

Fast Atomic Force Microscopy Imaging using Self-Intersecting Scans and Inpainting

Travis Meyer, Rodrigo Farnham, Nen Huynh,
Advisors: Alex Chen, Jen-Mei Chang, Andrea Bertozzi

August 5, 2011

Abstract

This is the paper's abstract ...

1 Introduction

Although much research has been directed towards improving the precision and resolution of atomic force microscopes (AFMs), recently interest in speeding up scans has intensified. Fast scans would enable the imaging of dynamic processes - such as oxidation or DNA replication - thus opening up a new dimension for scientific inquiry. Current state-of-the-art atomic force microscopy involves raster scanning the sample and removing noise from the image on a scan line basis. Specifically, an AFM scans a sample through minute force interactions between its probe and the sample. A laser shines on the needle cantilever and is reflected onto a photodiode, whereupon a detector records deviations. The scanning mechanism can be one of three methods: the sample surface moves while the cantilever is fixed, the cantilever moves while the sample surface is fixed, or both the cantilever and the sample surface are actuated. A signal is recovered from these interactions and used to construct an image of the topology of the sample.

AFMs have the advantage of not requiring special environments, such as being in a vacuum, allowing for in-situ experiments of evolving complexes, such as living organisms or chemical reactions. Figure 2 shows the basic elements of how an AFM collects data. However, the path taken by the scan dramatically affects speed and accuracy. For example, sharp turns tend to damage the probe and elicit unwanted mechanical vibrations. More information about AFMs can be found in [5]. We propose an entire class of scans conducive to expedient data collection, as well as novel processing techniques. In particular, we explore both cycloid and double Archimedean spiral scan paths and introduce multiple drift removal methods. Importantly, these methods generalize to any self-intersecting path and are therefore applicable to any programmable AFM.

2 Scan Types

The signal $S(t)$ generated by the detector in the AFM does not directly give the sample topology, but rather the topology information is distorted so the signal must be post-processed. We model the signal as:

$$S(t) = I(x(t), y(t)) + T(x(t), y(t)) + D(t) + \chi(t) + \eta(t)$$

The sample topology can be thought of as an image $I(x, y) : \Omega \rightarrow \mathbb{R}$. The signal from the detector is distorted from the tilt $T(t)$, drift $D(t)$, streaks $\chi(t)$, and noise $\eta(t)$ that occur during the scanning process.

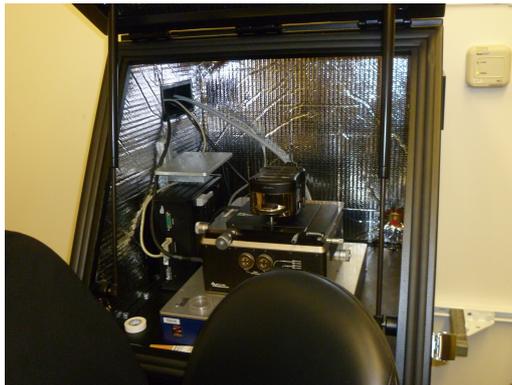
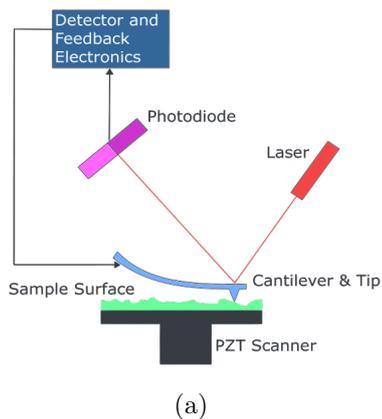


Figure 1: (a) The elements of an AFM. Figure from [2]. (b) A picture of an AFM setup.

As the probe scans the surface, the path $\phi(t) = (x(t), y(t))$ collects image information, $I(x(t), y(t))$. Because of the small scales involved, a minuscule tilt in the mounting assembly has a major effect on the signal. Tilt $T(x, y)$ is represented as a plane in \mathbb{R}^3 . The probe is also sensitive to thermal drift caused by fluctuations in the temperature of the machine. Since the drift is locally approximately linear, drift $D(t)$ is modelled as a continuous function with small second derivative. Moreover, during the scanning process, sudden changes to the signal such as the tip of the probe breaking or a sudden movement in the machine may occur as seen in Figure 2(b). These streaks $\chi(t)$ are modelled as simple functions of finite discontinuity. Finally, the background noise $\eta(t)$ is modelled as a Gaussian distribution of small magnitude. For the remainder of the paper, unless otherwise mentioned, we assume $\chi(t)$ and $\eta(t)$ to be 0.

To scan a surface, a path of the probe relative to the sample must be specified. Typically the AFM is set to scan from top to bottom with horizontal scan lines in a process called raster scanning. There are drawbacks in raster scanning which will be mentioned. In our work, we found that self-intersecting curves can produce better images at faster time than raster scan. In fact, self-intersections facilitate removal of drift distortions. As a consequence, the modes of distortion can be segregated and dealt with individually. Specifically, we use a cycloid scan and double Archimedean spiral scan as a demonstration of this process.

In general, any scan bearing self-intersections can be exploited. These self-intersections allow for sparse data collection while simultaneously enabling the removal of distortions inherent in AFM machines.

2.1 Raster Scan

The raster scan collects data top-down with horizontal lines. Each horizontal line has a different average intensity as can be seen in Figure 2(a). To fix the distortion due to tilt and drift, a line-fit is removed from each horizontal scan. This is called line-flattening. Figure 2(b) shows the image after line-flattening. Line-flattening produces an image where each horizontal line has the same mean. This is undesirable when horizontal sections have objects or backgrounds of different intensity. Figure 3 shows the simulated distortion and drawbacks of using line-flattening: if there is a contrast in intensity between the left and right side of the image window, intensity distortion appears around the boundary of the contrasting regions, and if a horizontal line has fixed intensity, the feature can disappear into the background.

Because of physical limitations (see [6] for detail), the scan speed is hindered by the resonance produced during sharp direction changes of the probe that occur between scan lines of the raster scan [14].

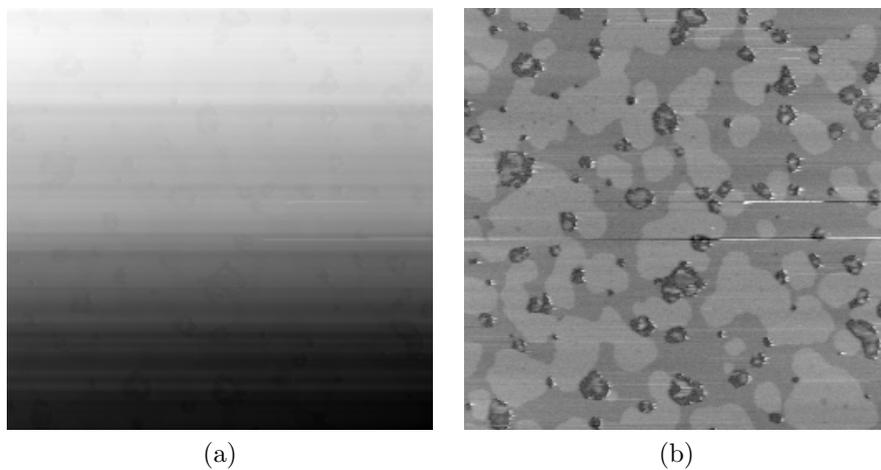


Figure 2: (a) The raster scanned signal from the AFM. (b) The horizontal scans are line-flattened. Notice the streaks which appear as out-of-place horizontal lines in the middle of the image.

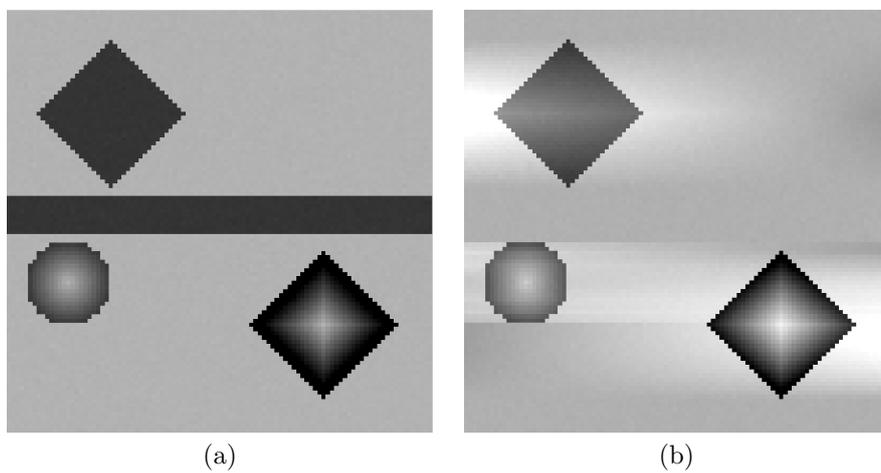


Figure 3: (a) A simulated image. (b) Line-flattened of (a). Notice the distortion in the foreground and background around objects.

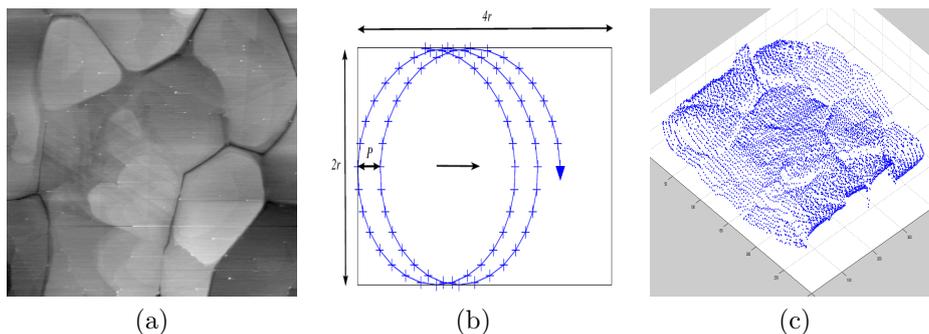


Figure 4: (a) Annealed gold raster scan image. Notice the bright dots artifacts. (b) The scan path of a cycloid scan. Figure from [14]. (c) A simulated data collection from (a) using a cycloid scan.

The speed limitation of raster scanning has led to the consideration of alternative scan paths. For example, because images with few features do not need to be thoroughly scanned, attempts have been made at scanning only salient locations and filling in the remainder. These include boundary tracking [6] and raster scan sub-sampling [3]. The missing regions are then inpainted with standard image processing techniques. Attempts at boundary tracking have been problematic because on-the-fly change detection used to identify boundaries is required but tilt, drift, and streaks conspire to make this very difficult.

2.2 Cycloid Scan

Yong et al. [14] proposed a cycloid path to sample points without making sharp turns. This allows for a faster scan. To hasten the data collection, one can modify parameters to sample fewer points and use inpainting to fill in the missing information.

The cycloid path, however, samples fewer points in the middle of the scanning window compared to the top and bottom of the window. Generally, sampling paths center about a feature. This implies the middle of the scanning window ought to be scanned at least as much as the outer regions of said window. The cycloid scan is therefore suboptimal for typical sparse surveys. Still, the ease of implementation makes cycloid scans desirable. See Figure 4 for an example.

2.3 Double Archimedean Spiral Scan

When inpainting, it is advantageous to have a small distance between regions of unknown data to regions of known data. Hence, scans that exhibit this are desirable. The question of optimal scan path can then be written as the minimization of the energy functional $E(\phi)$ where ϕ is a path of fixed length $L > 0$, Ω is the domain of the image, $d(\cdot, \cdot)$ is the L_2 distance and

$$E(\phi) = \int_{\Omega} \min_t d(z, \phi(t)) dz \quad (1)$$

Using simulated annealing on E and approximating the x and y components of ϕ by polynomials, one finds that the curve that minimizes this functional is an approximate Archimedean spiral. This spiral has the advantage of near uniform sampling. To elicit self-intersections, a double Archimedean spiral can be used. The parametrization equation is $\phi : [-2k\pi - \pi/2, 2k\pi + \pi/2] \rightarrow \mathbb{R}^2$ where $\phi(t) = (at \cos t, at \sin t)$ with a being the distance between successive turns and k being half the number of turns ($k = 4$ means 8 turns). We use $\phi(t) = (at^{1/2} \cos t^{1/2}, at^{1/2} \sin t^{1/2})$ with appropriate domain for approximately consistent AFM probe speed.

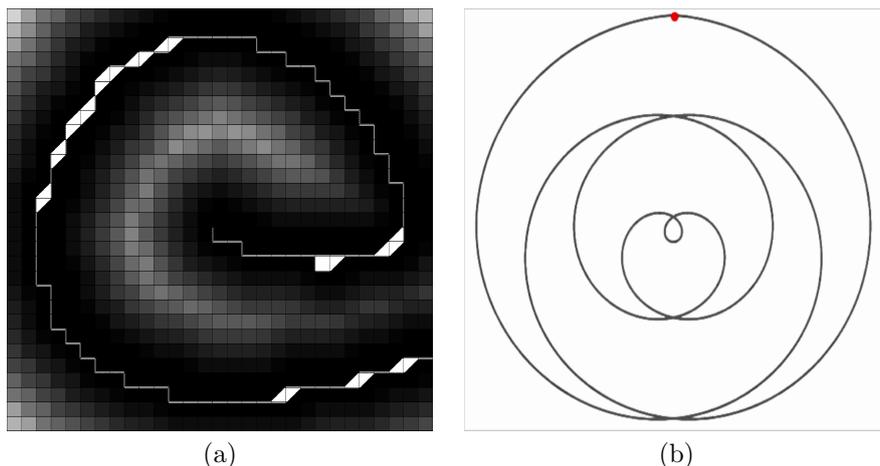


Figure 5: (a) A result using simulated annealing to minimize energy functional (Eq. 1). The white curve is the resulting curve and the grayscale represents the distance of a given pixel to nearest curve point (dark means close and light means far). (b) The double Archimedean spiral starts from the outside (the marked location) moving counter-clockwise inward, reaching the center and then moving outward until it reaches its starting point.

2.4 Double Archimedean Spiral Scan vs. Cycloid Scan

The double Archimedean spiral scan has the advantage of having lower energy value of Equation (1) than the cycloid scan. However, the disparate time intervals between intersection of the cycloid scan allow for more accurate drift removal (See Figure 6). This means that the spiral has more uniform covering of the scanned space, but produces less accurate image reconstruction, whereas cycloid scan enables more accurate drift removal, producing a better image, but is slower and samples less uniformly.

3 Removal of Signal Distortion using Self-Intersection

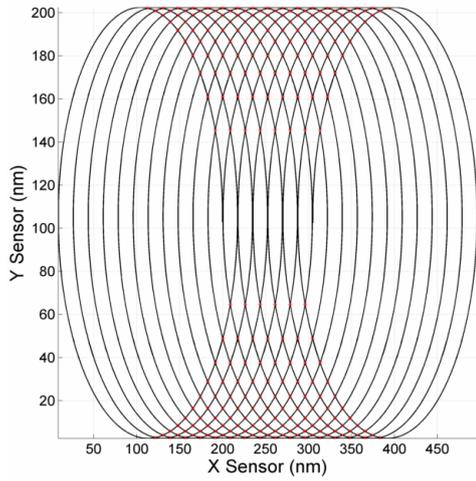
The motivation for self-intersecting curves is to enable the removal of drift and potentially facilitate streak removal. As the probe moves through the sample, drift accumulates but there is no physical way to measure drift intensity. However, because drift is continuous and changes slowly, one can approximate it given very scant information. Repeated observations at intersections provide a proxy to the underlying thermal drift.

3.1 Finding Intersections

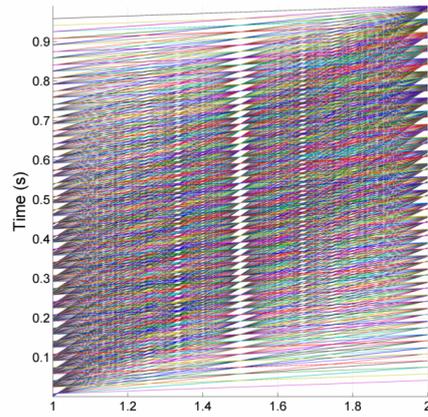
Before drift correction can be performed, the intersections must be found. We seek to find the points at which an explicitly discretized curve in \mathbb{R}^2 intersects itself. If the curve is composed of n connected line segments, then the calculation can be done in $O(n^2)$ by checking all line segments against all other line segments for intersections. The expected running time can be reduced to $O(n \log(n))$ by spatially sorting the points in the curve using a quadtree [7] structure.

Let r be the maximum length of all line segments in our discrete curve. For each line segment connecting some points \vec{a} and \vec{b} , we generate a list of curve points that are within a distance $2r + \epsilon$ of \vec{a} where ϵ need only be large enough to be such that in machine precision, $2r + \epsilon$ is distinct from $2r$. In practice, $\epsilon = r/100$.

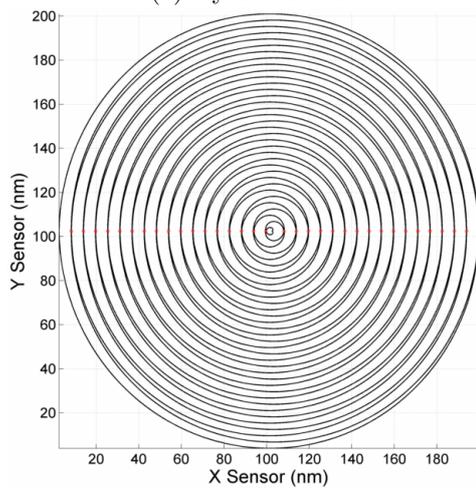
We associate with each point \vec{x} collected in this neighborhood about \vec{a} the line segment which begins at \vec{x} and ends at some \vec{y} . If \vec{x} is visited before \vec{a} we ignore the possibility of them



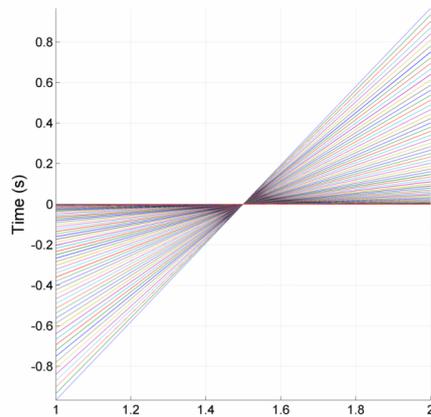
(a) Cycloid Path



(c) Cycloid Bundle



(b) Spiral Path



(d) Spiral Bundle

Figure 6: (a)-(b) The cycloid and Archimedean spiral. (c)-(d) Bundle plot: The left and right side indicates when an intersection is encountered and the connecting lines show the two times that the path intersected at the same location.

intersecting - when \vec{x} is visited then their intersection will be found. Thus in the case that \vec{x} is visited after \vec{a} , we check for an intersection between the line segments $\vec{x} \rightarrow \vec{y}$ and $\vec{a} \rightarrow \vec{b}$. The largest issue with this method is that the performance of the algorithm is very closely related to r . Assuming that the samples are taken at regular intervals, the restriction that the tip velocity is small naturally enforces that r remains constant. Erroneous data and sudden tip movements can cause the algorithm to perform poorly, but properly culling bad data points effectively eliminates this concern.

3.2 Removal of Drift

As a baseline comparison for our drift removal methods, we implemented a higher degree analogue of line-flattening called poly-flattening. In this method, (relatively) low degree polynomials are fitted to and removed from each piecewise arch of the scan. Like line flattening, polynomial flattening introduces artifacts and distortions to the image, even as much as destroying data and contrast similar to Figure 3. Furthermore, on intersecting curves, some points are ill-defined, leading to very poor image reconstruction.

Given the images in figure 7, it may seem as though intersections are undesirable from the standpoint of reconstruction, however they play an instrumental role in our proposed robust thermal drift removal. We expect the AFM readings to agree on intersections; any measured difference provides us with a glimpse of the behavior of the drift.

For (a_i, b_i) s.t. $x(a_i) = x(b_i)$ and $y(a_i) = y(b_i)$, we have $S(b_i) - S(a_i) = D(b_i) - D(a_i) = \Delta_i$.

Armed with such information, we can recover the original thermal drift (up to a constant) and remove it from our signal. Our approach was that of expressing the thermal drift in a convenient basis and minimizing the least squares error between observed differences and predicted differences. We explored differences fitting of polynomial, trigonometric, and spline bases (B-splines [1]), the latter being the most robust.

For simplicity, we derive our result using a general basis:

$$\begin{aligned}\tilde{D}(t) &= \sum_{k=1}^n \alpha_k \phi_k(t) \\ \tilde{D}(b_i) - \tilde{D}(a_i) &= \sum_{k=1}^n \alpha_k \{\phi_k(b_i) - \phi_k(a_i)\} \\ \mathfrak{D}(a_i, b_i) &= \sum_{k=1}^n \alpha_k \Phi_k(a_i, b_i) = [\Phi_1(a_i, b_i) \cdots \Phi_n(a_i, b_i)] \alpha\end{aligned}$$

where $\mathfrak{D}(a_i, b_i) = \tilde{D}(b_i) - \tilde{D}(a_i)$, $\Phi_k(a_i, b_i) = \phi_k(b_i) - \phi_k(a_i)$ and $\alpha = [\alpha_1 \cdots \alpha_n]^T$. Then the least squares error is given by

$$E = \|\Delta - \Phi \alpha\|^2 \text{ for } \Delta = [\Delta_1 \cdots \Delta_\ell]^T \text{ and } \Phi_{ij} = \Phi_j(a_i, b_i)$$

The optimal α is the solution to $\Phi^T \Phi \alpha = \Phi^T \Delta$. Furthermore, one may wish to control the smoothness of the spline fit. To that end a penalty term is added to our error:

$$\begin{aligned}\tilde{D}''(t) &= [\phi_1''(t) \cdots \phi_n''(t)] \alpha \\ \int_0^T [\tilde{D}''(t)]^2 dt &= \int_0^T \alpha^T [\phi_1'' \cdots \phi_n'']^T [\phi_1'' \cdots \phi_n''] \alpha dt = \alpha^T M \alpha \text{ where } M_{ij} = \int \phi_i'' \phi_j'' dt\end{aligned}$$

so that $E = \|\Delta - \Phi \alpha\|^2 + \lambda \alpha^T M \alpha$ and α is the solution to $(\Phi^T \Phi + \lambda M) \alpha = \Phi^T \Delta$.

In any case, solving for the optimal coefficients is a simple matter of solving a linear system. The specific Φ matrices are given below:

$$\Phi_{\text{poly}} = [(b_i)^j - (a_i)^j]_{ij}$$

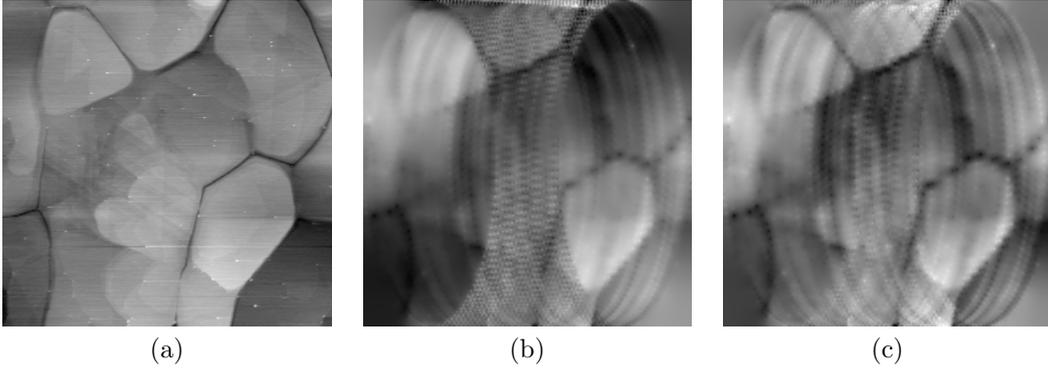


Figure 7: (a) Original Image. (b) Image is scan simulated, tilt removed, drift removed, and then inpainted using TV. (c) Image is scan simulated, drift subtracted, tilt removed and then inpainted using TV. Notice the distortions still left in (b) compared to (c).

$$\Phi_{\text{trig}} = [e^{\sqrt{-1}jb_i} - e^{\sqrt{-1}ja_i}]_{ij} \quad (-n \leq j \leq n, j \neq 0)$$

$$\Phi_{\text{spline}} = [N_{j,\text{degree}}(b_i) - N_{j,\text{degree}}(a_i)]$$

Both polynomial fitting and trigonometric fitting suffer from numerical instabilities stemming from the inclusion of higher order terms. Additionally, the former has a tendency to blow up around the edges of known data. Spline fitting, on the other hand, addresses both these issues. The only caveat is that the recursive definition of B-splines does not lend itself to speedy evaluation. However a dynamic programming implementation can achieve comparable running times to polynomial and trigonometric fitting.

3.3 Removal of Tilt

For tilt removal, the surface data $A(x, y)$ is least-squares plane-fitted $P(x, y)$ and the plane-fit is subtracted from the surface data ($A(x, y) - P(x, y)$). The removal of tilt can occur before drift removal or after. However, on certain scan types, drift removal should occur first. For example, the cycloid scan goes left to right so if there is an overall increase in drift, the effect on the signal is similar to the effect of tilt. One can see this effect from figure 7.

3.4 Detection of Streaks

Streaks appear when the tip/sample interaction suffers a regime change, so one method of detection is to find these large changes. After removal of drift and tilt, the signal is Gaussian smoothed so that approximation of derivatives is not affected by noise. The derivative is calculated and points having high magnitude derivative are then marked. The above requires a threshold, which may be user-specified or dynamically chosen, as in Canny edge detection. To refine the marking of points, one can use a 1-D version of non-maximum suppression also similar to Canny edge detection [4].

False positives are common on edges of objects and artifacts such as those shown in Figure 4. Thus, this method of streak detection is not ideal for removing streaks. It is hoped that after detection of the streaks, a secondary check can be used to confirm or reject these marks using nearby scanned values. Currently, there is no algorithm that can do such confirmation and no algorithm that removes these streaks.

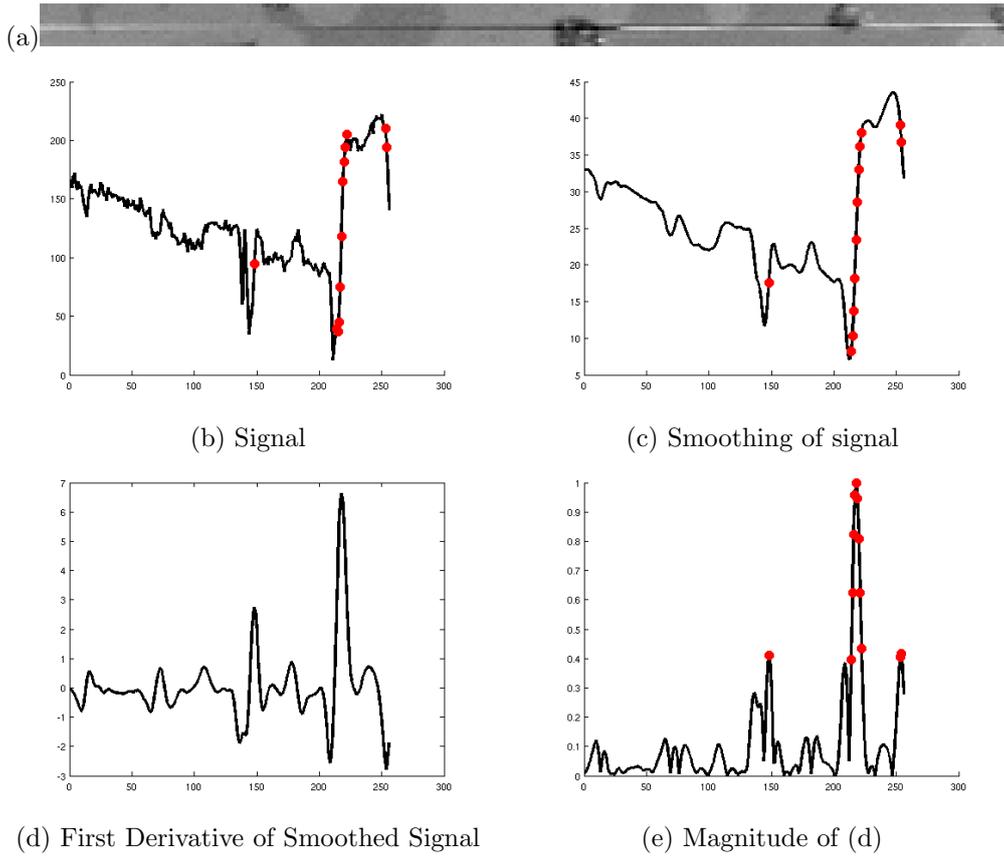


Figure 8: (a) The streak from Figure 2 (b) represented as a signal is shown in (a). (c) The signal is then Gaussian smoothed, and (d) the derivative is calculated. (e) the red marks label the points with magnitude derivative larger than a specified threshold.

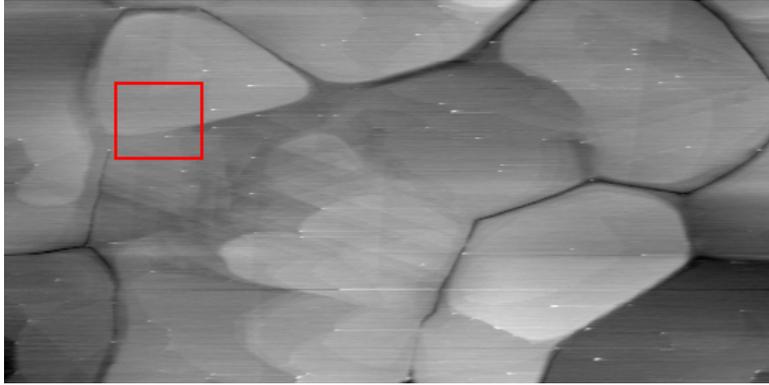


Figure 9: Image with overlapping matrix B

4 Inpainting

The literature is replete with many different inpainting algorithms. For comparison, a mention of literature inpainting methods is provided on a simulated scan of a nanowire. We then introduce a new inpainting method called Penalized Dictionary Inpainting.

4.1 Brief comparison of literature inpainting methods

We experimented with a few methods for inpainting the sparse height data. Foremost is total variation minimization inpainting, or TV inpainting, which is explained in [11]. TV performs generally well with the data provided. The downside to TV is that edges which have width less than the spacing of the scan arcs are not completed, and the initial edge data there is simply ignored. Two more methods which do not suffer from this effect that we considered are absolutely minimizing Lipschitz extensions (AMLE) [13] and low-curvature image simplifiers (LCIS) [12]. Both of these methods perform quite well, with LCIS generally performing the best. Unfortunately, the algorithms used for AMLE and LCIS do not compare, in speed, to the performance of fast Bregman iterative methods for TV inpainting [9].

4.2 Penalized Dictionary Inpainting

The inpainting methods introduced above are not well-suited to the inpainting of sparse data. They require large known regions, so that data can be well propagated into the unknown region. But the data collected from the AFM is preferably sparse so that minimal scanning is required. Thus, our data requires an inpainting method that can deal with sparse data. So naturally we set out to create our own inpainting algorithm to address the present issues. Some degree of blurring is necessary to fill in large missing regions, but such approach is not kind to edges. To fill in missing edges the inpainting method uses nonlocal techniques. The inspiration comes from nuclear norm inpainting [10] and a work on the Netflix prize [8].

An image is comprised of many (overlapping) $k \times k$ neighborhoods.

$$B = [\vec{B}_1 \cdots \vec{B}_\ell]^T$$

We begin with the (nonlocal) assumption that there exists a basis which can succinctly describe these neighborhoods. By intelligently constructing the basis and neighborhood representations, we can proceed to nonlocally inpaint the missing regions. This is similar to the nuclear norm inpainting of Travis Meyer [10]. It is also akin to the process of making movie recommendations from sparse information [8]. However, the inherent geometric constraints of

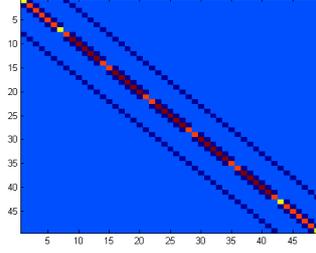


Figure 10: A color representation of matrix M

an image are not enforced by these methods. To that end we introduce a penalty term to our error.

$$\tilde{B} = PV$$

$$E = \|B - PV\|^2 + \lambda \text{Penalty}$$

Penalty = $\sum_{i=1}^{\ell} \sum_{j \in N_i} a_{ij} P P^T a_i^T$ where N_i is the relevant index set, roughly $\{i+1, i-1, i+m, i-m\}$ modulo boundary cases.

$$\frac{\partial \text{Penalty}}{\partial P_i} = 2 \sum_j a_{ij} P = 2A_i P$$

$$\frac{\partial \text{Penalty}}{\partial P} = 2MP \text{ where } M = [A_1 \cdots A_\ell]^T$$

$$\frac{\partial E}{\partial P} = 2(B - PV)V^T + 2\lambda MP$$

$$\frac{\partial E}{\partial P} = 2P^T(B - PV)$$

$$P_{k+1} = \tau \text{error}_k V_k^T + \lambda M P_k$$

$$V_{k+1} = \tau P_k \text{error}_k$$

where error is the difference $B - PV$ restricted to the known region, a_{ij} is a row vector

$$a_{ij} = \begin{cases} 1, & \text{at } i^{\text{th}} \text{ component} \\ -1, & \text{at } j^{\text{th}} \text{ component} \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

The lambda term penalizes differences between "adjacent" windows. To wit, it is the (approximate) vector derivative. Interestingly, if the window size is restricted to a single pixel, the algorithm recovered is not unlike the heat equation.

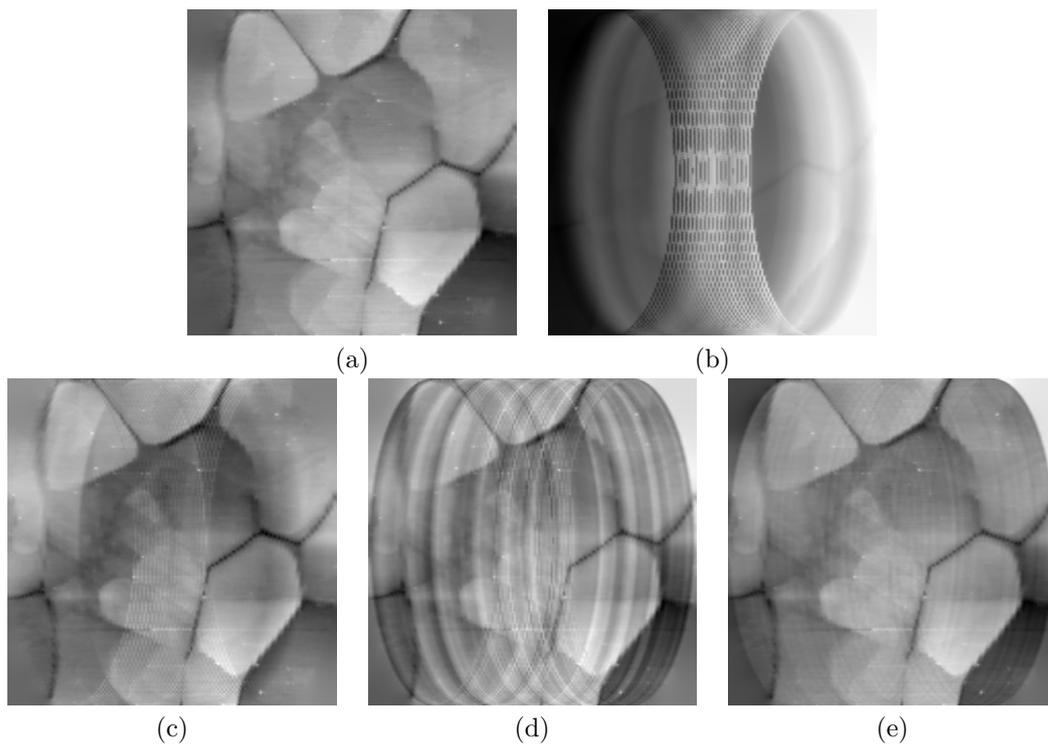


Figure 11: Simulations of drift removal. (a) The original AFM image. (b) The signal generated from the simulated scan. (c) Image generated with poly-flattening drift removal. (d) Image generated with poly-difference-flattening drift removal. (e) Image generated with spline drift removal.

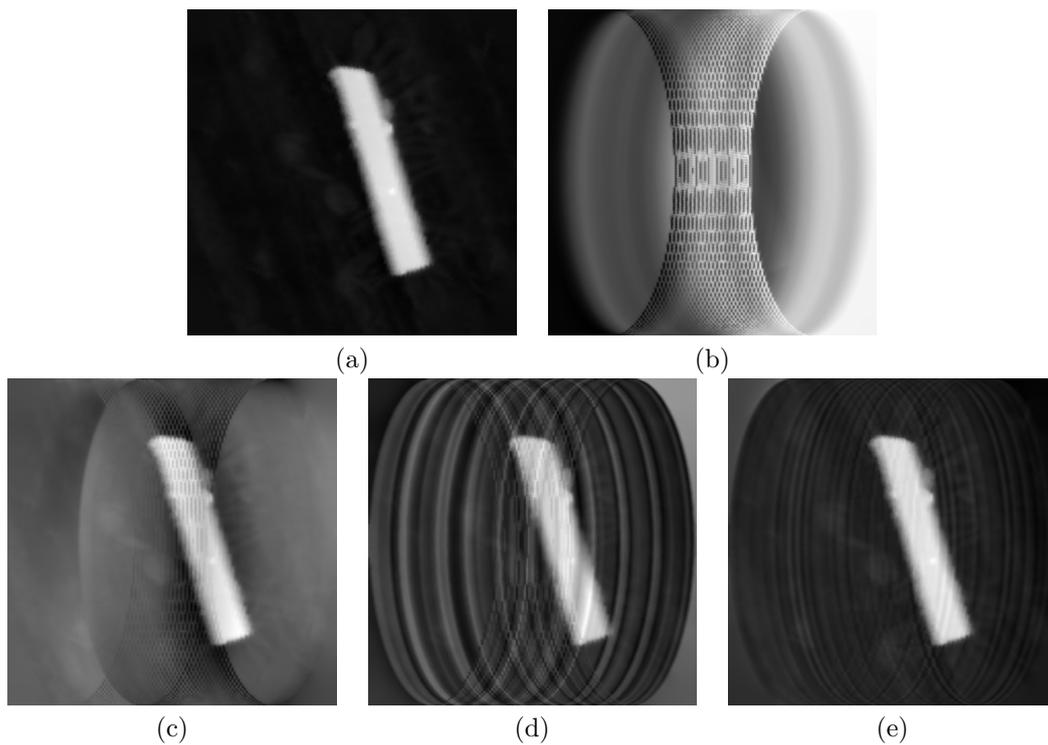


Figure 12: Simulations of drift removal. (a) The original AFM image. (b) The signal generated from the simulated scan. (c) Image generated with poly-flattening drift removal. (d) Image generated with poly-difference-flattening drift removal. (e) Image generated with spline drift removal.

5 Empirical Results and Discussions

A quick review suffices; A sample is scanned using either a cycloid scan or a double Archimedean spiral scan, drift and tilt is removed, and inpainting is done to get the image. For simulation, an image is cycloid scanned, drift removed using different methods, and inpainted using TV inpainting in figure 11 and figure 12.

In figure 12, poly-difference-flattening produces better results than poly-flattening because poly-flattening forces the mean of each arc to be the same value. But in figure 11, poly-flattening produces better results than poly-difference-flattening because of numerical instability. In both cases, drift removal using splines was successful.

6 Summary and Future work

We have shown a process of AFM scanning that will produce images with less scanning time without hardware modifications by using nonraster scan curves, distortion removal, and inpainting. We believe that this process can be used as a standard method in AFM imaging and will produce better images than raster scanning when time is an issue.

There is still much work needed to be done. The proposed process can handle any self-intersecting curves used for scanning but some curves are better at specific requirements than others (See section 2.4 for example). Hence, an exploration of different curves is warranted. Currently, streaks have been difficult to remove because they require the consideration of neighboring points. Because the scanning produces sparse data, a novel method is required. Because of limited time, we did not go deeper than detection but hope to create a streak removal procedure to further improve imaging quality. There is much to explore with the new inpainting methods. Modifications such as including multi-resolution considerations and incorporating different regularizations shows promise in improving the inpainting. Finally, the GUI shows great promise in being used as a companion to a programmable AFM machine.

7 Acknowledgements

We thank Paul Ashby, Dominik Ziegler, Andreas Frank (Lawrence Berkeley National Laboratory) for the sample pictures and assistance in understanding hardware and software related to the AFM. Also, we thank Christoph Brune for his generous help on inpainting.

References

- [1] *Bézier and B-spline techniques*. Springer-Verlag Berlin Heidelberg New York, 2002.
- [2] Atomic force microscopy. http://en.wikipedia.org/wiki/Atomic_force_microscope, 11 2011. accessed 08/04/11.
- [3] Paul D. Ashby Pascal Getreuer Yifei Lou Alex Chen, Andrea L. Bertozzi. Enhancement and recovery in atomic force microscopy images. To be submitted in Proceedings of the February Fourier Talks.
- [4] John Canny. A computational approach to edge detection. *IEEE Transactions in Pattern Analysis and Machine Intelligence*, 1986.
- [5] Ioannis Chasiotis. Atomic force microscopy in solid mechanics. In *Springer Handbook of Experimental Solid Mechanics*.
- [6] Annie I-An Chen, Jeff Huang, and Samuel Lim. Multi-point boundary tracking for atomic force microscopy. Technical report, UCLA Department of Mathematics, 2009.
- [7] R. A. Finkel and J. L. Bentley. Quad trees a data structure for retrieval on composite keys. *Acta Informatica*, 4:1–9, 1974. 10.1007/BF00288933.
- [8] Simon Funk. Netflix update: Try this at home, 12 2006. accessed 08/04/11.
- [9] Tom Goldstein and Stanley Osher. The split Bregman algorithm for L1 regularized problems. *SIAM Journal on Imaging Sciences*, 2(2):323–343, 2009.
- [10] Travis Meyer. Non-local nuclear-norm inpainting. UCLA student Inpainting paper.
- [11] Leonid I. Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Phys. D*, 60:259–268, November 1992.
- [12] J. Tumblin and G. Turk. LCIS: A boundary hierarchy for detail-preserving contrast reduction. In *Siggraph, Computer Graphics Proceedings*, pages 83–90, 1999.
- [13] C. Sbert V. Caselles, J.M. Morel. An axiomatic approach to image interpolation. *IEEE Transactions on Image Processing*, 1998.
- [14] Y.K Yong, Moheimani.S.O.R., and I.R. Petersen. High-speed cycloid-scan atomic force microscopy. *Nanotechnology*, 2010.