An algorithm for boundary tracking in AFM

Sean B. Andersson

Abstract— **The standard approach to investigating dynamic phenomena in atomic force microscopy (AFM) is through the use of time-lapse imaging techniques. Because the time to acquire each image is often on the order of minutes, the temporal resolution of this approach is severely limited. There are a variety of examples where the interesting dynamics occur on the boundary of a contiguous structure, such as the growth of lamellar protrusions for cell motility, the growth of crystal structures, or the process of endocytosis by cells. Motivated by this, we present here a high-level feedback control law for boundary tracking in AFM. By steering the tip of the atomic force microscope along the boundary, the acquired image is restricted only to the area of interest. With this approach the total area which needs to be imaged is drastically reduced, leading to a great reduction in the time to acquire each image.**

I. INTRODUCTION

Atomic force microscopy (AFM) images samples by measuring the interaction force between a sharp tip mounted on a cantilever and the sample surface. The technology has been remarkably effective at probing the structure of systems with nanometer-scale features and in recent years there has been increasing interest in using the technology to explore dynamic phenomena in systems at this scale. This has typically been done through the use of time-lapse imaging wherein a sequence of images acquired using the microscope is analyzed using image-processing techniques to detect motion in the image frames. There have been many successful experiments using this technique, involving, among other processes, the assembly of RNA polymerase-DNA complexes [1], protein-protein interactions [2], intercellular trafficking [3], and the motion of influenza molecules in planar bilayers [4].

As with all scanning probe technologies, the measurement in AFM is a local one. As a consequence, the image is built pixel-by-pixel by raster-scanning the tip with respect to the sample. Typical imaging times are on the order of minutes, depending on the size, resolution, and quality desired. The corresponding frame rates are far slower than the rates of many interesting dynamic phenomena, limiting the applicability of the approach. As a result there is a great interest in developing techniques for increasing the scan rate while maintaining image quality. Efforts have included tuning the various components of the microscope [5], application of image tracking ideas to compensate for lateral drift [6], active $Q-$control [7], model-based control [8], and stroboscopic techniques for periodic processes [9].

All of these approaches utilize the standard raster-scan pattern. This pattern can be viewed as an open-loop control scheme since no use is made of the data coming from the microscope. A different approach to improving imaging time is to replace the raster pattern with a high-level closed-loop algorithm which takes advantage of prior knowledge about the sample together with the measured data to steer the tip. In previous work this idea has been used to rapidly image string-like structures [10]. In this paper we focus on imaging the boundary of a sample. These efforts are motivated by several possible applications, such as tracking the growth of lamellar extensions for cellular motility, imaging the growth of crystal boundaries, exploring the fine details of the process of endocytosis by cells, and the investigation of binding and disassociation rates to self-assembled structures such as crystals formed in DNA-based computation (as in the work of Rothemund, Papadakis, and Winfree [11]).

The approach developed here for boundary tracking is inspired by curvature-based control ideas for vehicle formations [12] and is related to work on contour tracking in image processing [13] and contour following using multiple sensor platforms [14]. The algorithm begins with a closed curve, specified by the user, which surrounds a sample of interest. This curve can be determined based on an initial image acquired using the raster-scan pattern. The curve does not need to be precise nor to resemble the sample boundary and therefore the image used to determine it can be of low quality. It can thus be acquired quickly. Once this initialization process is completed, the tip of the AFM is scanned transverse to the curve and the data from the scan are used to update the closed curve via the algorithm described in Sec. II. To track a moving boundary, we assume the rate of change of the boundary of the sample is much slower than the update rate for the surrounding curve. In Sec. II we show that under this assumption our control law causes the surrounding curve to converge to the boundary. In Sec. III the algorithm is applied to a simulated data set and to a a data set collected by imaging with an atomic force microscope crystals formed from self-assembling DNA tiles (provided by Erik Winfree).

II. THE BOUNDARY TRACKING ALGORITHM

Our goal is to steer the tip of the atomic force microscope along the boundary of a sample. The boundary is assumed to form a continuous, closed curve. However, it is not assumed to be smooth nor concave. We surround the sample with a closed, piecewise linear curve referred to as the *envelope curve*. This curve is represented by a finite set of points, $p_j \in \mathbb{R}^2$, $j = 1, 2, \ldots, N$. Associated to each point is a

This work was supported in part by ARO ODDR&E MURI01 Grant No. DAAD19-01-1-0465 (Center for Communicating Networked Control Systems, through Boston University).

S.B. Andersson is with the Aerospace and Mechanical Engineering Department, Boston University, Boston, MA 02215. sanderss@bu.edu

1-4244-0210-7/06/$20.00 ©2006 IEEE

scan line as shown in Figure 1. Note that for clarity most of the scan lines have been left off the image. The direction of each scan line is taken to be the bisector of the angle formed by the points p_{j-1}, p_j, p_{j+1} where we have adopted addition modulo N for all subscripts. Let $\theta_j \in [-\pi, \pi]$, $j = 1, \ldots, N$ denote these directions.

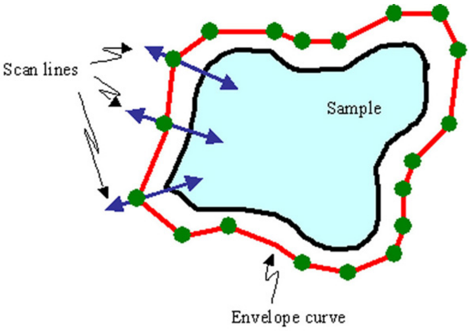


Fig. 1. An example of a sample and an envelope curve. The envelope curve is defined by a set of points. Associated to each point is a scan line. The AFM tip is moved along each scan line consecutively.

At the point p_j, the tip of the AFM is scanned along the direction θ_j towards the sample. The transition from the substrate to the sample can be modeled approximately as a step function. Based on this model, the location of the edge of the sample in the scan line can be estimated from the measured data in a variety of ways depending on the imaging mode used. For example, if the microscope is being operated in tapping mode then the location of the sample can be estimated from the error signal in the control loop maintaining the oscillation amplitude of the tip. As the tip moves from the substrate to the sample there will be a momentary spike in the error signal until the low-level controller can adjust the height of the tip. The location of the spike then corresponds to the location of the edge of the sample.

Let $x_j \in \mathbb{R}$ denote the position of the sample in the scan taken at point p_j. This position varies over time due to the motion of the boundary. This (slow) movement is modeled as a collection of independent simple random walks, i.e.

$$x_j(k+1) = x_j(k) + w_j(k) \tag{1}$$

where k indexes the number of times the scan has completed the envelope loop and $w_j(k)$ is a zero mean stochastic process. In general, one can include a noisy observation of x_j in the system model to account for any noise in the estimate of the location of the edge of the sample in the scan line. However, the capabilities of most atomic force microscopes are such that the position can be measured quite accurately. Therefore we omit modeling the measurement process here. Under the assumption that the rate of change of the boundary of the sample is much slower than the update rate of the controller, the boundary may be viewed as fixed, that is $w_j(k) = 0$.

The control in this scheme is the position of the points p_j. We seek to maintain the envelope curve at a constant distance away from the boundary. To achieve this, we want to drive the position of the boundary in the scan to a desired distance x_d along the scan line. Let q_j denote a unit vector pointing in the scan direction, i.e.

$$q_j(k) = \begin{pmatrix} \cos(\theta_j(k)) \\ \sin(\theta_j(k)) \end{pmatrix} \tag{2}$$

We define the update law

$$p_j(k+1) = p_j(k) - a(x_j(k) - x_d)q_j(k) \tag{3}$$

where $0 < a < 1$ is a gain. According to this law, the point p_j is moved along the scan direction either towards or away from the boundary depending on the error. Since x_j is the position of the sample in the scan line, a shift in p_j causes a corresponding shift in x_j. Using this in (1) yields

$$x_j(k+1) = x_j(k) - a(x_j(k) - x_d). \tag{4}$$

Define the error $e_j(k) = x_j(k) - x_d$. Then

$$\begin{aligned} e_j(k+1) &= x_j(k+1) - x_d \\ &= (1-a)e_j(k). \end{aligned}$$

Since $0 < a < 1$, this update law ensures that locally the envelope curve contracts to a distance x_d from the boundary. Under the assumption that the boundary moves slowly with respect to the update rate of the controller, the envelope curve will locally follow the boundary.

For practical implementation, the length of each scan line is fixed to a constant length L. If the sample is not encountered during the scan then x_k is defined to be ∞. The update law is modified to be

$$p_j(k+1) = \begin{cases} p_j(k) - a(x_j(k) - x_d)q_k, & x_j(k) \neq \infty \\ p_j(k) - aL, & x_j(k) = \infty \end{cases} \tag{5}$$

to handle this. Note that in adopting this modified update law we have assumed the sample lies in the *positive* scan direction, that is along increasing x_j. This in turn implies that the envelope curve is simply connected and that the sample lies in the interior of the curve. Because the update law for p_j does not consider the positions of neighboring points, it is possible that during an update step, two adjacent points will cross, thereby creating a second enclosed region. At the next iteration, the positive scan direction will point *away* from the sample as shown in Figure 2. As a result the loop will grow without bound once it is formed. This can be corrected by re-ordering the points after each update. However, in general there may be a large number of points and thus ordering them can be computationally expensive. The situation can be avoided by choosing the gain a small enough.

The spacing of the points p_j determines the image resolution along the boundary of the sample. As these points are updated, this spacing changes. Thus, after every update of the position the distance to the adjacent points must be checked. If the spacing between two adjacent points is too large, then a new point is added at the midpoint between the two. Similarly if the distance between two adjacent points is too small, then the points are merged by defining a new

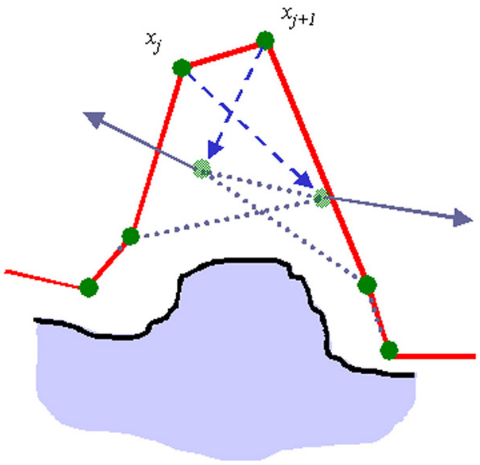


Fig. 2. If two points cross during an update step then a second loop is created. Because the next scan direction is directed away from the sample, the loop will grow without bound. This can be avoided either by re-ordering the points after each update or by keeping the gain a small enough.

point at the midpoint and deleting the two original points. Let Δs denote the desired spatial resolution. To prevent a chattering effect in adding and deleting points, a new point is added when the spacing is greater than $\frac{3}{2}\Delta s$ and points are merged when the spacing is less than $\frac{1}{2}\Delta s$.

To begin the algorithm, an initial envelope curve needs to be defined by the user. This can be achieved by first performing a low resolution, high-speed raster-scan with the microscope over the area of interest. The user can then select the set of points defining the initial curve based on the resulting image.

The boundary-tracking algorithm is summarized as follows.

Algorithm 2.1: Tip steering for boundary tracking

```
0. Initialize. Define a set of points encircling the sample.
1. Scan along the direction defined by $\theta_j$ and measure $x_j$.
3. Update the position of $p_j$ according to (5).
4. If $\|p_j - p_{j\pm1}\| > \frac{3}{2}\Delta s$ then add a point. If $\|p_j - p_{j\pm1}\| < \frac{1}{2}\Delta s$ then merge the points.
5. $j \rightarrow j+1$ and go to 1.
```

A single iteration of the algorithm is defined to be one cycle around the envelope curve.

A. Comments

Using the standard raster scan pattern, the time to image grows with the area, that is, it grows approximately as the *square* of a typical radius of the sample. The boundary, however, grows only *linearly* with this radius. Consider, for example, a simple triangular sample with a base of 900 nm and height of 900 nm. If this sample is imaged using the raster-scan pattern over a 1 μm square with 1 nm resolution at an average tip speed of 5 μm/s (5 Hz scan rate), then each frame will take 200 seconds to image. The perimeter of the triangle is $(1+\sqrt{2})900$ nm long. If the boundary-tracking algorithm is used with 1 nm steps along the perimeter and with each scan line being 30 nm with the average tip speed at 5 μm/s, then the perimeter will be imaged in only 21 seconds. Thus, the tracking algorithm promises a great reduction in the imaging time for the boundary of a sample.

The approach also allows for new experimental techniques. For example, if one is interested not in the detailed structure at the boundary but only on how the shape of the boundary changes over time, then one is not constrained by the need to obtain high quality data about the surface topology of the sample. Combining the boundary tracking algorithm presented in this paper with ideas for rapid detection of sample edges based on the transient dynamics of the cantilever tip [15] will allow the tip to be moved very rapidly along the boundary and will greatly increase the temporal resolution of measurements of the boundary shape.

We note that while the approach described here scans the tip of the microscope along a series of short line segments, the same approach can be used with alternative scan patterns. For example, the tip of the microscope could be moved in an oscillatory pattern transverse to the envelope curve.

III. Simulations

A. Simulated sample

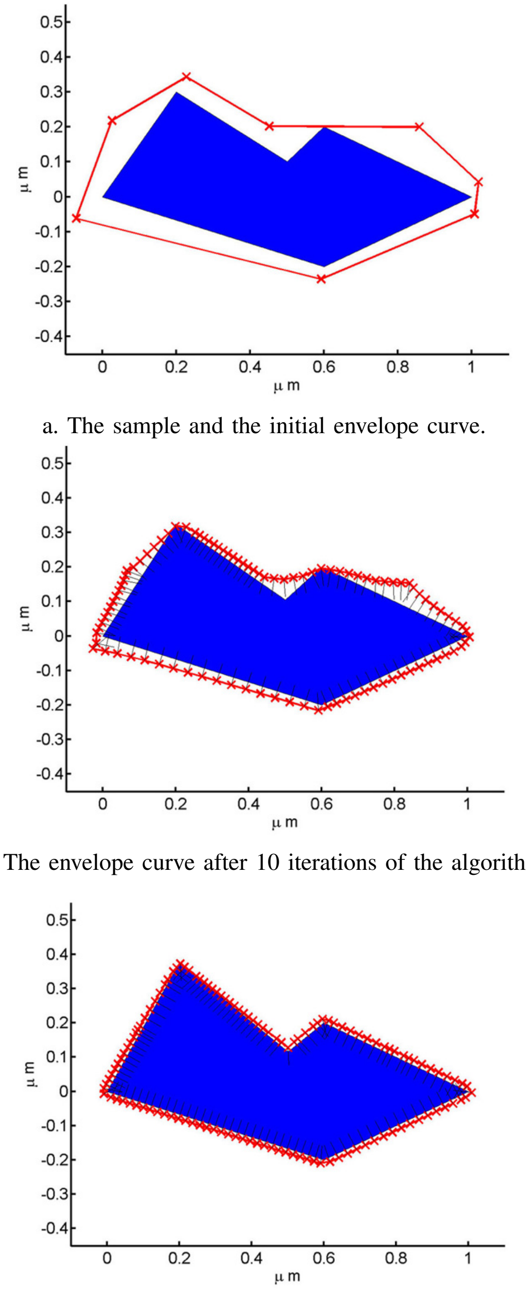
Consider the sample and initial envelope curve shown in Fig. 3a. The desired spacing was set to $\Delta s = 20$ nm, the scan line length to $L = 50$ nm, the desired distance of the envelope curve from the boundary to $x_d = 12$ nm, and the gain to $a = 0.3$. Note that the initial curve did not match the boundary, that the points defining it were not equally spaced, and that the distance between points was much larger than the desired spacing. The envelope curve after 5 and 15 iterations together with the scan lines are shown in Fig. 3b-c. The figure shows that after five iterations the number of points has increased to achieve the desired spacing and the envelope curve has begun to take the shape of the boundary. After 15 iterations the envelope curve has converged to the boundary.

In the next simulation, the boundary of the sample was changed over time by slowly moving two vertices, the one initially at (0.2,0.3) and the one initially at (0.6,-0.2). The first was moved 5 nm in the positive $y-$direction at the end of each iteration and the second by 2 nm. These rates of change were much slower than the update rate of the envelope curve. In Fig. 4 we show the simulation after 10, 20, and 30 iterations. The simulation was initialized with the final boundary curve in Fig. 3c. As can be seen from the figure, the algorithm tracks the boundary throughout its motion.

If the boundary moves faster than the controller can update the envelope curve, then the tracking will fail. To show this, a third simulation was run with the step size of the vertex at (0.2,0.3) set to 50 nm. The results after three and six iterations are shown in Fig. 5.

B. Sample of DNA tiles

Simulations were also run using data from a sample of DNA tiles provided by Erik Winfree [11]. A sample of the tiles was deposited on freshly-cleaved mica and the image

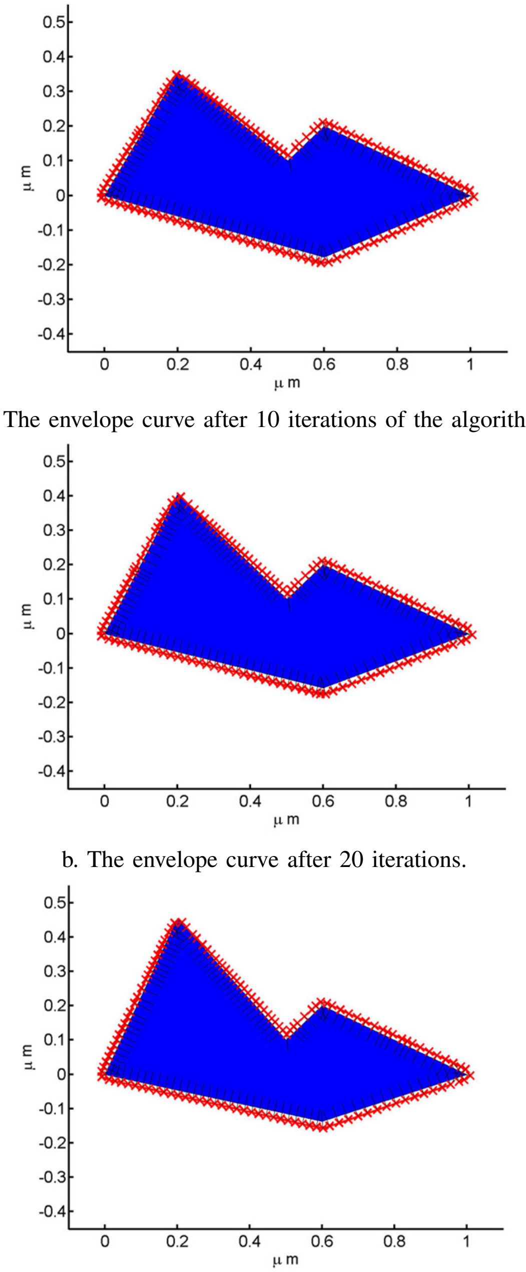


a. The sample and the initial envelope curve.

b. The envelope curve after 10 iterations of the algorithm.

c. The envelope curve after 20 iterations.

Fig. 3. Simulation using a fixed sample. a. The initial envelope curve surrounded the sample but did not have the desired point spacing nor did it approximate the sample boundary. b. The simulation after 10 iterations. c. The simulation after 20 iterations.



a. The envelope curve after 10 iterations of the algorithm.

b. The envelope curve after 20 iterations.

c. The envelope curve after 30 iterations.

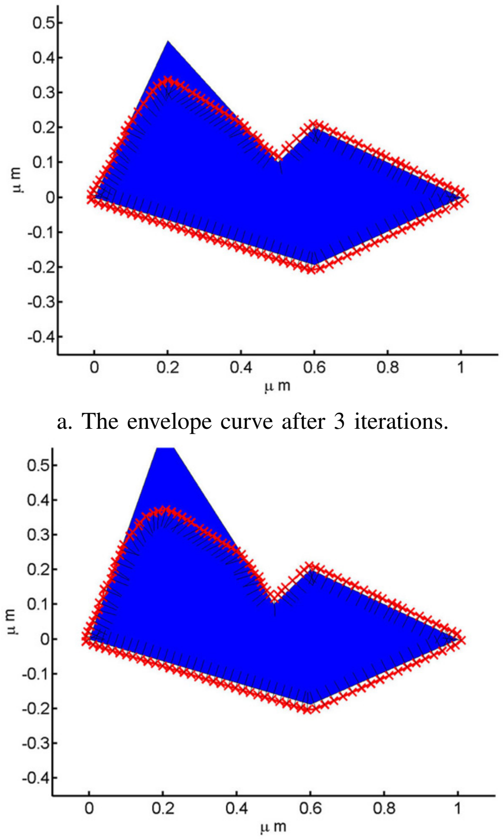
Fig. 4. Simulation using a changing sample. The top left and bottom-most vertex were moved in the positive $y-$direction at a rate much slower than the update rate of the boundary tracking controller. The simulation was begun with the final envelope curve in Fig. 3c.

in Fig. 6 was obtained using an Asylum Research MFP-3D atomic force microscope operated in tapping mode. The tip speed was set to 7 μm/s and the resolution in the plane was 5 nm. The data were imported into Matlab and the algorithm run in simulation on the data. Measurements were simulated by sampling 32 data points from the data set at equally spaced points along a given scan line.

For the first simulation, the control parameters were set to $\Delta s = 5.6$ nm, $L = 11$ nm, $x_d = 2.8$ nm, and $a = 0.5$ and the initial envelope curve shown in Fig. 7a was chosen. Notice that the enclosed DNA crystal has many sharp edges on its boundary. The envelope curve at 10 and 20 iterations is shown in Fig. 7b-c. After 20 iterations the envelope curve has converged to the boundary of the crystal.

In Fig. 8a we show a second initial condition. For this simulation the gain was reduced to $a = 0.3$. The envelope curve at 20 and 40 iterations is shown in Fig.8b-c. As before, the envelope curve converges to the boundary of the enclosed crystal.

Finally we show an example where the algorithm failed.



a. The envelope curve after 3 iterations.

b. The envelope curve after 6 iterations

Fig. 5. Simulation using a rapidly changing sample. The top left vertex was moved at a rate faster than the update rate of the controller. Consequently the tracking controller was unable to follow the boundary.

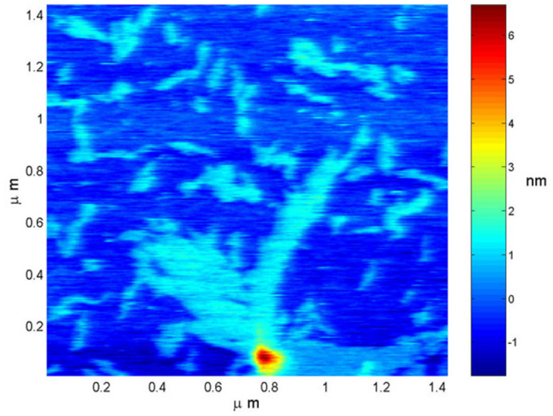
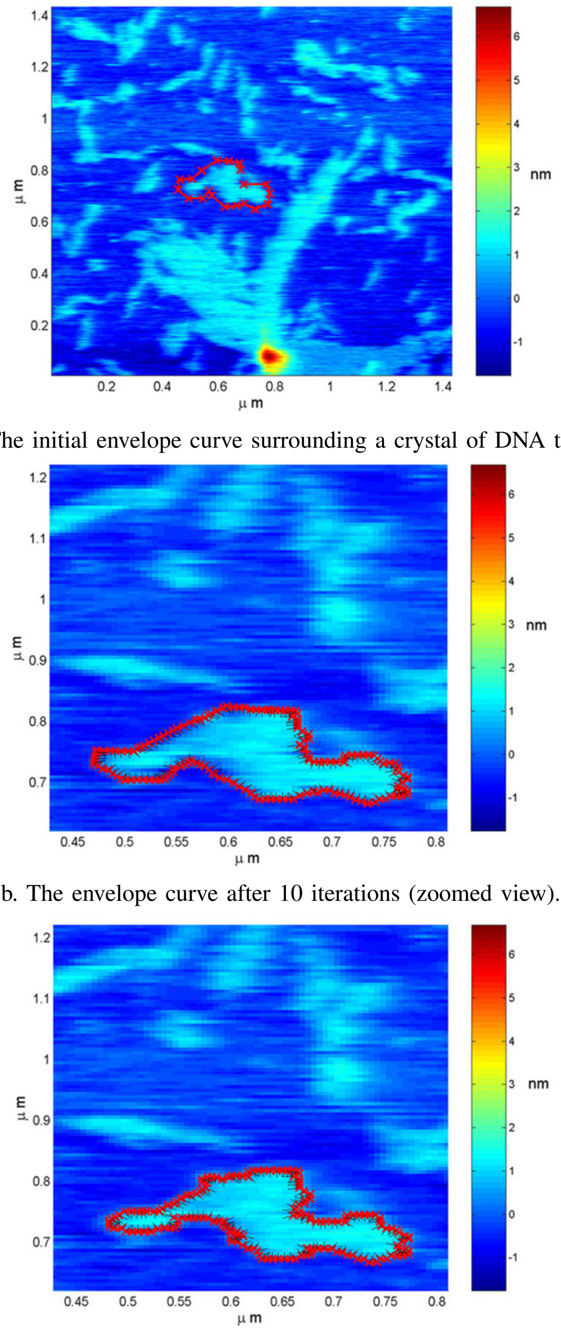


Fig. 6. Tapping-mode AFM image of DNA tiles.



a. The initial envelope curve surrounding a crystal of DNA tiles.

b. The envelope curve after 10 iterations (zoomed view).

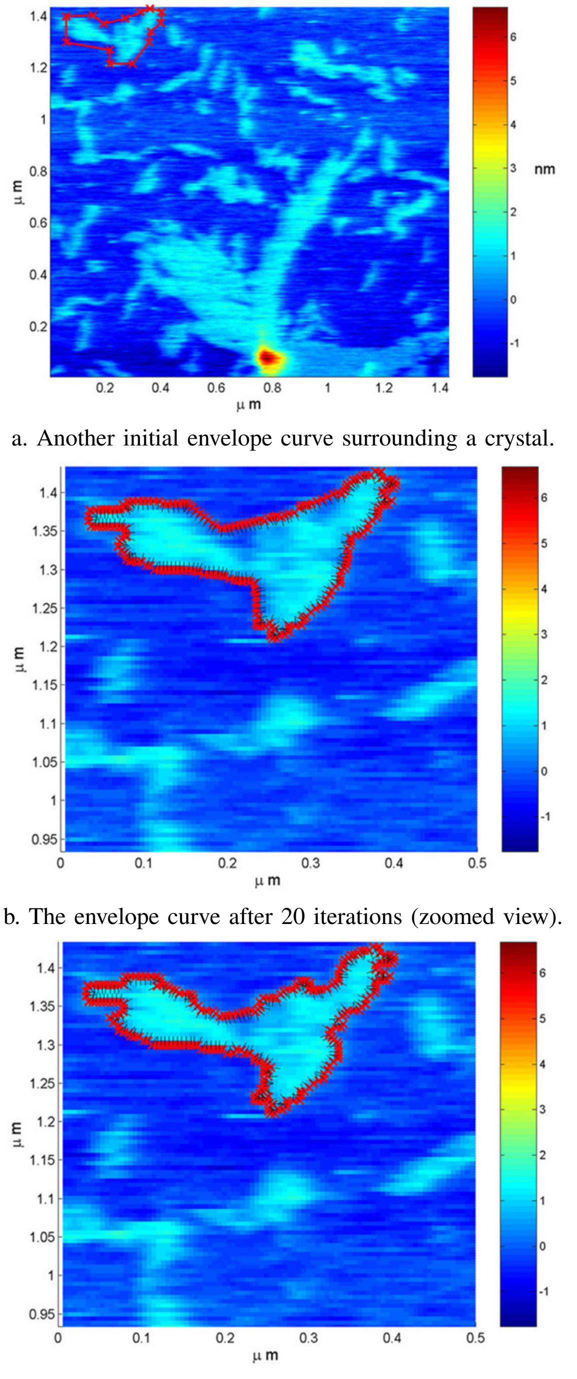
c. The envelope curve after 20 iterations (zoomed view).

Fig. 7. A simulation using measured data of a sample of crystals formed from DNA tiles. a. The initial envelope curve is chosen to surround one of the crystals in the image. b. The envelope curve after 10 iterations. c. The envelope curve after 20 iterations.

The same initial condition as in Fig. 8 was chosen but in this case the gain was set to $a = 0.7$. The result after 14 iterations is shown in Figure 9. Note that two new loops were spawned. As discussed in Sec. II, these loops would grow without bound.

IV. Conclusions

We have presented a high-level feedback control law for performing boundary tracking for rapid imaging in AFM. The scheme moves the tip along an envelope curve surrounding the sample, making short line scans perpendicular to the curve. The measured data is used in real-time to modify the curve so that it converges to the boundary. Boundary tracking is accomplished through a two time-scale assumption, that is, it is assumed that the boundary moves very slowly with respect to the rate of motion of the tip around the boundary. The scheme has been tested in simulation using simulated



a. Another initial envelope curve surrounding a crystal.

b. The envelope curve after 20 iterations (zoomed view).

b. The envelope curve after 40 iterations (zoomed view).

Fig. 8. A simulations using a different initial envelope curve. a. The initial envelope curve surrounds a different crystal. b. The envelope curve after 20 iterations. c. The envelope curve after 40 iterations.

data and data obtained from imaging a sample of crystals of DNA tiles with an atomic force microscope.

V. Acknowledgments

The author gratefully acknowledges E. Winfree and R. Hariadi of Caltech for providing samples of their DNA tiles and J. Park of the Rowland Institute at Harvard for the use of his atomic force microscope.

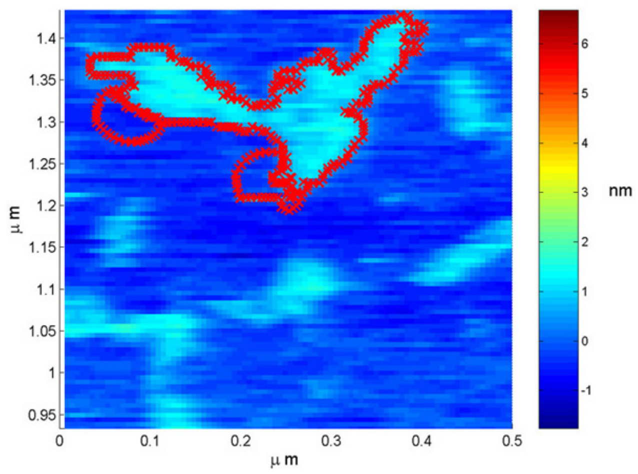


Fig. 9. A simulation with the gain set to $a = 0.7$. With this gain the algorithm points cross each other. As a result new loops are spawned and begin to grow. In this image two such loops have been created.

References

[1] M. Guthold, M. Bezanilla, D. Erie, B. Jenkins, H. Hansma, and C. Bustamante, "Following the assembly of RNA polymerase-DNA complexes in aqueous solutions with the scanning force microscope," *P. Natl. Acad. Sci. USA*, vol. 91, pp. 12 927–12 931, 1994.

[2] M. Viani, L. Pietransanta, J. Thompson, A. Chand, I. Gebeshuber, J. Kindt, M. Richter, H. Hansma, and P. Hansma, "Probing protein-protein interactions in real-time," *Nat. Struct. Biol.*, vol. 7, no. 8, pp. 644–647, 2000.

[3] S. Kumar and J. Hoh, "Probing the machinery of intracellular trafficking with the atomic force microscope," *Traffic*, vol. 2, pp. 746–756, 2001.

[4] T. Hughes, B. Strongin, F. Gao, V. Vijayvergiya, D. Busath, and R. Davis, "AFM visualization of mobile influenza A M2 molecules in planar bilayers," *Biophys. J.*, vol. 87, pp. 311–322, 2004.

[5] T. Ando, N. Kodera, E. T. an dd. Maruyama, K. Saito, and A. Toda, "A high-speed atomic force microscope for studying biological macromolecules," *P. Natl. Acad. Sci. USA*, vol. 98, no. 22, pp. 12 468–12 472, 2001.

[6] S. van Noort, K. van der Werf, B. de Grooth, and J. Greve, "High speed atomic force microscopy of biomolecules by image tracking," *Biophys. J.*, vol. 77, pp. 2295–2303, 1999.

[7] T. Sulcheck, R. Hsieh, J. Adams, G. Yaralioglu, S. Minne, C. Quate, J. Cleveland, A. Atalar, and D. Adderton, "High-speed tapping mode imaging with active q-control for atomic force microscopy," *Appl. Phys. Lett.*, vol. 76, no. 11, pp. 1473–1475, 2000.

[8] G. Schitter, R. Stark, and A. Stemmer, "Fast contact-mode atomic force microscopy on biological specimen by model-based control," *Ultramicroscopy*, vol. 100, pp. 253–257, 2004.

[9] M. Anwar and I. Rousso, "Atomic force microscopy with time resolution of microseconds," *Appl. Phys. Lett.*, vol. 86, no. 1, p. 014101, 2005. [Online]. Available: http://link.aip.org/link/?APL/86/014101/1

[10] S. Andersson and J. Park, "Tip steering for fast imaging in AFM," in *Proc. American Control Conference*, 2005, pp. 2469–2474.

[11] P. Rothemund, N. Papadakis, and E. Winfree, "Algorithmic self-assembly of DNA sierpinski triangles," *PLoS Biol.*, vol. 2, no. 12, pp. 2041–2053, 2004.

[12] E. Justh and P. Krishnaprasad, "Equilibria and steering laws for planar formations," *Syst. Control Lett.*, vol. 52, no. 1, pp. 25–38, 2004.

[13] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *Int. J. Comput. Vision*, vol. 1, pp. 321–331, 1988.

[14] F. Zhang and N. Leonard, "Generating countour plots using multiple sensor platforms," in *Proc. IEEE Swarm Intelligence Symposium*, June 2005, pp. 309–314.

[15] D. Sahoo, A. Sebastian, and M. Salapaka, "Transient-signal-based sample-detectin in atomic force microscopy," *Appl. Phys. Lett.*, vol. 83, no. 26, pp. 5521–5523, 2003.