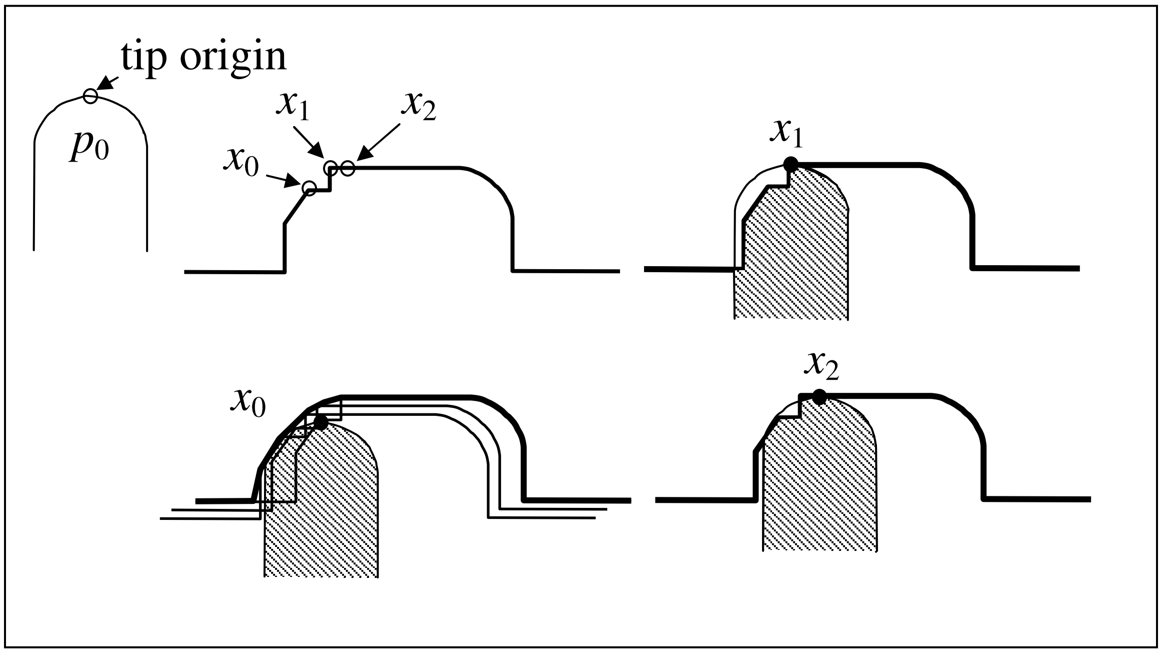


Figure C-4: Intermediate results for blind reconstruction from three points in an AFM image. p_0 is our initial upper bound on the (reflected) tip shape with apex assumed to be at the origin. The bound is refined by taking the intersection with bounding surfaces generated for each point in the image. Here the construction of the bounding surfaces is shown for three example points (x_0, x_1, x_2). For each point x_i, we take the union of all translations ($-x_i$+d) of the surface such that the point that was at x_i lies on the tip surface and the origin (tip apex) lies below the translated surface. In the case of x_0 there are multiple translations allowed but for x_1 and x_2 the only translation allowed is the one where either x_1 or x_2 is at the origin. For each point, the resulting bounding surface shown would be intersected with the current tip estimate to compute a new tip estimate.

C.2.3 Finding Points on the Surface from Real AFM Images

Using tips estimated using a tip characterizer or by blind tip reconstruction, several authors have attempted to apply Pingali and Jain's certainty map algorithm to determine what parts of an eroded AFM image match a real surface. Because a real AFM image does not represent a pure dilation but contains artifacts due to noise, surface/tip deformation, and imperfect feedback, their certainty map algorithm tends to underestimate the number of certain points and misclassify some points that are not on the surface as certain. The problem is that downward spikes in the AFM image due to these artifacts incorrectly erode tip-shaped depressions in the reconstructed surface. When the tip is scanned over the reconstructed surface, it cannot contact any points inside these depressions except when it is contacting all of the points so there is no point that is certain inside the depression. This problem is illustrated in Figure C-5.

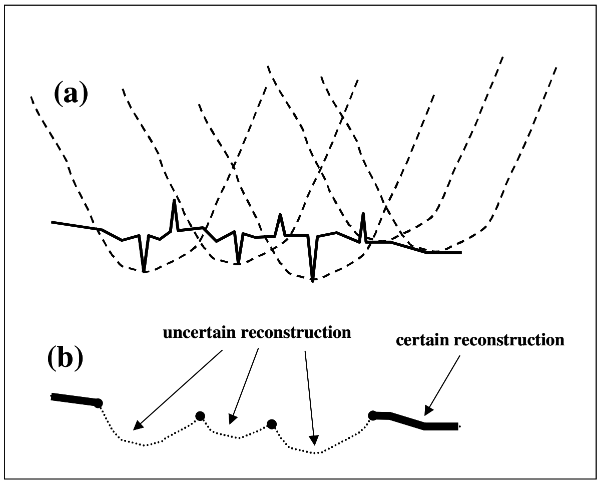


Figure C-5: Points in the certainty map are mislabeled when Pingali and Jain's algorithm is applied to a noisy dilation

C.2.4 Dealing with Noise

One approach to dealing with the noise is to try to filter it out before applying the erosion operation [Castle98]. This method attempts to first reconstruct the ideal dilated surface from a noisy dilated surface by using a closing operation (see Equation C-5) to prefilter the dilated image before eroding using the tip shape. The kernel for the closing operation is essentially a small flat-tipped probe with width slightly less than or equal to the estimated radius of the real probe. The closing filter was shown to be better at eliminating downward noise spikes than a median filter of the same size. Also, a flat function was found to be the optimum choice for a closing filter because it reduces downward spikes while minimizing distortion to other parts of the surface. However, the closing filter can remove significant detail that is not due to noise [Castle98]. In fact, very sharp indentations in a surface dilated with a spherical tip may be due solely to surface features as shown in Figure C-6.

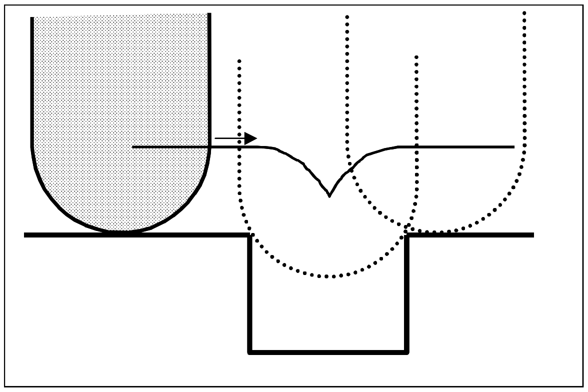


Figure C-6: Example of sharp features in a dilated surface that would be removed by a flat closing filter as proposed by Castle et al.

Noise, feedback artifacts or deformation of the tip or specimen during scanning typically cause problems for the basic blind tip reconstruction algorithm that assumes the AFM image is a pure dilation. Villarrubia estimates that the error due to deformation is typically on the order of 0.1 nm for materials used in integrated circuits. In practice there is no tip shape that will be completely consistent with Equation C-6. In [Villarrubia94] this problem is solved by ignoring differences between the opened and original images that were smaller than a threshold. In tests on synthetic AFM data with independently distributed Gaussian noise for each pixel, it was shown that to get a reasonable result, a threshold of at least 3σ was needed where σ is the standard deviation for measurements of a pixel in the image. In practice, the noise is usually correlated between nearby pixels. An ad hoc approach consisting of line-wise flattening followed by median filtering (replacing each pixel value with the median value from a small neighborhood surrounding that pixel) was recommended in order to filter out such errors inconsistent with the independent Gaussian error model. An alternative approach described in [Todd01] is to use two different threshold values to allow for greater variation between neighboring pixels in the *x* and *y* directions. For real data, the ideal choice for σ, may be estimated by trying different values and comparing the resulting tip reconstructions. Experiments with synthetic data suggested that a sharp change in the tip reconstruction will occur (tip becomes drastically sharper) when the estimate of σ becomes too small [Villarrubia97]. Experiments by Dongmo et al showed that this transition also occurs with real AFM images and that the resulting reconstruction compares reasonably well with tip profiles observed in an SEM [Dongmo00].

[Williams98] describes an alternative method to that in [Villarrubia97] for deriving a tip shape from a noisy AFM image. In [Williams98], an ellipsoid representing noise and instrumental errors is placed at each point on the noisy dilated surface. The union of the ellipsoids is considered as the volume in which the true image lies. The top and bottom surfaces for this volume are computed and during the tip reconstruction procedure, the tip is scanned over the lower surface while the upper surface is used to refine the tip model. This approach produced noticeable improvement in the tip reconstruction but only when used in conjunction with a threshold median filter. This filter works by comparing each value with the median value of its neighbors and if it exceeds that median value by a defined limit then its value is replaced by the median value. The authors suggest a reasonable threshold for the median filter is the rms value for the noise distribution.

I have not been able to locate any previous research that takes a more principled approach to the noise problem. However, some previous analysis of the statistical properties of morphological operations may be relevant. Several authors have analyzed the effect of morphological operations such as erosion, opening and closing on the expectation and variance of 1-D and 2-D signals with independently distributed noise [Morales93], [Wang95], [Mohamed95]. While this work might give some insight into what happens when we apply an erosion operation to a noisy image it doesn't address the loss of information when a morphological filter is applied to an image. Ideally, the reconstruction of a surface from a noisy image should take into account all measurements and not just those that are propagated by the erosion operator. Also, it would be preferable to incorporate a more complete noise model into a reconstruction procedure. Such a noise model would properly account for correlation between measurements that is currently handled by the ad hoc line-wise flattening method.

C.3 Continuous vs. Discrete Grayscale Morphological Operations

Existing methods for reconstructing surfaces from AFM images and simulating AFM images typically assume that discrete forms of morphological operations provide a reasonable approximation of the more realistic continuous forms of morphological operations [Villarrubia97][Villarrubia94] [Williams98][Williams96][Todd01]. In this approximation, the process of acquiring an AFM image is modeled using a discrete approximation to the dilation operation. A more correct assumption is that the AFM image is the sampled output of a continuous dilation. The discretized operations are more commonly used because it is easier to calculate the grayscale morphological operations for arbitrary discretely sampled inputs than for arbitrary continuous inputs. By choosing a sufficiently high sampling resolution, the errors in the discrete calculations can be made insignificant. In this section I explore the differences between dilation and erosion operations as they are defined on discrete grids and corresponding continuous grayscale morphological operations.

A continuous grayscale dilation of a surface $f(x,y)$ by a surface $p(x,y)$ is defined as

$$h(x,y) = \max_{(b_x,b_y)} \left\{ f\left(x-b_x, y-b_y\right) + p\left(b_x,b_y\right) \right\}$$

where x, y, b_x, and b_y are real numbers.

A discrete grayscale dilation of a discrete sampling of a surface $f_d(x,y)$ by a discrete sampling of a surface $p_d(x,y)$ is defined as

$$h_d(x,y) = \max_{i,j} \left\{ f_d\left(x-i, y-j\right) + p_d\left(i,j\right) \right\}$$

Equation C-7: discretized grey-level dilation

and discrete erosion is defined as

$$\begin{aligned} h_d(x,y) &= \min_{i,j} \left\{ f_d\left(x+i, y+j\right) - p_d\left(i,j\right) \right\} \\ &= -\max_{i,j} \left\{ -f_d\left(x+i, y+j\right) + p_d\left(i,j\right) \right\} \end{aligned}$$

where x, y, i and j are integers.

For an arbitrary surface, sampling an image at regular intervals does not in general commute with dilation (or erosion). For surfaces that are piecewise linear with vertices only at the sample points, the operations do commute. For example, if one linearly interpolates $f_d(x,y)$ and $p_d(x,y)$ to create piecewise linear surfaces $f(x,y)$ and $p(x,y)$ and takes the continuous dilation of $f(x,y)$ and $p(x,y)$ then the result $h(x,y)$ will match the discrete dilation of $f_d(x,y)$ and $p_d(x,y)$, $h_d(x,y)$, at integer coordinates. However, this is true neither for cascaded dilations nor erosions between or at integer coordinates. For example, if one first dilates and then erodes a surface using the continuous operations, the sampled resulting surface will not be exactly the same as the result using the discrete operations (Figure C-7).

A good discussion of the differences between discrete and continuous morphological operations can be found in [Hendriks03].

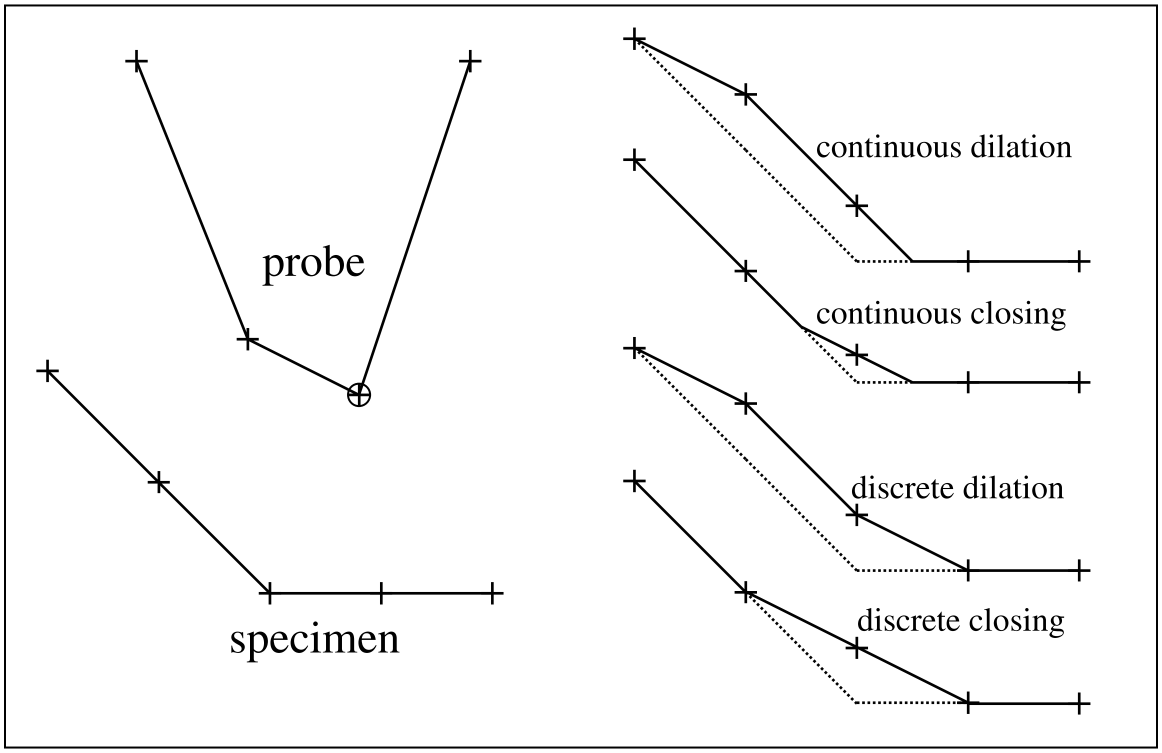


Figure C-7: Sampling cascaded continuous morphological operations on a regular grid does not produce the same result as the corresponding discrete morphological operations on inputs that have been sampled on that grid (even when applied to a piecewise linear surface with vertices at the grid points. In the discrete domain, the results are the same for a single dilation or erosion operation.

An AFM image is better described as a continuous grayscale dilation sampled on a regular grid than as a discrete dilation but by calculating the discrete dilation on a sufficiently high resolution grid, the differences can be made insignificant. I represent the specimen surface with a continuous height function that may be sampled at a resolution that is as high as is necessary to capture important details. The tip image is calculated at the same resolution as the original AFM data using Villarrubia's method (section 2.1) but is linearly interpolated to match the resolution required for the specimen surface. The discrete dilation will be a sufficient approximation as long as every patch of the continuous surface corresponding to the triangular facets of the tessellated sampled surface is sufficiently planar. Figure C-8 illustrates how an error in the approximation of the continuous dilation by the discrete dilation is related to the error in piecewise linear approximations to the tip and specimen surfaces. The magnitude of these errors for the specimens I am targeting cannot be determined without independent measurements of the tip and specimen surfaces at a significantly higher resolution than the acquired AFM images. However, if the resolution of the discrete grid used for simulation is too low, errors can result in an inability to reproduce a good approximation to the observed AFM image. In the case of the thin line example described in section 10.3.2 I found it necessary to triple the resolution of the AFM image by interpolation and use that same tripled resolution for the tip model and sampled height of the specimen model.

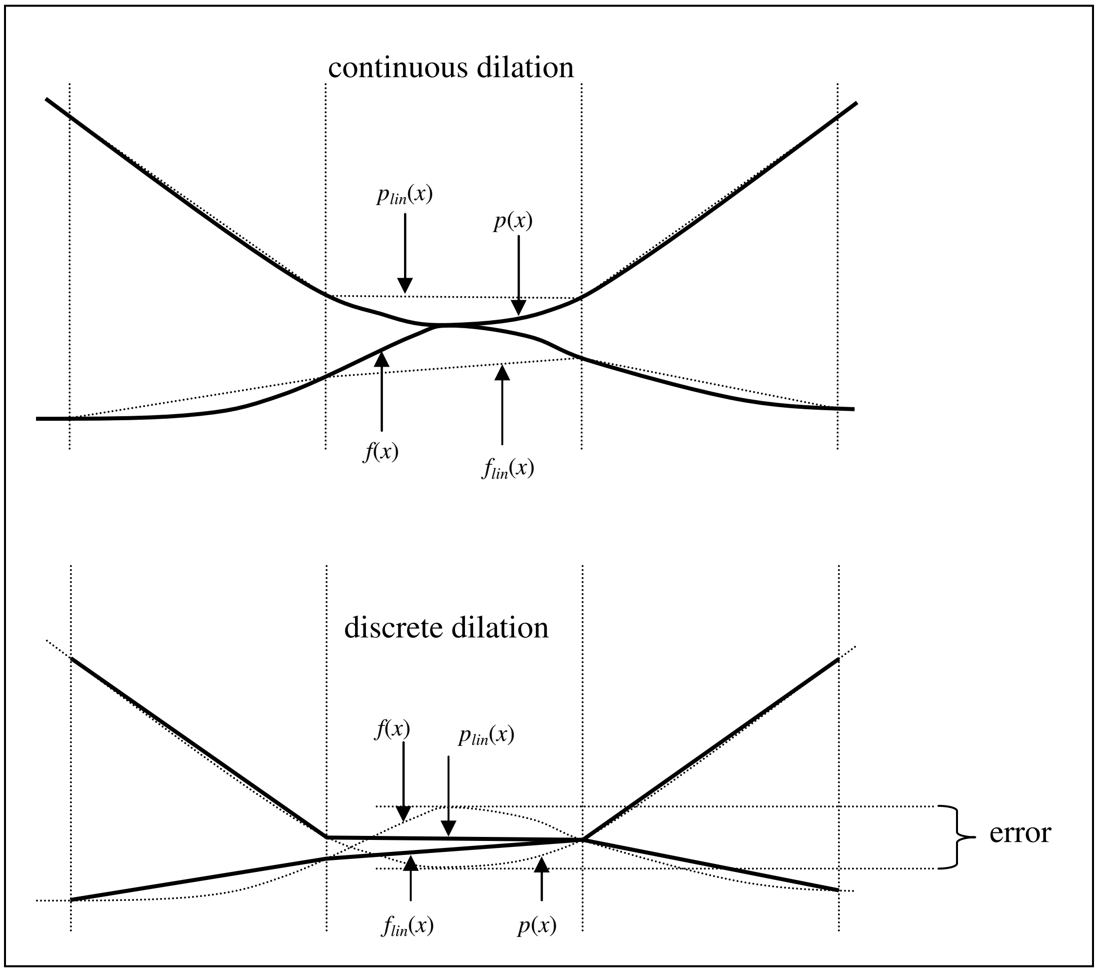


Figure C-8: Approximation of continuous dilation by applying discrete dilation to a regular sampling of the two input surfaces. The linear interpolation of the result of the discrete dilation is equivalent to the continuous dilation of the piecewise linear approximations $p_{lin}(x)$ and $f_{lin}(x)$ to the two surfaces.

Appendix D Uniformly-weighted and Gaussian-weighted Facet Models

D.1 Uniformly-weighted Facet Model

In this section I explain how the convolution kernels are computed which when convolved with an image give the facet model representation of that image [Haralick81]. In [Kleyman98] it is shown that by minimizing

$$e^2 = \sum_{r\in R}\sum_{c\in C}\left(K_1 + K_2 r + K_3 c + K_4 r^2 + K_5 rc + K_6 c^2 + K_7 r^3 + K_8 r^2 c + K_9 rc^2 + K_{10} c^3 - f(r,c)\right)^2$$

one can determine a general solution for the K coefficients. Following the notation in [Kleyman98] but with the correction of a typographical error I define

$$R_n = \sum_{r\in R} r^{2n} \text{ and } C_n = \sum_{c\in C} c^{2n} \text{ for } n = 0,1,2,3$$
$$G = R_0 R_2 C_0 C_2 - R_1^2 C_1^2$$
$$A = R_1 R_3 C_0 C_2 - R_2^2 C_1^2$$
$$B = R_0 R_2 C_1 C_3 - R_1^2 C_2^2$$
$$Q = C_0\left(R_0 R_2 - R_1^2\right)$$
$$T = R_0\left(C_0 C_2 - C_1^2\right)$$
$$U = C_0\left(R_1 R_3 - R_2^2\right)$$
$$V = C_1\left(R_0 R_2 - R_1^2\right)$$
$$W = R_1\left(C_0 C_2 - C_1^2\right)$$
$$Z = R_0\left(C_1 C_3 - C_2^2\right)$$

Equation D-1: Intermediate variables useful for computing polynomial fit

In terms of these definitions, the solution is

$$K_1 = \frac{1}{QT}\sum_r\sum_c\left(G - TR_1 r^2 - QC_1 c^2\right)f(r,c)$$

$$K_2 = \frac{1}{UW}\sum_r\sum_c\left(A - WR_2 r^2 - UC_1 c^2\right) r f(r,c)$$

$$K_3 = \frac{1}{VZ}\sum_r \sum_c \left(B - ZR_1 r^2 - VC_2 c^2\right) cf\left(r,c\right)$$

$$K_4 = \frac{1}{Q}\sum_r \sum_c \left(R_0 r^2 - R_1\right) f\left(r,c\right)$$

$$K_5 = \frac{\sum_r \sum_c rcf\left(r,c\right)}{\sum_r \sum_c r^2 c^2}$$

$$K_6 = \frac{1}{T}\sum_r \sum_c \left(C_0 c^2 - C_1\right) f\left(r,c\right)$$

$$K_7 = \frac{1}{U}\sum_r \sum_c \left(R_1 r^2 - R_2\right) rf\left(r,c\right)$$

$$K_8 = \frac{1}{V}\sum_r \sum_c \left(R_0 r^2 - R_1\right) cf\left(r,c\right)$$

$$K_9 = \frac{1}{W}\sum_r \sum_c \left(C_0 c^2 - C_1\right) rf\left(r,c\right)$$

$$K_{10} = \frac{1}{Z}\sum_r \sum_c \left(C_1 c^2 - C_2\right) cf\left(r,c\right)$$

Each of the *K* coefficients corresponds to a 2D image where each pixel represents the fit to a neighborhood centered on the corresponding pixel in an input image. The image for a *K* coefficient can be efficiently computed by a convolution with a convolution kernel the size of the neighborhood.

D.2 Gaussian-weighted Facet Model

The facet model in [Haralick81] is limited to neighborhood sizes that are odd integers and fits a cubic polynomial to the height field using a uniform weighting of all pixels in the neighborhood. I found that the uniform weighting causes the fit to vary discontinuously as the neighborhood is shifted which creates some undesirable artifacts in simulated images. To overcome this problem and the limitation on neighborhood sizes I replaced the uniform weighting function with a Gaussian weighting function. The support neighborhood size is still an odd integer but an additional width parameter for the Gaussian function provides continuous control over the effective neighborhood size. The Gaussian weighting function has the advantage of preserving separability and is defined as

$$w(r,c) = w_r\left(|r|\right) \cdot w_c\left(|c|\right) = k \cdot e^{-\left(r^2+c^2\right)/\left(2\sigma^2\right)}$$

where $w_r(u) = w_c(u) = \sqrt{k} \cdot \exp\left(-u^2/\left(2\sigma^2\right)\right)$ and *k* is a normalizing factor such that $\sum_r \sum_c w(r,c) = 1$. The polynomials K_1, K_2, K_3... K_{10} are found in a similar way to those for Haralick's original facet model except that they minimize

$$e^2 = \sum_{r\in R}\sum_{c\in C} w(r,c) \cdot \begin{pmatrix} K_1 + K_2 r + K_3 c + K_4 r^2 + K_5 rc + K_6 c^2 + \\ K_7 r^3 + K_8 r^2 c + K_9 rc^2 + K_{10} c^3 - f(r,c) \end{pmatrix}^2$$

For computing the *K* coefficients using the Gaussian-weighted facet model, the variables *G, A, B, Q, T, U, V, W,* and *Z* from Equation D-1 are computed by the same formulae except using variables R_n and C_n defined as

$$R_n = \sum_{r\in R} w_r(r) \cdot r^{2n} \text{ and } C_n = \sum_{c\in C} w_c(c) \cdot c^{2n} \text{ for } n = 0,1,2,3.$$

Then the coefficients are computed as

$$K_1 = \frac{1}{QT}\sum_r \sum_c w(r,c)\left(G - TR_1 r^2 - QC_1 c^2\right) f(r,c)$$

$$K_2 = \frac{1}{UW}\sum_r \sum_c w(r,c)\left(A - WR_2 r^2 - UC_1 c^2\right) rf(r,c)$$

$$K_3 = \frac{1}{VZ}\sum_r \sum_c w(r,c)\left(B - ZR_1 r^2 - VC_2 c^2\right) cf(r,c)$$

$$K_4 = \frac{1}{Q}\sum_r \sum_c w(r,c)\left(R_0 r^2 - R_1\right) f(r,c)$$

$$K_5 = \frac{\sum_r \sum_c w(r,c) rcf(r,c)}{\sum_r \sum_c w(r,c) r^2 c^2}$$

$$K_6 = \frac{1}{T}\sum_r \sum_c w(r,c)\left(C_0 c^2 - C_1\right) f(r,c)$$

$$K_7 = \frac{1}{U}\sum_r \sum_c w(r,c)\left(R_1 r^2 - R_2\right) rf(r,c)$$

$$K_8 = \frac{1}{V}\sum_r \sum_c w(r,c)\left(R_0 r^2 - R_1\right) cf(r,c)$$

$$K_9 = \frac{1}{W}\sum_r \sum_c w(r,c)\left(C_0 c^2 - C_1\right) rf(r,c)$$

$$K_{10} = \frac{1}{Z} \sum_r \sum_c w(r,c)\left(C_1 c^2 - C_2\right) cf(r,c)$$

For efficiency in convolving an image with the coefficient filters, I take advantage of the separability of the filters into a 1D convolution in the x-direction followed by a 1D convolution in the y-direction. All of the filters except for those for K_2 and K_3 are directly separable. The filters for K_2 and K_3 can each be decomposed into the sum of two separable filters as follows

$$K_2 = A_2 + B_2$$

$$A_2 = \frac{1}{UW} \sum_r \sum_c w(r,c)\left(A - WR_2 r^2\right) rf(r,c)$$

$$B_2 = \frac{1}{UW} \sum_r \sum_c w(r,c)\left(-UC_1 c^2\right) rf(r,c)$$

$$K_3 = A_3 + B_3$$

$$A_3 = \frac{1}{VZ} \sum_r \sum_c w(r,c)\left(B - VC_2 c^2\right) cf(r,c)$$

$$B_3 = \frac{1}{VZ} \sum_r \sum_c w(r,c)\left(-ZR_1 r^2\right) cf(r,c)$$

D.3 Estimating Gradient Magnitude and Principal Curvatures

The gradient magnitude of a surface *f(x,y)* is $G = \sqrt{f_x^2 + f_y^2}$. Using the facet model, the gradient magnitude is estimated as $G = \sqrt{K_3^2 + K_2^2}$.

The curvature of a surface *f(x,y)* is characterized by the Hessian matrix, defined in terms of the second derivatives as

$$\mathbf{H} = \begin{pmatrix} f_{xx} & f_{xy} \\ f_{yx} & f_{yy} \end{pmatrix}$$

The normal curvature is the second derivative of the height of the surface measured in a particular direction. For a direction **v**=(v_x,v_y), the normal curvature is defined as

$$\mathbf{v}^T\mathbf{H}\mathbf{v} = v_x^2 f_{xx} + v_x v_y f_{xy} + v_y v_x f_{yx} + v_y^2 f_{yy}$$

The directions in which the normal curvature is minimum or maximum are called the principal directions and the corresponding normal curvatures are the principal curvatures.

The principal directions and curvatures are the eigenvectors and eigenvalues of the Hessian. By definition, if **v** is an eigenvector of a matrix **H** then $\mathbf{Hv} = \lambda \mathbf{v}$ where $\lambda \in \mathbb{R}$ is called the eigenvalue. Subtracting $\lambda \mathbf{v}$ from both sides of the equation we get $(\mathbf{H} - \lambda I)\mathbf{v} = 0$ where $I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$. This equation only has a solution $\mathbf{v} \neq 0$ if $\det(\mathbf{H} - \lambda I) = 0$. By solving the quadratic equation $\det \begin{pmatrix} f_{xx} - \lambda & f_{xy} \\ f_{xy} & f_{yy} - \lambda \end{pmatrix} = (f_{xx} - \lambda)(f_{yy} - \lambda) - f_{xy}^2 = 0$ for λ we get two solutions which are the maximum and minimum principal curvatures κ_+ and κ_-:

$$\kappa_+ = \frac{1}{2}\left(f_{xx} + f_{yy} + \sqrt{(f_{xx} - f_{yy})^2 + 4 f_{xy}^2} \right)$$
$$\kappa_- = \frac{1}{2}\left(f_{xx} + f_{yy} - \sqrt{(f_{xx} - f_{yy})^2 + 4 f_{xy}^2} \right)$$

Using the facet model, the principal curvatures are estimated as

$$\kappa_+ = K_6 + K_4 + \sqrt{(K_6 - K_4)^2 + K_5^2}$$
$$\kappa_- = K_6 + K_4 - \sqrt{(K_6 - K_4)^2 + K_5^2}$$

D.4 Examples

Examples of the convolution kernels used to compute cubic coefficients for a neighborhood size of 31x31 are shown in Figure D-1. The gradient magnitude and two principal curvatures were computed for various neighborhood sizes for the input image in Figure D-2 using both the uniformly-weighted and Gaussian-weighted cubic fit. Comparison of the results in Figure D-3, Figure D-4, and Figure D-5 shows a significant reduction in the Gaussian-weighted filter output of the discontinuity effect observed as a ringing phenomenon in the gradient images computed using the uniformly-weighted cubic fit in Figure D-3.

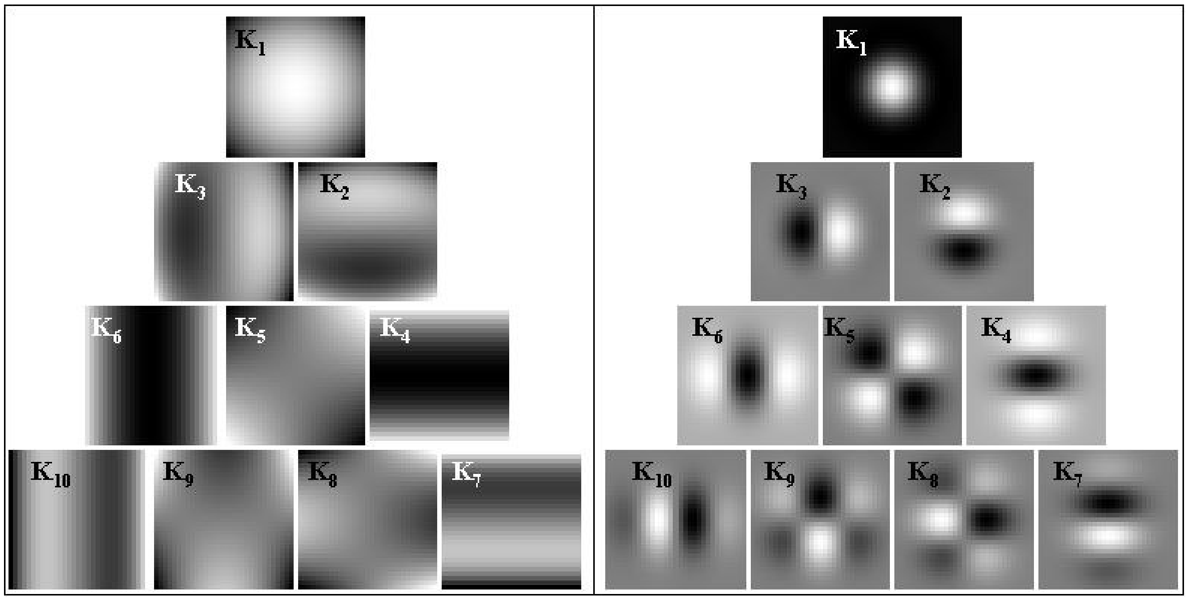


Figure D-1: 31x31 *K*-coefficient masks for uniformly-weighted least squares fit (left) and for Gaussian-weighted least squares fit (right)

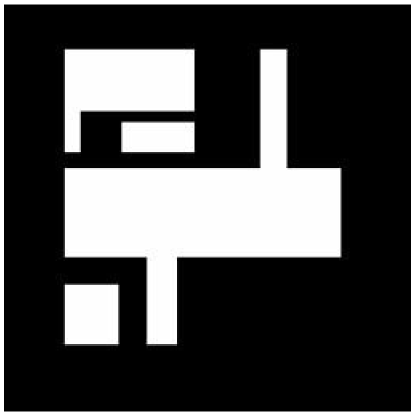


Figure D-2: Input image for comparison of outputs from filters based on uniformly-weighted and Gaussian-weighted cubic fit

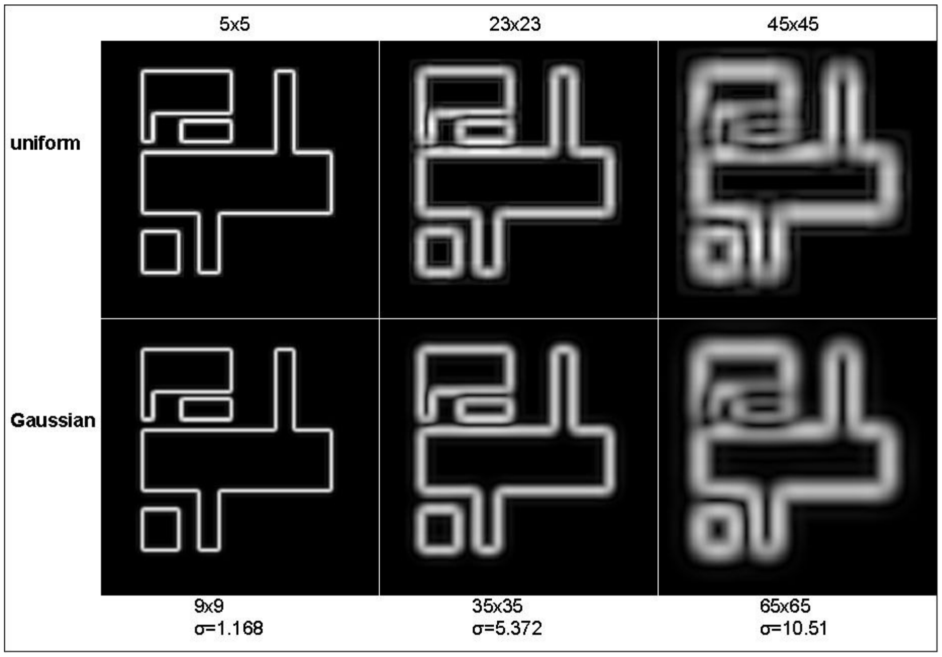


Figure D-3: Gradient magnitude. The ringing phenomenon especially visible in the upper rightmost image is reduced in the corresponding lower image. Filter neighborhood sizes and Gaussian standard deviation (σ) are specified adjacent to the corresponding filter output.

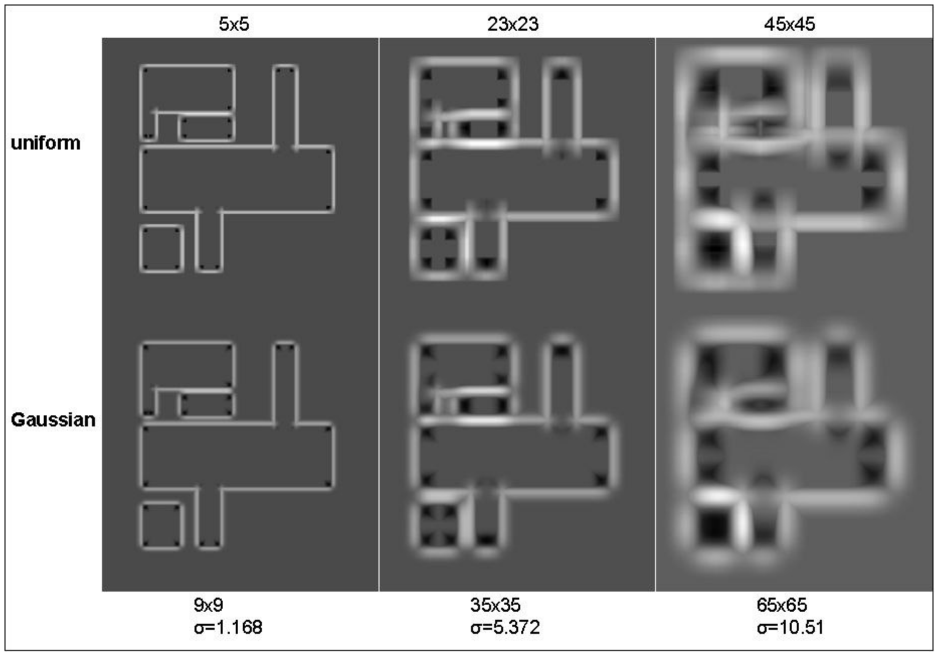


Figure D-4: Larger principal curvature. Filter neighborhood sizes and Gaussian standard deviation (σ) are specified adjacent to the corresponding filter output.

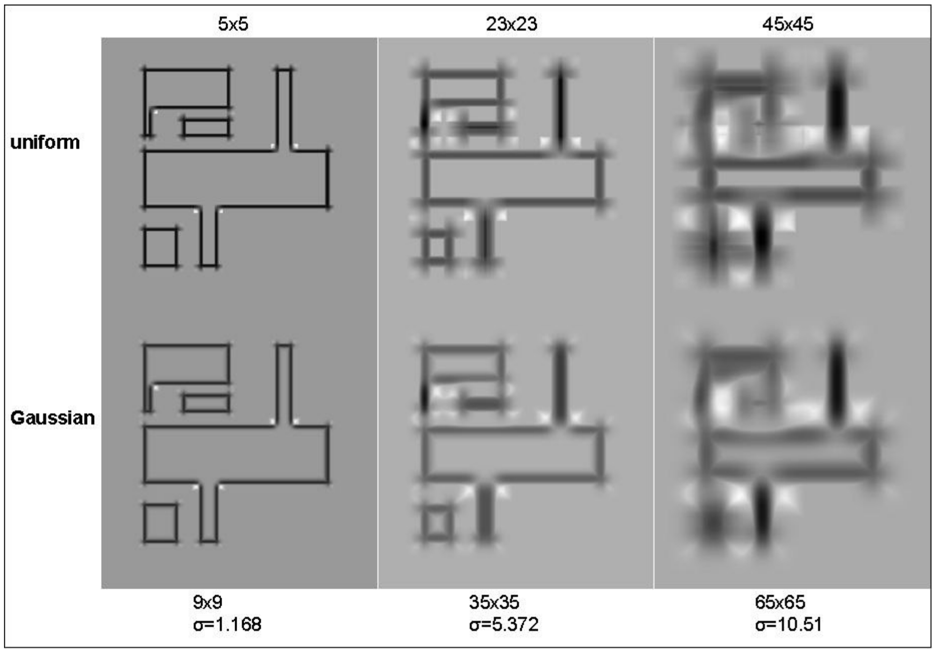


Figure D-5: Smaller principal curvature. Filter neighborhood sizes and Gaussian standard deviation (σ) are specified adjacent to the corresponding filter output.

Appendix E   Related Projects

E.1 Comparison of Monte Carlo and Slope-based SEM Simulation

Given the deviation of the Monte Carlo simulation from previously used reflectance functions, I decided to create a new reflectance function fit to the Monte Carlo simulation. The function of slope was constructed as a piecewise linear function directly from Monte Carlo simulation output from simulations on calibration surfaces with different slopes. A simple model consisting of only a single tilted plane allows no escaping electron to reenter the specimen even for large tilt angles. This situation is unrealistic because on a real specimen, curvature of the surface enables some of the escaping electrons to reenter the specimen where they generate additional backscattered and secondary electrons. A more realistic model is a surface consisting of a tilted plane intersecting a horizontal plane. The combination of a tilted plane with a horizontal plane approximates the situation where some of the escaping electrons are recaptured by the surface and can generate additional escaping electrons (see Figure E-1). A more approximate approach to take this effect into account is described in [Firsova91].

For each of 18 calibration surfaces with tilt angles spaced at 5 degree intervals between 0 and 85 degrees, 31,000 electrons were simulated to compute the yield for secondary and backscattered electrons. The output from these simulations is shown in Figure E-2.

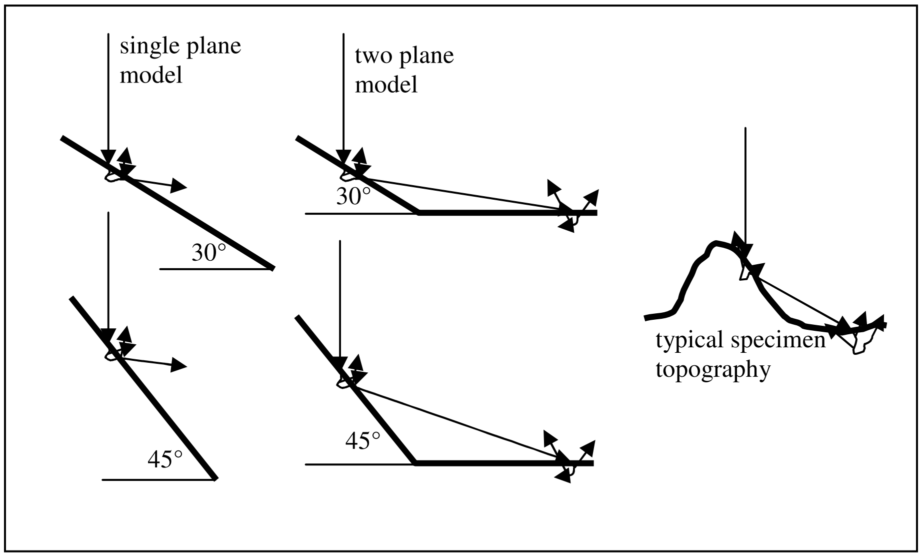
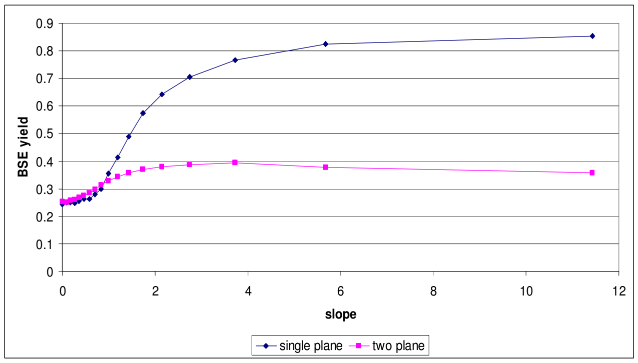


Figure E-1: Two different ways of estimating the output of the Monte Carlo simulation on a surface with a given slope. In the single plane model, escaping electrons travel directly to the detector. In the more realistic two plane model, some escaping electrons are recaptured by the surface and can generate additional secondary electrons. The two plane model better approximates what happens for a real specimen.



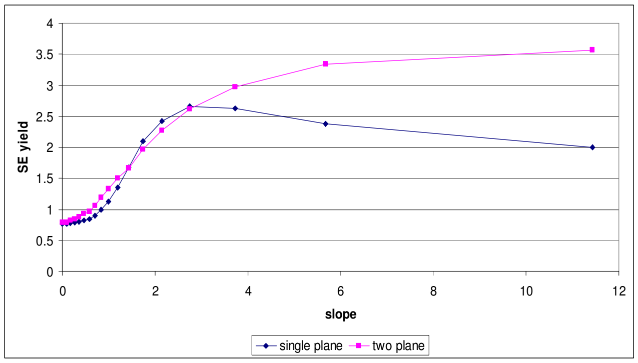


Figure E-2: Output of Monte Carlo simulation used to construct piecewise linear function of slope

The piecewise linear function of slope computed using the two plane calibration method was used to generate the images shown in Figure E-3 and Figure E-4. These images clearly show a discrepancy between the Monte Carlo simulation and the slope function that was fit to the same Monte Carlo model for a square object that is 45nm wide. The difference becomes greater as the width of the object shrinks to 15nm. A function only of slope is insufficient to model the topographic contrast at these scales.

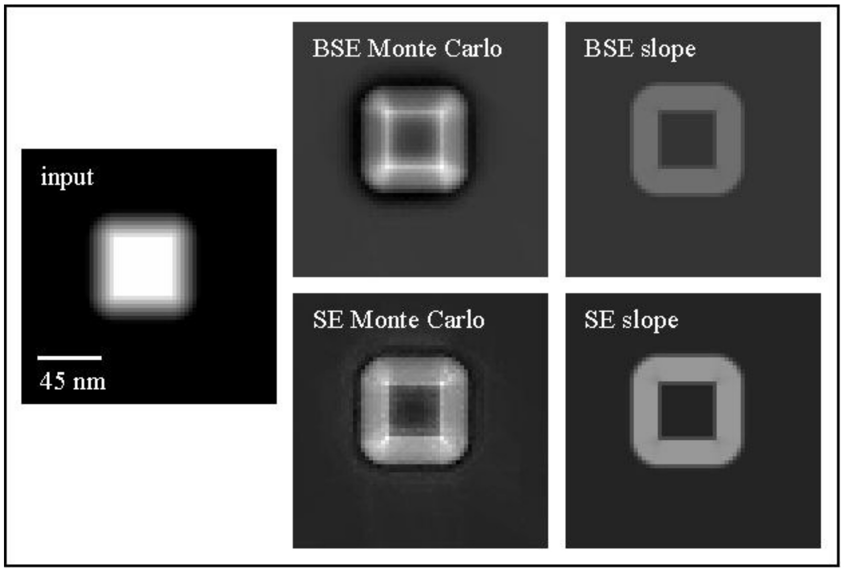


Figure E-3: Monte Carlo simulation compared with a function of slope on a 20nm high raised square 45 nm wide at the top

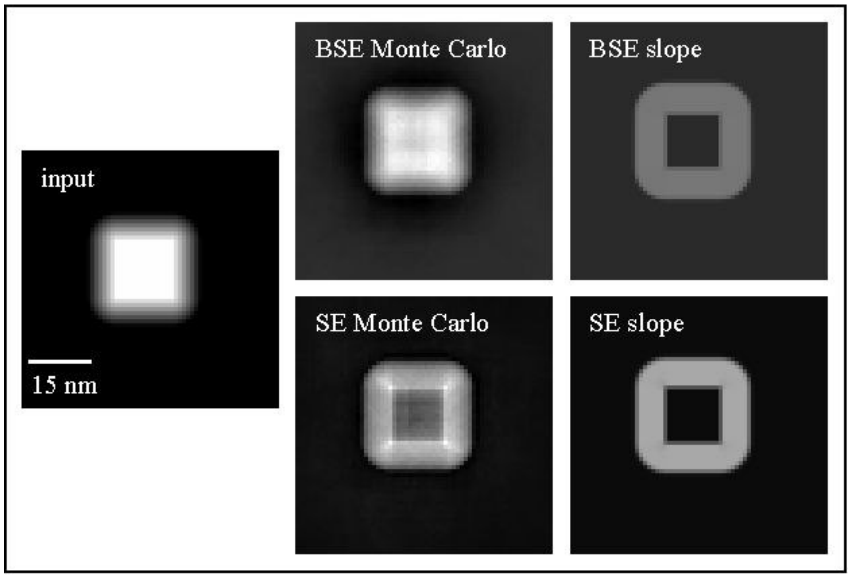


Figure E-4: Monte Carlo simulation compared with a function of slope on a 20nm high raised square 15nm wide at the top

E.2 Hardware Acceleration of Discrete Grayscale Morphological Operations

Varadhan et al [Varadhan02] demonstrated how the z-buffer hardware commonly used in computer graphics can be used to speed up the grayscale dilation operation. In this section I describe a variation on this implementation to compute discrete approximations to morphological operations.

Discrete dilation, erosion and other operations that involve taking a maximum over a set of surfaces can be accelerated using z-buffer hardware. I accessed this hardware through the OpenGL API. While this API provides both maximum and minimum operations, testing revealed that the implementation for maximum was faster than that for minimum so both dilation and erosion operations were implemented using the maximum operation. The sampled height values for the probe surface were rendered as points on a grid corresponding to the centers of display pixels. This probe display was repeated for each height sample from the specimen surface. Finally, the result was read back from the depth buffer. The z-buffer implementation of dilation is illustrated in Figure E-5 and the implementation of erosion is illustrated in Figure E-6.

In practice, I have found that the implementation using graphics hardware is not always faster and in some cases is actually slower than a software implementation depending on the particular CPU and graphics hardware. On a desktop PC with a Pentium III running at 500 MHz and a 3D Labs Wildcat graphics card of about the same vintage as the CPU, the graphics hardware implementation was about 5 times faster than the software implementation. On a laptop with a 1.3 GHz Pentium M with an ATI Mobility Radeon 9000 GPU, the graphics hardware implementation was about 3 times slower. For example, with a 300x300 pixel AFM image and a 41x41 tip image, the graphics implementation of the dilation operation took about 3 seconds while the software implementation took about 1 second. The performance for erosion was the same.

Although the graphics hardware provided acceleration on some machines, for this project, I only used CPU-based implementations because they happened to be faster than the graphics hardware implementation on my computer and it was also somewhat inconvenient to create graphics hardware implementations to perform the gradient calculations described in Section 9.3.1.

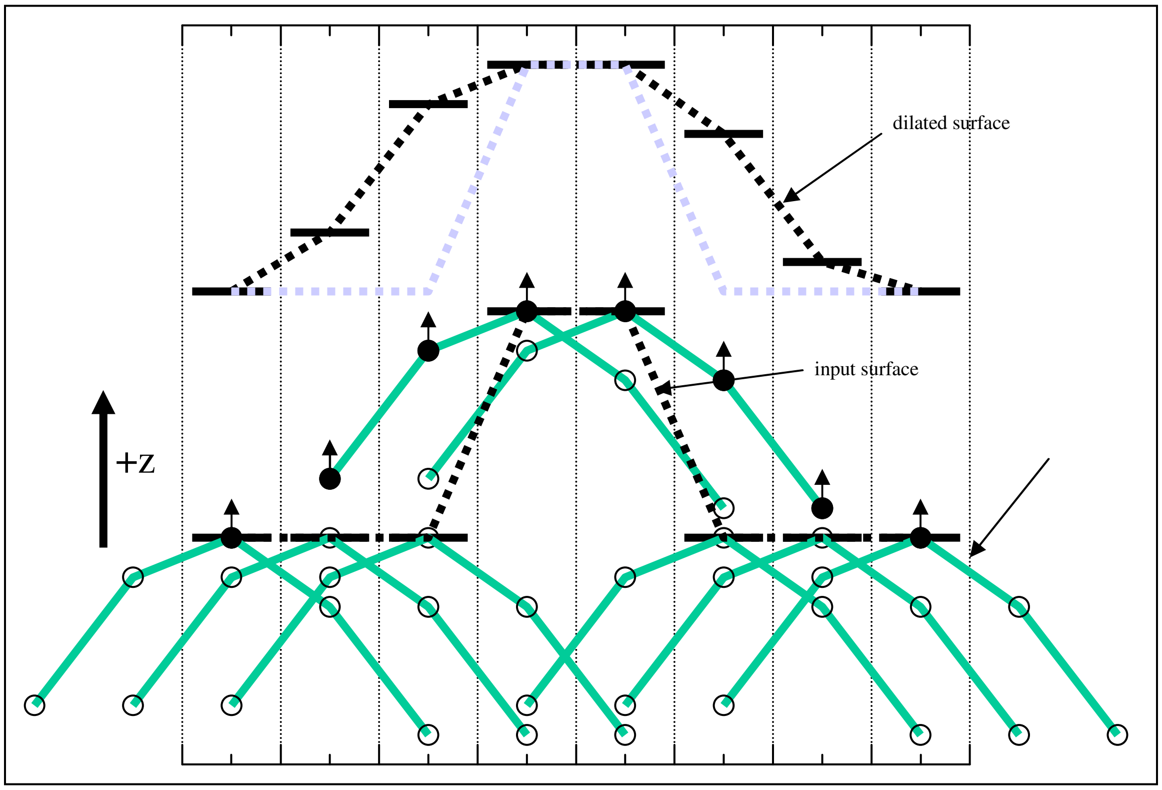


Figure E-5: Dilation as implemented using the z-buffer. Vertical lines represent pixel boundaries.

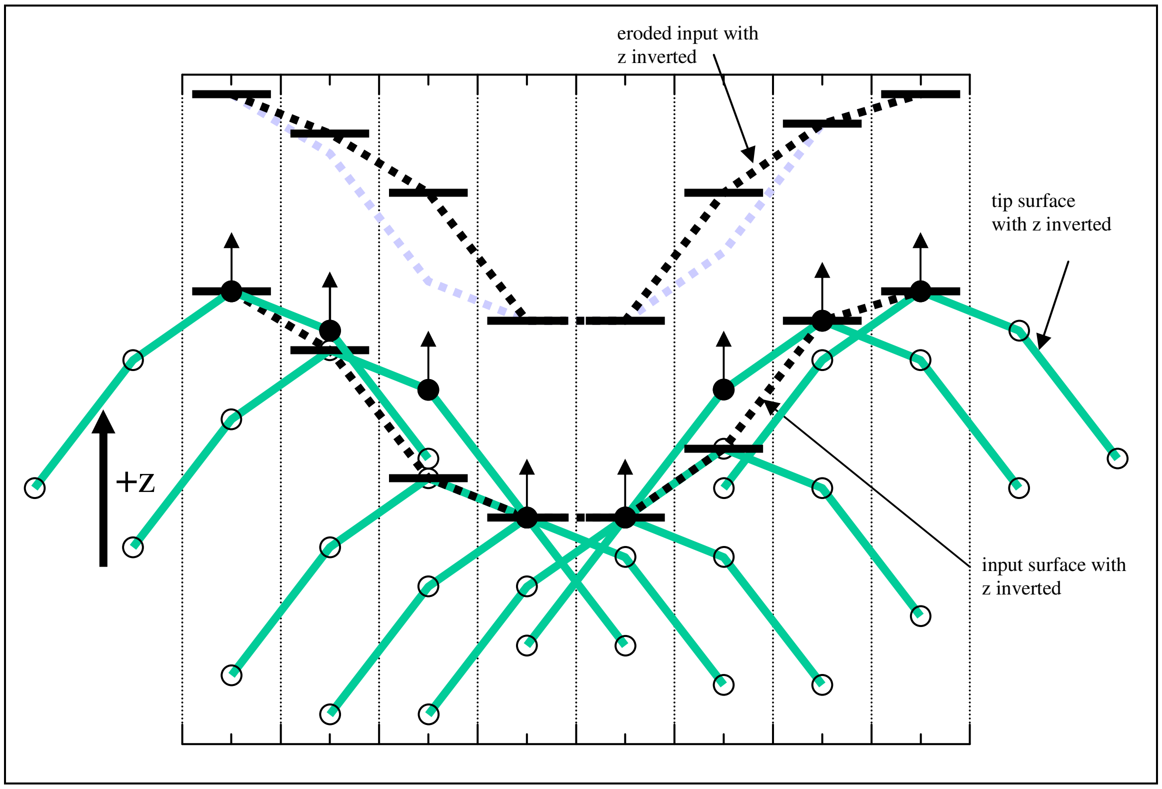


Figure E-6: Erosion as implemented using the z-buffer.

Appendix F Entropy and Mutual Information for Image Registration

In the following description, the word sample refers to a set of measurements of a random variable. Given a sample $A=\{x_a\}$ with size N_A for a random variable X there are two common methods of estimating the entropy $H(x)$. One method is to count the values in a histogram that discretizes the space of values taken on by X. The fraction of measurements that fall into a bin in the histogram is an estimate of the probability density for values of X represented by the bin. The integral in the expression for entropy is replaced by a sum over all the bins of the fraction of measurements in that bin times the log of that fraction. The estimate for the entropy of X would be defined as

$$\hat{H}(X)=\sum_{k=1}^{m}\frac{n_k}{N_A}\cdot\log\left(\frac{n_k}{N_A}\right)$$

where m is the number of histogram bins, k is an index over the bins and n_k is the number of measurements x_a falling into bin k. A similar procedure using a two-dimensional histogram is used to estimate the joint entropy. This method is used in [Collignon95].

The histogram approach implicitly models the probability distribution by placing a spike at each point represented by a histogram bin before computing the entropy of that distribution. An alternative approach is to model the distribution with a continuous function before computing entropy [Viola95]. A standard technique for creating such a continuous function from the set of measurements is Parzen Windowing. The Parzen Window density estimate is constructed by summing a set of translated versions of a function called the window function where the translations come from the sample values. Given a window function $f_w(x)$, the Parzen Window density estimate from the set of measurements A is defined as

$$\hat{p}_A(x)=\frac{1}{N_A}\sum_{x_a\in A}f_w(x-x_a)$$

A normalized Gaussian (it must be normalized so that the density integrates to 1) is commonly used as a window function because it simplifies mathematical analysis of the distribution [Viola95]. For example, given a sample of size 5 for X as A={0,0,1,3,8}, N_A=5, and a window function $f_w(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}$ then the Parzen Window density estimate would look like Figure F-1.

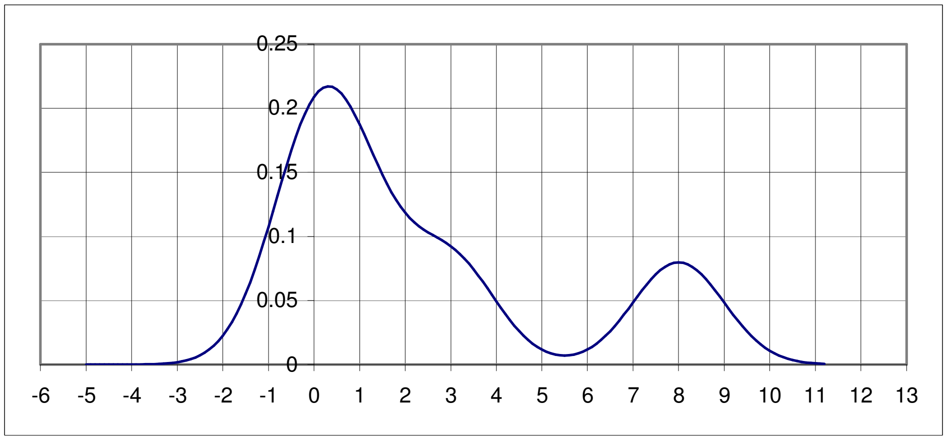


Figure F-1: Example of Parzen Window density estimate

The average value of some function over a set of measurements of a random variable gives an unbiased estimate of the mean value of that function of the random variable. Based on this idea, the entropy of the estimated probability density $\hat{p}_A(x)$ can be estimated by using the entropy function and considering a second sample of N_B values B={x_b}. The entropy is estimated by the mean value of $-\log(\hat{p}_A(x_b))$ for the sample B:

$$\hat{H}(X) = -\frac{1}{N_B} \sum_{x_b \in B} \log(\hat{p}_A(x_b))$$

For registration using mutual information, the pixel values in an image are treated in the same way as the independent measurements of a random variable in the above description for calculating entropies and mutual information. In place of a sample for X, one constructs a sample of intensities from one image and in place of a sample for Y one constructs a sample of intensities from the other image. A sample from the joint distribution for (X, Y) is constructed by repeatedly picking a pixel in one image and computing the interpolated intensity at a corresponding location in the other image (as defined by a candidate transformation that registers the two images). Ideally one would use all pixels lying in the intersection of the two images for computing the mutual information but to accelerate the calculation a random subset may be used [Viola95]. Once the pixels are selected or interpolated, as described above, one can estimate the entropy by either binning the intensities in a histogram and estimating the probabilities for each bin [Collignon95] or by estimating an integral by sampling from a Parzen Window density estimate [Viola95].

In the histogram-based method for estimating entropy it is important to have the variation in entropy with changes in the transformation be as smooth as possible to avoid getting stuck in local maxima when optimizing the mutual information. To help smooth out the mutual information cost function, particularly near the global maximum, Maes et al. used a method called partial volume interpolation [Maes97]. The transformed point in the floating image typically does not lie on a pixel but one can interpolate its value from the 4 nearest neighbor pixels (in 2D) and update the bin corresponding to the interpolated value. Instead, in partial volume interpolation, one updates the histogram by adding a fractional value to the bin for each neighboring pixel value based on the distance between the transformed point and the neighboring pixel. This modification allows the histogram values to vary smoothly with changes in the transformation and thereby smoothes out the calculated mutual information.

Jenkinson and Smith point out that the width of histogram bin that gives the most efficient, unbiased estimation of the probability distribution function is

$$W = 3.49\sigma N^{-1/3}$$

where σ is the standard deviation of the distribution and *N* is the sample size [Jenkinson00]. However, in practice, an estimate of the standard deviation needs to be used instead of the actual standard deviation, making the method less robust. They note that in practice choosing the bin size as

$$W = 2(IQR)N^{-1/3}$$

where (*IQR*) is the interquartile range (75th percentile minus the 25th percentile) tends to produce better results. For the joint histogram, the number of bins is the product of the number for one image and the number for the other image while the sample size is only the sum. Therefore, the number of values per bin for the joint histogram is greatly reduced compared with the number for the single image histogram. One might try adjusting the bin size to keep the same number of values per bin as in the 1D histogram but this is not recommended because it results in a bin size that is too large [Jenkinson00].

BIBLIOGRAPHY

[Anderson94] Anderson, E., Z. Bai, C. Bischof, et al. (1994). LAPACK Users' Guide - Release 2.0. Philadelphia, SIAM.

[Aristov91] Aristov, V. V., N. N. Dreomova, A. A. Firsova, et al. (1991). "Signal Formation of Backscattered Electrons by Microinhomogeneities and Surface Relief in a SEM." Scanning **13**: 15-22.

[Binnig86] Binnig, G., C. Quate and C. Gerber (1986). "Atomic Force Microscope." Physical review Letters **56**: 930-933.

[Binnig82] Binnig, G., H. Rohrer, C. Gerber, et al. (1982). "Surface Studies by Scanning Tunneling Microscopy." Physical Review Letters **49**: 57.

[Borgefors88] Borgefors, G. (1988). "Hierarchical Chamfer Matching: A Parametric Edge Matching Algorithm." PAMI **10**(6): 849-865.

[Brack93] Brack, M. (1993). "The physics of simple metal clusters: self-consistent jellium model and semiclassical approaches." Reviews of Modern Physics **65**(3): 677-732.

[Bresenham65] Bresenham, J. E. (1965). "Algorithm for Computer Control of a Digital Plotter." IBM Systems Journal 4(1): 25-30.

[Browning94] Browning, R., T. Z. Li, B. Chui, et al. (1994). "Empirical forms for the electron/atom elastic scattering cross sections from 0.1 to 30 keV." J. Appl. Phys. **76**(4): 2016-2022.

[Canny86] Canny, J. (1986). "A computational approach to edge detection." IEEE Transactions on Pattern Analysis and Machine Intelligence **8**(6): 679-698.

[Castle98] Castle, J. E., P. A. Zhdan and P. Singjai (1998). "Enhanced morphological reconstruction of SPM images." Journal of Physics D: Applied Physics **31**: 3427-3445.

[Chien02] Chien, F. S.-S., W.-F. Hsieh, S. Gwo, et al. (2002). "Silicon nanostructures fabricated by scanning probe oxidation and tetra-methyl ammonium hydroxide etching." Journal of Applied Physics **91**(12): 10044-10050.

[Ciraci90] Ciraci, S., A. Baratoff and I. P. Batra (1990). "Tip-sample interaction effects in scanning-tunneling and atomic-force microscopy." Physical Review B **41**(5): 2763-2775.

[Cohen00] Cohen, I., R. Golan and S. Rotman (2000). "Applying Branching Processes Theory for Building a Statistical Model for Scanning Electron Microscope Signals." SPIE Optical Engineering **39**(1): 254-259.

[Collignon95] Collignon, A., F. Maes, D. Delaere, et al. (1995). Automated multimodality image registration using information theory. Information Processing in Medical Imaging, Ile de Berder, France.

[Czyzewski91] Czyzewski, Z. and D. C. Joy (1991). "Calculation of Secondary-Electron Production Using a Diffusion Matrix." Scanning **13**(3): 227-232.

[Davidson99] Davidson, M. P. and A. E. Vladar (1999). "An Inverse Scattering Approach to SEM Line Width Measurements." SPIE Conference on Metrology, Inspection, and Process Control for Microlithography XIII **3677**: 640-649.

[Desai90] Desai, V. and L. Reimer (1990). "Calculation of the Signal of Backscattered Electrons Using a Diffusion Matrix from Monte Carlo Calculations." Scanning **12**: 1-4.

[Ding96] Ding, Z. and R. Shimizu (1996). "A Monte Carlo modeling of electron interaction with solids including cascade secondary electron production." Scanning **18**(2): 92-113.

[Ding01] Ding, Z. J., X. D. Tang and R. Shimizu (2001). "Monte Carlo study of secondary electron emission." Journal of Applied Physics **89**(1): 718-726.

[Dongmo00] Dongmo, L. S., J. S. Villarrubia, S. N. Jones, et al. (2000). "Experimental Test of Blind Tip Reconstruction for Scanning Probe Microscopy." Ultramicroscopy **85**(3): 141-153.

[Drouin99] Drouin, D. and B. J. Griffin (1999). Measurement and Application of the Detector Quantum Efficiency of Se Detectors Used in SEM, FESEM, and ESEM. AMAS V - The Fifth Biennial Symposium, Australia, The University of Sydney.

[Firsova91] Firsova, A. A., L. Reimer, N. G. Ushakov, et al. (1991). "Comparison of a Simple Model of BSE Signal Formation and Surface Reconstruction with Monte Carlo Calculations." Scanning **13**: 363-368.

[Fritsch95] Fritsch, D. S., E. L. Chaney, et al. (1995). "Core-based portal image registration for automatic radiotherapy treatment verification." International Journal of Radiation Oncology, Biology, Physics 33(5): 1287-1300.

[Garcia-Lekue03] Garcia-Lekue, A. and J. M. Pitarke (2003). "Surface effects on the electronic energy loss of charged particles entering a metal surface." Journal of Electron Spectroscopy and Related Phenomena **129**: 223-227.

[Gerig87] Gerig, G. (1987). "Linking Image-Space and Accumulator-Space: A New Approach for Object-Recognition." IEEE (maybe Transactions Computing): 112-117.

[Gilbert92] Gilbert, J. C. and J. Nocedal (1992). "Global Convergence Properties of Conjugate Gradient Methods for Optimization." SIAM Journal on Optimization **2**: 21-42.

[Goldstein92] Goldstein, J. I., D. E. Newbury and P. Echlin (1992). Scanning Electron Microscopy and X-Ray Microanalysis, Plenum Pub Corp.

[Hadjiiski96] Hadjiiski, L., S. Munster, E. Oesterschulze, et al. (1996). "Neural network correction of nonlinearities in scanning probe microscope images." Journal of Vacuum Science Technology B **14**(2): 1563-1568.

[Haralick87] Haralick, R. M., S. R. Sternberg and X. Zhuang (1987). "Image Analysis Using Mathematical Morphology." IEEE Transactions on Pattern Analysis and Machine Intelligence **9**(4): 532-550.

[Haralick81] Haralick, R. M. and L. Watson (1981). "A Facet Model for Image Data." Computer Graphics and Image Processing **15**: 113-129.

[Hendriks03] Hendriks, C. L. L. and L. J. van Vliet (2003). Basic morphological operations, band-limited images and sampling. 4th International Conference, Scale-Space 2003, Isle of Skye, UK, Springer, Berlin.

[Hoel71] Hoel, P. G., S. C. Port and C. J. Stone (1971). Introduction to Statistical Theory. Boston, Houghton Mifflin Company.

[Horn70] Horn, B. (1970). Shape from Shading: A Method for Obtaining the Shape of a Smooth Opaque Object from One View. Department of Electrical Engineering. Cambridge, MA, MIT.

[Hough62] Hough, P. V. C. (1962). Method and means for recognizing complex patterns. U.S. Patent 3,069,654.

[Hovington97a] Hovington, P., D. Drouin and R. Gauvin (1997). "CASINO: a new Monte Carlo code in C language for electron beam interaction. I. Description of the program." Scanning **19**(1): 1-14.

[Hovington97b] Hovington, P., D. Drouin, R. Gauvin, et al. (1997). "CASINO: a new Monte Carlo code in C language for electron beam interactions. III. Stopping power at low energies." Scanning **19**(1): 29-35.

[Hovington02] Hovington, P., D. Drouin, R. Gauvin, et al. (2002). CASINO (software). Université de Sherbrooke, Québec, Canada.

[Ikeuchi81] Ikeuchi, K. and B. K. P. Horn (1981). "Numerical Shape from Shading and Occluding Boundaries." Artificial Intelligence **17**: 141-184.

[Illingworth88] Illingworth, J. and J. Kittler (1988). "A Survey of the Hough Transform." CVGIP **44**: 87-116.

[Jain89] Jain, A. K. (1989). Fundamentals of Digital Image Processing, Prentice-Hall.

[Jenkinson00] Jenkinson, M. and S. Smith (2000). Optimisation in Robust Linear Registration of Brain Images, TR00MJ2, FMRIB Image Analysis Group.

[Jollife86] Jollife, I. T. (1986). Principal Components Analysis. New York, Springer-Verlag.

[Jones94] Jones, A. G. and C. J. Taylor (1994). "Robust shape from shading." Image and Vision Computing **12**(7): 411-421.

[Joy94] Joy, D. C. (1994). A Database On Electron-Solid Interactions. Scanning **17**: 270–275

[Joy95] Joy, D. C. (1995). Monte Carlo Modeling for Electron Microscopy and Microanalysis. New York, Oxford University Press.

[Katsaggelos89] Katsaggelos, A. K. (1989). "Iterative image restoration algorithms." SPIE Optical Engineering **28**(7): 735-748.

[Kleyman98] Kleyman, S. and D. Livshitz (1998). Computer Vision Facet Model.

[Kotera90] Kotera, M., R. Ijichi, T. Fujiwara, et al. (1990). "A simulation of electron scattering in metals." Japanese Journal of Applied Physics, Part 1 (Regular Papers & Short Notes) **29**(10): 2277-2282.

[Lagendijk91] Lagendijk, R. L. and J. Biemond (1991). Iterative Identification and Restoration of Images. Boston/Dordrecht/London, Kluwer.

[Leclerc91] Leclerc, Y. G. and A. F. Bobick (1991). The direct computation of height from shading. Computer Vision and Pattern Recognition, Maui, HI, IEEE Computer Society.

[Lee98] Lee, T. (1998). "Segmenting Images Corrupted by Correlated Noise." IEEE Transactions on Pattern Analysis and Machine Intelligence **20**: 481-492.

[Liu00] Liu, G., J. Nocedal and R. Waltz (2000). CG+.

[Lorenzen04] Lorenzen, P., B. Davis, et al. (2004). "Model Based Symmetric Information Theoretic Large Deformation Multi-Modal Image Registration." Proceedings of IEEE International Symposium on Biomedical Imaging (ISBI): 720-723.

[Lowney95] Lowney, J. R. (1995). "Monsel-II: Monte Carlo Simulation of Sem Signals For Linewidth Metrology." Microbeam Analysis **4**(3): 131-136.

[Lowney96a] Lowney, J. R. (1996). "Application of Monte Carlo simulations to critical dimension metrology in a scanning electron microscope." Scanning Microscopy **10**(3): 667-78.

[Lowney96b] Lowney, J. R. (1996). "Monte Carlo Simulation of Scanning Electron Microscope Signals for Lithographic Metrology." Scanning **18**: 301-306.

[Ly95] Ly, T. D., D. G. Howitt, M. K. Farrens, et al. (1995). "Monte Carlo calculations for specimens with microstructures." Scanning **17**(4): 220-6.

[Maes97] Maes, F., A. Collignon, D. Vandermeulen, et al. (1997). "Multi-Modality Image Registration By Maximization of Mutual Information." IEEE Transactions on Medical Imaging **16**(2): 187-198.

[Magonov96] Magonov, S. N. and M.-H. Whangbo (1996). Surface Analysis with STM and AFM, Experimental and Theoretical Aspects of Image Analysis. Weinheim, VCH.

[Maintz98] Maintz, J. B. A. and M. A. Viergever (1998). "A Survey of Medical Image Registration." Medical Image Analysis **2**(1): 1-37.

[Marr80] Marr, C. and E. Hidreth (1980). "Theory of Edge Detection." Proceedings of the Royal Society London **B207**: 187-217.

[Mohamed95] Mohamed, M. A. and J. Saniie (1995). "Statistical evaluation of sequential morphological operations." IEEE Transactions on Signal Processing **43**.

[Morales93] Morales, A. and R. Acharya (1993). "Statistical Analysis of Morphological Openings." IEEE Transactions on Signal Processing **41**(10): 3052-3056.

[Moré94] Moré, J. J. and D. J. Thuente (1994). "Line Search Algorithms with Guaranteed Sufficient Decrease." ACM Transactions on Mathematical Software **20**(3): 286-307.

[Musgrave90] Musgrave, F. K. (1988). Grid Tracing: Fast Ray Tracing for Height Fields, Yale University Dept. of Computer Science Research.

[Peleg90] Peleg, S. and G. Ron (1990). "Nonlinear Multiresolution: a Shape from Shading Example." IEEE PAMI **12**(12).

[Pingali92] Pingali, G. S. and R. Jain (1992). "Restoration of scanning probe microscope images." IEEE Workshop Applied Computer Vision: 282-289.

[Postek03] Postek, M. T., J. S. Villarrubia and A. E. Vladar (2003). A Simulation Study of Repeatability and Bias in the CD-SEM. Metrology, Inspection, and Process Control for Microlithography XVII.

[Press95] Press, W., S. Teukolsky, W. Vetterling, et al. (1995). Numerical Recipes in C, Cambridge University Press.

[Reimer85] Reimer, L. (1985). Scanning electron microscopy : physics of image formation and microanalysis. Berlin ; New York, Springer-Verlag.

[Reimer87] Reimer, L., R. Böngeler and V. Desai (1987). "Shape From Shading Using Multiple Detector Signals in Scanning Electron Microscopy." Scanning Microscopy **1**(3): 963-973.

[Robin03] Robin, F., A. Orzati, E. Moreno, et al. (2003). "Simulation and Evolutionary Optimization of Electron-Beam Lithography With Genetic and Simplex-Downhill Algorithms." IEEE Transactions on Evolutionary Computation **7**(1): 69-82.

[Robini97] Robini, M. C., T. Rastello, D. Vray, et al. (1997). Space-variant deconvolution for synthetic aperture imaging using simulated annealing. International Conference on Image Processing, Washington, DC, IEEE.

[Rohr99] Rohr, K. (1999). "Approximating Thin-Plate Splines for Elastic Registration: Integration of Landmark Errors and Orientation Attributes." Proceedings of IPMI '99: 252-265.

[Romer00] Romer, J., M. Plaschke and J. I. Kim (2000). "Alignment of AFM images using an iterative mathematical procedure." Ultramicroscopy **85**: 99-105.

[Russell01] Russell, P. and D. Batchelor (2001). "SEM and AFM: Complementary Techniques for Surface Investigations." Microscopy and Analysis(49): 5-8.

[Sabatier00] Sabatier, P. C. (2000). "Past and future of inverse problems." Journal of Mathematical Physics **41**(6): 4082-4124.

[Sarid94] Sarid, D. (1994). Scanning Force Microscopy, with Applications to Electric, Magnetic and Atomic Forces. New York, Oxford University Press.

[Schneir96] Schneir, J., J. S. Villarrubia, T. H. McWaid, et al. (1996). "Increasing the value of atomic force microscopy process metrology using a high-accuracy scanner, tip characterization, and morphological image analysis." Journal of Vacuum Science & Technology B (Microelectronics and Nanometer Structures) **14**(2): 1540-6.

[Seeger03] Seeger, A., C. Fretzagias and R. M. Taylor II (2003). "Software Acceleration Techniques for the Simulation of SEM Images." Scanning **25**(5): 264-273.

[Shewchuk94] Shewchuk, J. R. (1994). An Introduction to the Conjugate Gradient Method Without the Agonizing Pain.

[Shimizu76] Shimizu, R., Y. Kataoka, T. Ikuta, et al. (1976). "A Monte Carlo approach to the direct simulation of electron penetration in solids." Journal of Physics D: Applied Physics **9**(1): 101-113.

[Sokolov97] Sokolov, I. Y., G. S. Henderson and F. J. Wicks (1997). "The contrast mechanism for true atomic resolution by AFM in non-contact mode: quasi-non-contact mode?" Surface Science **381**: L558-L562.

[Studholme97] Studholme, C. (1997). Measures of 3D Medical Image Alignment. Computational Imaging Science Group. London, University of London.

[Todd01] Todd, B. A. and S. J. Eppell (2001). "A method to improve the quantitative analysis of SFM images at the nanoscale." Surface Science **491**: 473-483.

[Topometrix95] Topometrix (1995). SPMLab 4.0 User Manual. Santa Clara, CA, Topometrix Corporation.

[Ukraintsev03] Ukraintsev, V. (2003). Effect of bias variation on total uncertainty of CD measurements. Metrology, Inspection and Process Control for Microlithography XVII.

[Varadhan02] Varadhan, G., W. Robinett, D. Erie, et al. (2002). Fast Simulation of Atomic-Force-Microscope Imaging of Atomic and Polygonal Surfaces Using Graphics Hardware. SPIE Conference on Visualization and Data Analysis 2002, San Jose, CA, SPIE.

[Villarrubia94] Villarrubia, J. S. (1994). "Morphological estimation of tip geometry for scanned probe microscopy." Surface Science **321**(3): 287-300.

[Villarrubia97] Villarrubia, J. S. (1997). "Algorithms for Scanned Probe Microscope Image Simulation, Surface Reconstruction, and Tip Estimation." Journal of Research of the National Institute of Standards and Technology **102**(4): 425-454.

[Villarrubia04] Villarrubia, J. S., A. E. Vladár, et al. (2004). "Dimensional Metrology of Resist Lines using a SEM Model-Based Library Approach." Metrology, Inspection, and Process Control for Microlithography XVIII Proceedings of SPIE Vol 5375: 199-209.

[Viola95] Viola, P. A. (1995). Alignment by Maximization of Mutual Information. Artificial Intelligence Lab. Cambridge, MA, MIT**:** 150.

[Wang95] Wang, D., V. Haese-Coat, A. Bruno, et al. (1995). "Some Statistical Properties of Mathematical Morphology." IEEE Transactions on Signal Processing **43**(8): 1955-1965.

[Wei97] Wei, G. Q. and G. Hirzinger (1997). "Parametric shape-from-shading by radial basis functions." IEEE Transactions on Pattern Analysis and Machine Intelligence **19**(4): 353-365.

[Williams98] Williams, P. M., M. C. Davies, C. J. Roberts, et al. (1998). "Noise-compliant tip-shape derivation." Applied Physics A **66**: S911-S914.

[Williams96] Williams, P. M., K. M. Shakesheff, M. C. Davies, et al. (1996). "Blind reconstruction of scanning probe image data." Journal of Vacuum Science Technology B **14**(2): 1557-1562.

[Yang93] Yang, C. (1993). “Efficient Stochastic Algorithms on Locally Bounded Image Space.” CVGIP: Graphical Models and Image Processing **55**(6): 494-506.

[Zangwill88] Zangwill, A. (1988). Physics at Surfaces, Cambridge University Press.