Massively Parallel Scanning Probe Nanolithography

Daniel J. Arbuckle and Aristides A. G. Requicha
Laboratory for Molecular Robotics
University of Southern California
Los Angeles, CA
{djarb, requicha}@lipari.usc.edu

Abstract—**Direct-writing lithographic processes such as electron-beam lithography or techniques based on Scanning Probe Microscopy (SPM) are sequential, and therefore have a low throughput. This paper discusses parallel approaches to SPM lithography that use multiple tips to achieve high throughputs. Algorithms are presented for planning the motion of an SPM multi-tip array so as to write complex patterns in an efficient, parallel manner. The input is a set of features (polygons) defined in the Caltech Intermediate Format (CIF), which is a *de facto* standard used by most electronic Computer-Aided Design systems. Simulation results are presented to validate the approach.**

Keywords-Nanorobotics, Scanning Probe Microscopy (SPM), Multi-Tip SPM Arrays, Dip Pen Lithography.

I. INTRODUCTION

Scanning Probe Microscopes (SPMs) have been used by various researchers to draw lines and construct nanoscale features on substrate surfaces, using techniques that range from removal of atoms from a passivated silicon surface to the recently introduced Dip Pen Lithography (DPL). DPL is a technique in which writing is performed by touching a surface with an SPM tip that has been inked (*i.e.*, functionalized) with the chemical substance to be deposited [1]. All of these approaches to direct-writing lithography suffer from a major drawback: they are serial and consequently their throughput is low. However, the same SPM processes can be parallelized in a massive fashion by using arrays of SPM tips. The tips are produced by semiconductor fabrication techniques which are amenable to very large-scale integration, and several laboratories around the world are currently developing arrays with hundreds or thousands of tips [2].

This paper discusses algorithms for exploiting the parallelism inherent in SPM tip arrays, and presents results of simulations that show how these algorithms can be used to dramatically increase the speed of SPM lithography. The focus is on DPL, but the algorithms are applicable, with minor modifications, to other forms of SPM-based lithography.

II. BASIC ALGORITHM

The basic strategy used by the algorithms presented here consists of dividing the area to be processed into square cells with a uniform size that equals the inter-tip separation, and then assigning all the writing within a cell to a single tip moving in a raster scanning fashion [3]. Neighboring tips cooperate to produce features that extend beyond the boundaries of a single cell.

We make the following assumptions. The tip arrays are square or rectangular, with a fixed distance between adjacent rows and between adjacent columns. We call this distance the *inter-tip separation*. The tips are assumed to move together in the *x, y* plane of the substrate, but they can be controlled individually in *z.* (Individual control in the *x*, *y* plane would be difficult to implement in hardware, and would cause algorithmic difficulties to ensure collision-free paths.) Manhattan geometry is assumed. The input is a set of polygons with boundaries aligned with the *x* or *y* axes, and defined in CIF (Caltech Intermediate Format), which is used by most Computer-Aided Design (CAD) systems in the electronics area. This choice of input format is a matter of convenience and does not imply that the methods described here are restricted to the electronics domain. Finally, we assume that the tip moving a distance L in a straight line in contact with the surface draws a "thick line" of width D, *i.e*., produces a feature which is the union of a rectangle of width D and length L with two disks of diameter D located at the ends of the rectangle. (More precisely, the feature is the Minkowski sum of the line with length L with the disk of diameter D.) The diameter D determines the minimum line width or constriction in the *x* or *y* directions that can be achieved. D depends on the tip radius, on the speed at which the tip moves, and on the materials used, and must be determined empirically for each experimental setup.

In the basic algorithm each tip scans its cell in a raster motion either in an active (drawing, or lowered) mode, or passive (raised) mode. The tip array moves across the surface in a series of parallel sweeps along the fast scan direction. (Unless otherwise stated, we assume henceforth that the fast scan direction is along the *x* axis.) Each sweep is separated by some constant distance, the *step length*, from its neighbors. The step length must be smaller than D to ensure that the tip traces along neighboring sweeps overlap, but is otherwise unconstrained. (We assume that multiple traces over any region of the desired pattern have no harmful effects.)

The path that the array follows is independent of the features to be written, being instead a regular raster that guarantees total coverage of the area occupied by the features. This results in a division of labor among the tips, with each tip being responsible for writing whatever part of the pattern falls

This work was supported in part by the National Science Foundation under grant IIS-99-87977.

into the area that it traverses. Each tip is lowered as it passes over surface area which should be within the final pattern, and raised otherwise. The width D of the deposited line must be taken into account. (In essence, this amounts to shrinking the pattern by $D/2$.) To ensure that features extending beyond a single cell are drawn as continuous objects we proceed as follows. The scan line being drawn is intersected with the next feature edge. If this intersection is outside the current cell, the line is drawn until the center of the tip reaches the cell's boundary, otherwise writing stops when the tip center is one radius away from the feature's edge. Because the pattern to be written is made out of rectangles, all the required computations are very simple. Fig. 1 depicts the motion pattern of a tip array consisting of four tips being controlled by this algorithm to draw a two-polygon pattern.

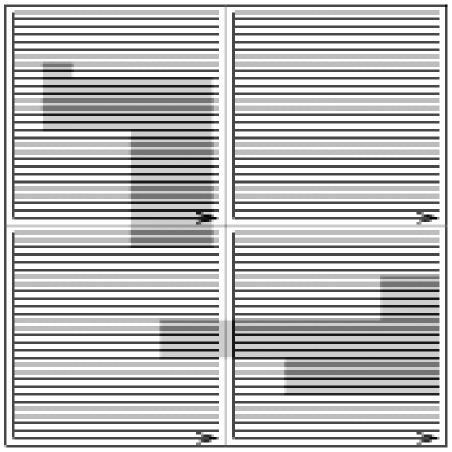
Figure 1. Paths followed by four tips to write a two-rectangle pattern using the basic algorithm. The fast scan direction is along x, and the slow scan direction along y.

III. REFINEMENTS

The basic algorithm is wasteful, because it can spend time on motions that are not needed for the successful completion of the pattern. This can be remedied by shortening scan lines when feasible, and by skipping unnecessary scan lines entirely.

Observe that if any of the tips needs a particular motion in order to draw its section, that motion must be performed; motions can only be skipped if *none* of the tips need them. We compute a *superimposed shape* by moving all the cells to the same origin and computing the union of all the features in this modified single cell. The path that the array should follow during lithography is the same path that a single tip should follow when drawing the superimposed shape.

Fig. 2 shows the motion pattern of tips being controlled by this algorithm, and Fig. 3 the superimposed shape used for planning the tip array path.

The basic algorithm as well as the refined version just described generate artifacts at the edges of the features being drawn. At the edges parallel to the fast scan direction, there is generally some shrinkage, while at the edges oriented along the slow scan axis the edge is scalloped instead of being smooth and straight.

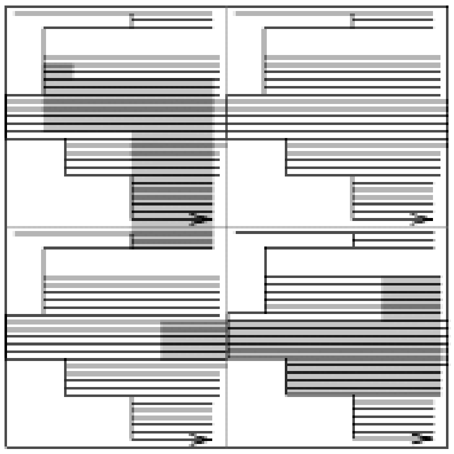
Figure 2. Paths that result from using the refined algorithm on the same pattern and tips as in Figure 1.

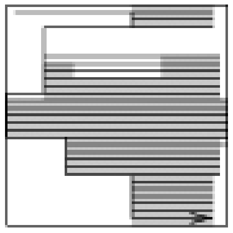
Figure 3. The superimposed shape used for planning the tip array motion with the refined algorithm.

These artifacts can be minimized by aligning the top and bottom of the features with the edges of the thick lines being drawn, and choosing a step size much smaller than the line width D. However, better results can be obtained by performing a "touch-up" pass after the features have been drawn. This pass follows the edges of the features keeping the center of the tip always at a constant $D/2$ distance from the feature edges and within the features. The result is a faithful reproduction of the desired features except for convex vertices, which are rounded with a radius $D/2$. Rounding cannot be helped because it results from the dip pen process itself, which has a characteristic diameter D.

IV. COMPLEXITY OF ALGORITHMS

Much of the complexity of multi-tip algorithms can be factored out into a planning stage which need be run only once for any given pattern to be lithographed. As more copies of the pattern are made, the time cost of the planning stage is amortized, such that as the number of copies approaches infinity the amortized cost of the planning stage approaches zero. What cannot be factored out is the time required to move the SPM tip array.

The time required to generate a series of n patterns via the basic algorithm is shown in equation 1, where Δ_x represents the average velocity of the array when moving the tips along a raster line, Δ_y represents the average velocity when moving tips from one raster line to the next, s_y and s_x represent the separation between tip rows and columns on the array, and l represents the step length. The factor of 2 takes into account the return path ("retrace" in SPM terminology) during which the tip does not write. Since all of these parameters are independent of the complexity of the desired pattern, the time required to generate a set of patterns grows linearly in their number.

$$\sum_n \left(\frac{2s_x}{\Delta_x} \frac{s_y}{l} + \frac{s_y}{\Delta_y} \right) = O(n) \qquad (1)$$

Equation 1 also describes an upper bound on the time required to generate n lithographs via the refined algorithm without touch-up pass. However, the refined algorithm will generally execute in less time than the basic algorithm, hitting that upper bound only in degenerate cases. The behavior of the refined algorithm is described by equation 2. The symbols carry the same meaning as those in equation 1, with the addition that $s_s \leq s_x$ represents the average length of scan lines in the pattern.

$$\sum_n \left(\frac{2s_s}{\Delta_x} \frac{s_y}{l} + \frac{s_y}{\Delta_y} \right) = O(n) \qquad (2)$$

The touch-up pass introduces an object-dependent term in the complexity: the time necessary to trace the boundaries of all the features, which is proportional to the sum of the features' perimeters.

V. SIMULATION RESULTS

We have implemented the multi-tip nanolithography algorithms described in this paper, and used them to control a simulated SPM multi-tip array. Our software reads a CAD drawing in Caltech Interchange Format (CIF) and produces a motion plan for execution on our SPM simulator. The software supports multiple layers in the CAD drawing, simulating each layer as a different DPL "ink."

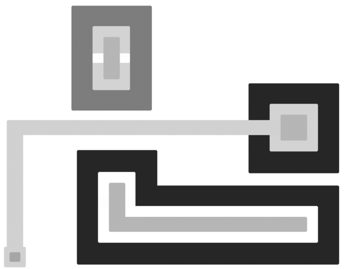

Figure 4. An examplary pattern generated by the multi-tip nanolithography simulator.

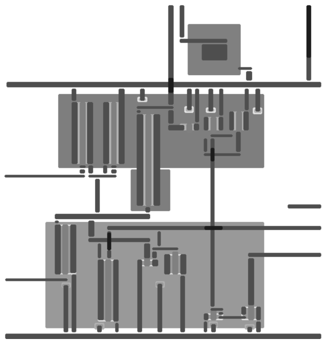

Figure 5. A "neural" cell [4] generated by the multi-tip nanolithography simulator.

The simulator reads a motion plan generated by the planner. Motion plans consist of events, such as "raise tip" or "set velocity", each coupled with a time when each event should occur. The simulator queues up these events and then executes them in order, advancing time and array position as appropriate between events.

Fig. 4 and Fig. 5 present examples of patterns generated by the simulator. The simulation results show that the CAD features are reproduced faithfully by the tip array lithography. The basic algorithm with N tips generates a lithograph in $1/N$ of the time taken by a single tip.

VI. SUMMARY AND CONCLUSIONS

This paper presents algorithms for writing patterns on surfaces using arrays of Scanning Probe Microscope (SPM) tips. The algorithms described here are simple to implement and increase the throughput of SPM lithography by a factor proportional to the number of tips in a tip array. Because tips are made using semiconductor technology, it is expected that future tip arrays will have many thousands or even millions of tips. This will make multi-tip nanolithography practical as a small-batch manufacturing technique, and perhaps even as a mass production process. The algorithms were described in the context of Dip Pen Lithography, which is a promising new approach to nanofabrication, but can be readily modified for other SPM nanolithography processes.

REFERENCES

[1] R. D. Piner, J. Zhu, F. Xu, S. Hong and C. A. Mirkin, "Dip-pen nanolithography", *Science*, Vol. 283, pp. 661-663, 29 January 1999.

[2] P. Vettiger, G. Cross, M. Despont, U. Drechsler, U. Dürig, B. Gotsmann, W. Häberle, M. A. Lantz, H. E. Rothuizen, R. Stutz and G. K. Binnig, "The "millipede" - nanotechnology entering data storage", *IEEE Trans. on Nanotechnology*, Vol. 1, No. 1, pp. 39-55, March 2002.

[3] A. A. G. Requicha, "Massively parallel nanorobotics for lithography and data storage", *Int'l J. Robotics Research* , Vol. 18, No. 3, pp. 344-350, March 1999.

[4] R. Newcomb, ftp://medinfo.eng.umd.edu/pub/vlsi/circuits/ntc20.cif.doc.