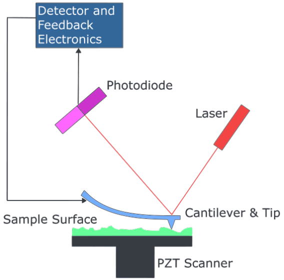
Fast Atomic Force Microscopy Imaging using Self-Intersecting Scans and Inpainting

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

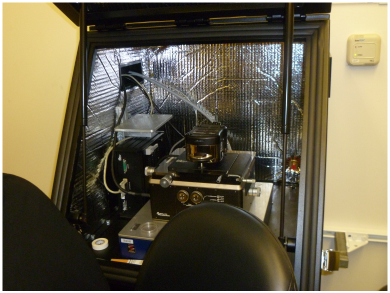
2011 UCLA CAM REU

Outline

Atomic Force Microscopy (AFM)



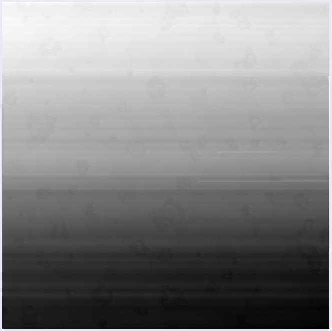
[1]



[1] Source: Wikipedia

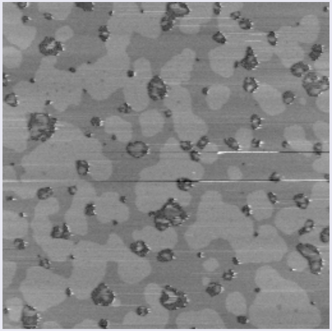
Raster Scan

Raster Scanned Image



(a) The AFM scans top-down with horizontal scan paths. Notice the different mean intensities of each line.

Line-Flattening



(b) Each horizontal scan is line-fitted and the fit is subtracted from the scan line.

Problems with Raster Scan

- Scanning is slow due to resonant frequencies.
- Forces the mean of each line to be the same.
- Distorts image when a dark/light object is on the sides.

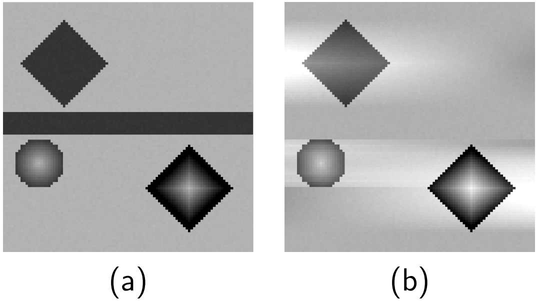
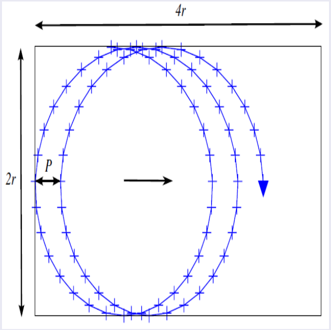


Figure: (a) A simulated image. (b) Line-flattened. Notice the loss of the horizontal bar and the distortion in the foreground and background around objects.

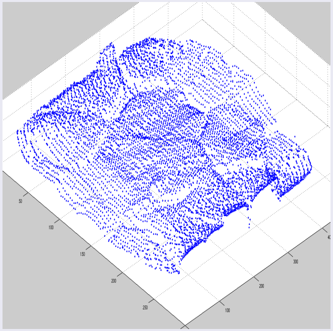
Cycloid Scan

Step 1: Scan[1]



Cycloid Scan Path

Step 2: Collect Data



Data from Cycloid Scan

The Cycloid scan was studied as a feasible way to collect data using a "smooth" path.

Model of Distortion

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

- S: Signal from AFM.
- I: Height of sample - function of bounded variation.
- T: Tilt - approximately a plane in x and y.
- D: Drift - continuous function with small second derivative.
- χ: Streaks - simple function of finite discontinuity.
- η: Gaussian noise.

Poly-Flattening

Subtract polynomial fit from scan arcs in analogue with line-flattening.
Issues:

- Assumes mean is constant across flattening interval.
- Does not enforce continuity between arcs.
- Is multi-valued in intersections.
- Distorts image when there is a high contrast.

Difference-Flattening

We can exploit self-intersections on the scan path.
If $(x(a), y(a)) = (x(b), y(b))$ then
$\Delta_i = S(b_i) - S(a_i) = D(b_i) - D(a_i)$ (ignoring χ and η).
Let ℓ be the number of intersections.

Poly-Difference-Flattening

- Approximate $D(t)$ by an n-degree polynomial $\tilde{D}(t) = \sum_{k=1}^{n} \alpha_k \phi_k(t)$ where $\phi_k(t) = t^k$.
- For simplicity, represent $\alpha = [\alpha_1\, \alpha_2\, \cdots \alpha_n]^T$.
- Using least squares, we minimize $E = \sum_{i=1}^{\ell} (\tilde{D}(b_i) - \tilde{D}(a_i) - \Delta_i)^2$.
- We simplify the process by introducing a general basis for the related function

$$\begin{aligned} \mathfrak{d}(a_i, b_i) &= \tilde{D}(a_i) - \tilde{D}(b_i) = \sum_{k=1}^{n} \alpha_k(\phi_k(t_1) - \phi_k(t_2)) \\ &= \sum_{k=1}^{n} \alpha_k \Phi_k = [\Phi_1(a_i, b_i)\, \Phi_2(a_i, b_i)\, \ldots\, \Phi_n(a_i, b_i)]\alpha. \end{aligned}$$

Poly-Difference-Flattening

- Since $E = \|\Delta - \Phi\alpha\|^2$,
  $\frac{\partial E}{\partial \alpha} = 2\Phi^T\Phi\alpha \ - \ 2\Phi^T\Delta \ = \ 0$ gives

  $$\boxed{\Phi^T\Phi\alpha \ = \ \Phi^T\Delta}$$

  where given the i^{th} intersection, a_i is the first visit to an intersection, b_i is the second visit and $\Phi = [(b_i)^j - (a_i)^j]_{ij}$.

Poly-Difference-Flattening

Benefits:

- Takes into account the intersections.
- Enforces continuity between arcs.
- Preserves shades, i.e. does not have line-flattening distortions.
- Produces better results than Poly-Flattening.

Issues:

- The polynomial requires a high degree for fitting severe thermal drift; this causes numerical instability.
- “Blows up“ at the ends.

Trig-Difference-Flattening

Same idea as poly differences, but using $\{e^{ikx}\}_{0<|k|\leq n}$ for our basis and $\Phi = [e^{\sqrt{-1}jb_i} - e^{\sqrt{-1}jb_i}]_{ij}$.

Benefits:

- Takes into account the intersections.
- Fits the boundedness and oscillation of Tilt and Drift better than Poly-Flattening.

Issues:

- Slower than the previous drift removal methods.
- The trig functions require a high degree for fitting severe thermal drift; this causes numerical instability.

Smoothing Spline over Differences

- For splines, $E = \text{LSQ} + \lambda\text{Penalty}$ where LSQ is the least squares term and Penalty is a penalty term for "waviness" of the spline.
- Arbitrary splines can be constructed with a basis of B-splines:

$$
\begin{aligned}
N_{i,0}(t) &= \begin{cases} 1 & t_i \le t < t_{i+1} \\ 0 & \text{otherwise} \end{cases} \qquad \text{(Base case)} \\
N_{i,j}(t) &= \frac{t - t_i}{t_{i+j} - t_i} N_{i,j-1}(t) + \frac{t_{i+j+1} - t}{t_{i+j+1} - t_{i+1}} N_{i+1,j-1}(t)
\end{aligned}
$$

where j is the degree of the spline polynomial and $1 \le i \le m - 1 - j$ with m being the number of knots.

Smoothing Spline over Differences

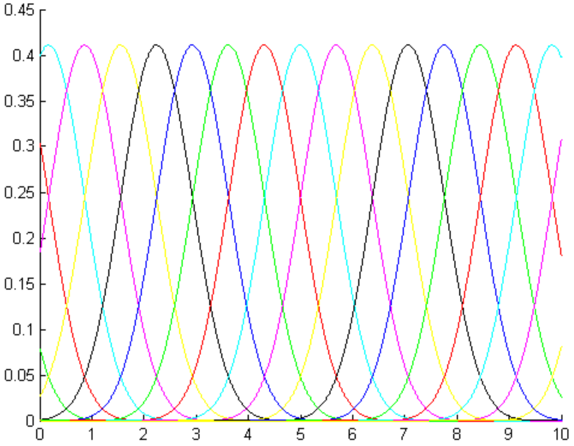


Figure: An example of a spline basis

Smoothing Spline over Differences

- LSQ $= \|\Phi\alpha - \Delta\|^2$ and

$$\begin{aligned}\text{Penalty} &= \int_0^T \|\Phi''\alpha\|^2\, dt \\ &= \alpha^T \left(\int_0^T \Phi''^T \Phi''\, dt\right) \alpha \\ &= \alpha^T M \alpha \quad \text{where T is the ending time.}\end{aligned}$$

- The functional is then

$$E = \|\Phi\alpha - \Delta\|^2 + \lambda\alpha^T M\alpha$$

$$0 = \frac{\partial E}{\partial \alpha} = 2\Phi^T\Phi\alpha + 2\Phi^T\Delta - 2\lambda M\alpha$$

$$\boxed{\Phi^T\Delta = \lambda M\alpha - \Phi^T\Phi\alpha}$$

Smoothing Spline over Differences

Benefits:

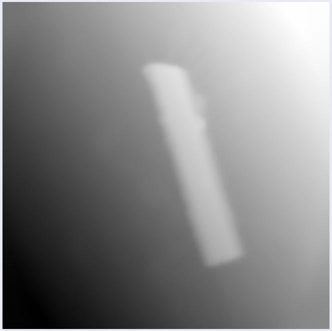
- Takes into account the intersections.
- Fits the boundedness and oscillation of tilt and drift better than Poly-Flattening.
- Does not require a high degree to work well.

Issue:

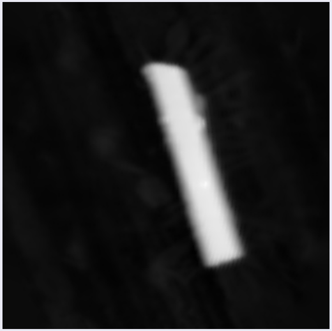
- Requires a recursively defined basis.

Tilt Removal

Image with Tilt



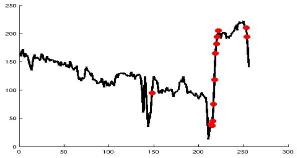
Tilt Removed



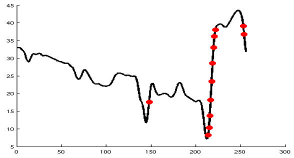
After drift is removed, a plane is fitted to the image and subtracted off.

Streak Detection

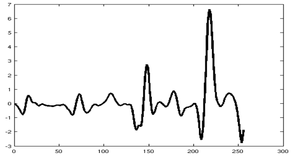
A threshold is used to mark the locations of the smoothed signal with high derivative. This then marks the streaks.



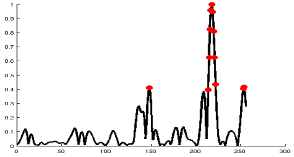
(a) Signal



(b) Smoothed



(c) Derivative of Smoothed



(d) Magnitude of Derivative

Streak Detection

Issues:

- A threshold must be found.
- Conservative thresholds have too many false positives.
- Cannot be used alone to remove streaks.

Double Archimedean Spiral Scan

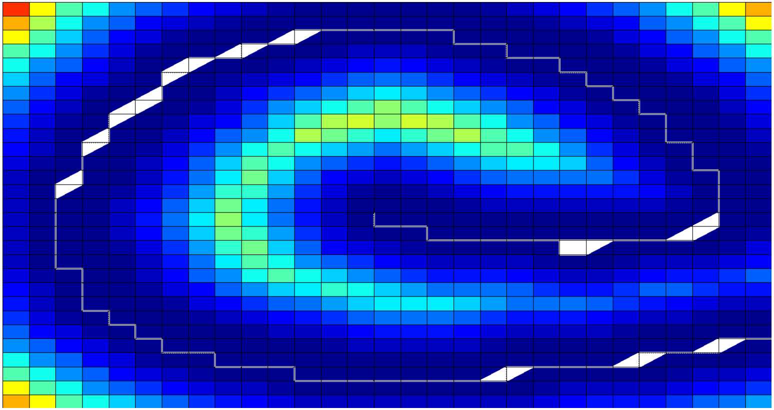
The problem of finding a curve ϕ that covers the most space can be written as an energy minimization:

$$E(\phi) = \int_\Omega \min_t d(z, \phi(t))dz$$

with ϕ having fixed length $L > 0$ and $d(\cdot, \cdot)$ being L^2 distance.

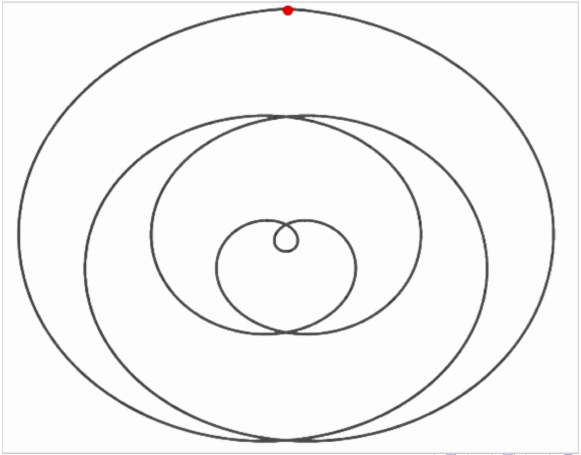
Double Archimedean Spiral Scan

The expression $\min_t d(z, \phi(t))$ can be best visualized as a heat map. Running simulated annealing on Matlab, we get:



Double Archimedean Spiral Scan

The image suggests a path similar to an Archimedean Spiral. To be able to remove drift, intersections are needed, so a Double Archimedean Spiral Scan is used:



Cycloid vs. Double Archimedean Spiral

Benefit of Double Archimedean Spiral over Cycloid:

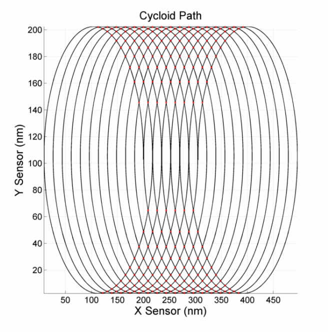
- The Archimedean Spiral fits the Energy functional $E(\phi) = \int_{\Omega} \min_{t} d(z, \phi(t))dz$ better than Cycloid. Hence, it covers more space than Cycloid for the same length of the curve.

Benefit of Cycloid over Double Archimedean Spiral:

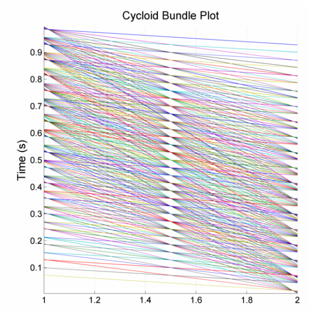
- The Cycloid scan has short-term and long-term self-intersections in regular intervals which facilitates good drift removal.

Cycloid vs. Double Archimedean Spiral

Each connecting line represents a temporal connection between intersections. A connection which starts from the right is when the scan hits the intersection and the left indicates when it is revisited.

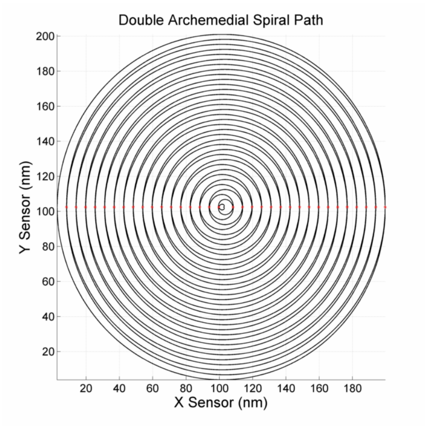


(a) Cycloid Path

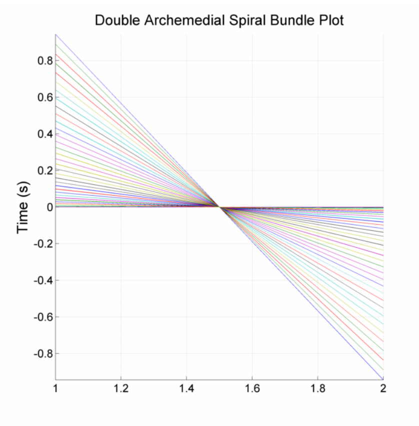


(b) Cycloid Bundle

Cycloid vs. Double Archimedean Spiral



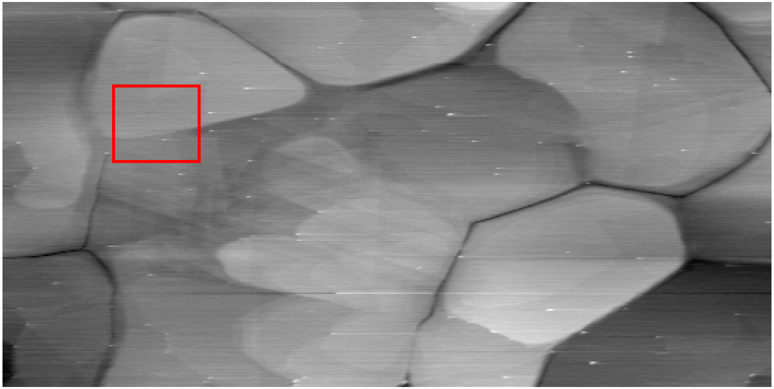
(c) Spiral Path



(d) Spiral Bundle

Penalized Dictionary Inpainting

Given image I, associate with it a neighborhood matrix $B = [\vec{B}_1 \cdots \vec{B}_\ell]^T$, where each $\vec{B}_i$ is a column vector representation of a neighborhood box.



Penalized Dictionary Inpainting

Assume there is a dictionary which succinctly describes these neighborhoods. Then we can write $\tilde{B} = PV$ where P are the coefficients, V is the dictionary, and $\tilde{B}$ approximates B. Then we can minimize the least squares error to simultaneously solve for P and V:

$$E = \|B - PV\|^2$$

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

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

Gradient descent gives,

$$P_{k+1} = P_k + \tau(B - P_k V_k)V_k^T$$

$$V_{k+1} = V_k + \tau P_k^T(B - P_k V_k)$$

Notice that in the case of inpainting the difference $B - PV$ is taken only on known data.

Penalized Dictionary Inpainting

However, this formulation does not enforce a geometric constraint. To regularize it, we include a penalty term. We require adjacent neighborhoods' dictionary basis expression to be similar, i.e. if $B_i = P_i V$ is adjacent to $B_j = P_j V$ then $\|P_i - P_j\|^2$ is small.
This corresponds to a high dimensional derivative and is analogous to Heat equation inpainting.
Combined, these arrive at the evolution:

$$P_{k+1} = P_k + \tau(B - P_k V_k)V_k^T - \lambda M P_k$$
$$V_{k+1} = V_k + \tau P_k^T (B - P_k V_k)$$

Penalized Dictionary Inpainting

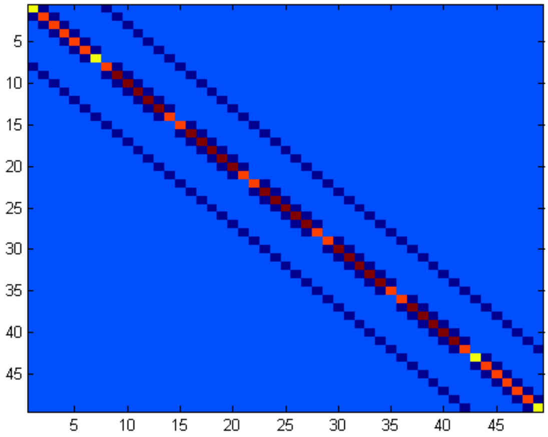
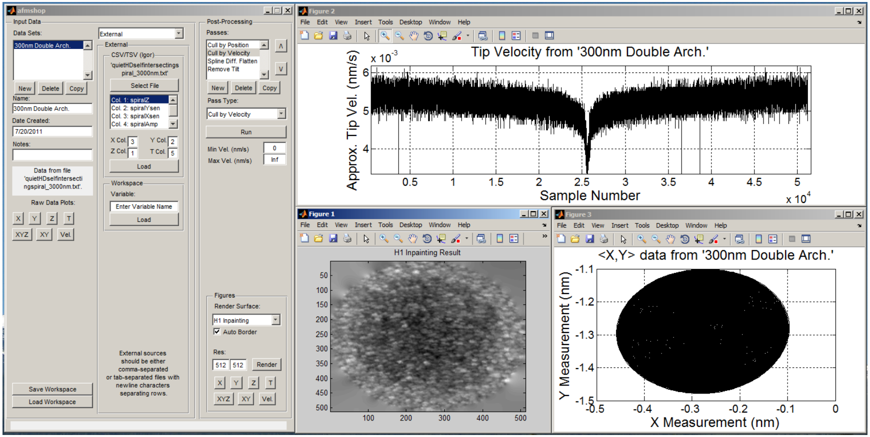


Figure: Banded Matrix M

GUI


afmshop

Input Data

Data Sets: External

300nm Double Arch.

New Delete Copy

Name: 300nm Double Arch.

Date Created: 7/20/2011

Notes:

Data from file 'quietHDselfintersectingspiral_3000nm.txt'.

Raw Data Plots: X Y Z T XYZ XY Vel.

External

CSV/TSV (Igor)

'quietHDselfintersectingspiral_3000nm.txt'

Select File

Col. 1: spiralZ
Col. 2: spiralYsen
Col. 3: spiralXsen
Col. 4: spiralAmp

X Col. 3 Y Col. 2
Z Col. 1 T Col. 5

Load

Workspace

Variable: Enter Variable Name

Load

External sources should be either comma-separated or tab-separated files with newline characters separating rows.

Save Workspace
Load Workspace

Post-Processing

Passes:
Cull by Position
Cull by Velocity
Spline Diff. Flatten
Remove Tilt

New Delete Copy

Pass Type: Cull by Velocity

Run

Min Vel. (nm/s) 0
Max Vel. (nm/s) Inf

Figures

Render Surface: H1 Inpainting

☑ Auto Border

Res: 512 512 Render

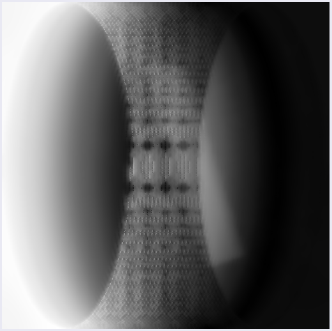
X Y Z T XYZ XY Vel.

Figure: Afmshop

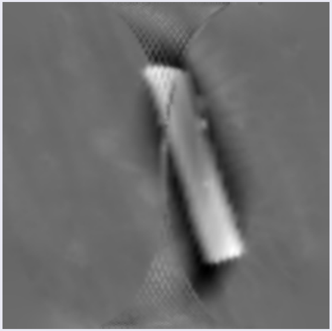
Results: Cycloid Scan

Simulated Scan

TV Inpainting without Drift Removal



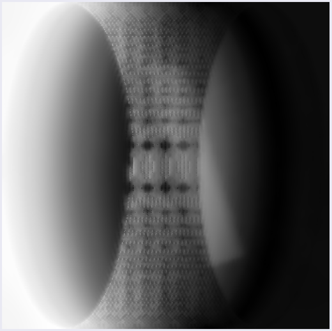
TV Inpainting with Poly-Flattening



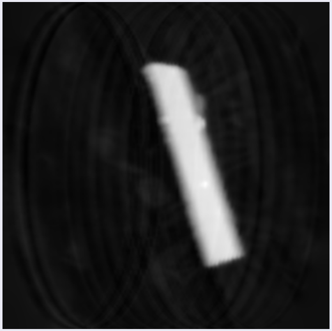
Results: Cycloid Scan

Simulated Scan

TV Inpainting without Drift Removal



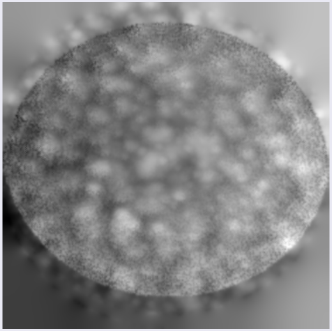
TV Inpainting with Spline Drift Removal



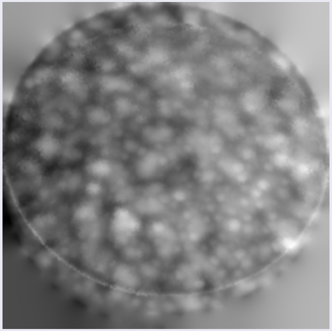
Results: Double Archimedean Spiral Scan

Courtesy of Lawrence Berkeley Laboratory

TV Inpainting without Drift Removal



TV Inpainting with Spline Drift Removal



Results: Penalized Dictionary Inpainting

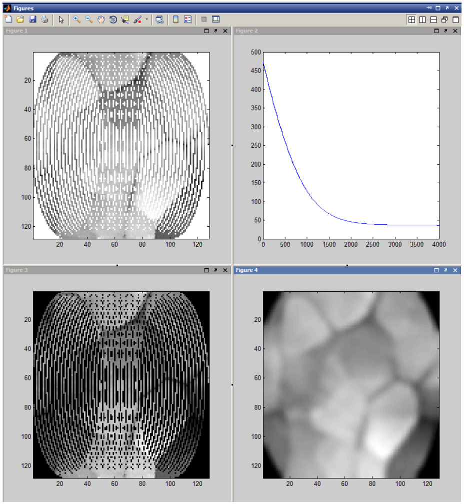
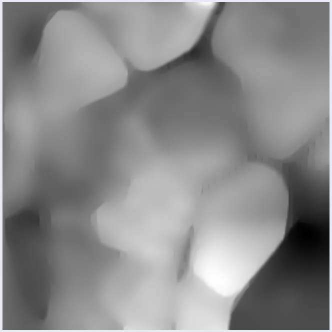


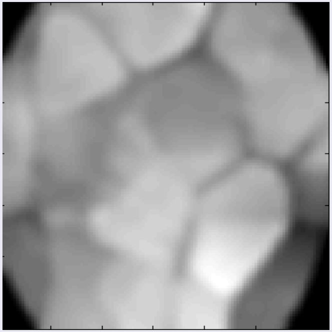
Figure: An example of Penalized Dictionary Inpainting

Results: Penalized Dictionary Inpainting

TV Inpainting



Penalized Dictionary Inpainting



Future Work

- Automate/Improve Streak Detection.
- Investigate other scan paths.
- Incorporate different regularization into penalized dictionary inpainting.
- Extend our inpainting method to include multi-resolution considerations.
- Finalize a GUI that can be used on actual AFM data.

Acknowledgements

Paul Ashby

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.

Reference

Y.K Yong, Moheimani.S.O.R., and I.R. Petersen.
High-speed cycloid-scan atomic force microscopy.
Nanotechnology, 2010.