A.2.24

Two new methods for simulating photolithography development in 3D

John Helmsen, Elbridge Gerry Puckett, Phillip Colella and Milo Dorr

Center for Computational Sciences and Engineering,
Lawrence Livermore National Laboratory, Livermore, CA 94550
EGP, Department of Mathematics,
University of California, Davis, CA 95616

ABSTRACT

Two methods are presented for simulating the development of photolithographic profiles during the resist dissolution phase. These algorithms are the volume-of-fluid[1] algorithm, and the steady level-set algorithm. These methods are compared with the ray-trace, cell and level-set techniques employed in SAMPLE-3D[2]. The volume-of-fluid algorithm employs an Euclidean Grid with volume fractions. At each time step, the surface is reconstructed by computing an approximation of the tangent plane of the surface in each cell that contains a value between 0 and 1. The geometry constructed in this manner is used to determine flow velocity vectors and the flux across each edge. The material is then advanced by a split advection scheme[3]. The steady Level Set algorithm is an extension of the Iterative Level Set algorithm[2,4]. The steady Level Set algorithm combines Fast Level Set concepts[5] and a technique for finding zero residual solutions to the eikonal function[6]. The etch time for each cell is calculated in a time ordered manner. Use of heap sorting data structures allows the algorithm to execute extremely quickly. A similar technique was submitted by J. Sethian in[7,8]. Comparisons of the methods have been performed and the results are shown.
Keywords: photolithography, volume-of-fluid, level-set, steady level-set, fast level-set, three dimensions, etch, advection, eikonal.

1. INTRODUCTION

Most photolithography dissolution models describe the dissolution process as the advancement of the surface according to a spatially varying etch rate. The etch rate is determined by chemical densities that were assigned during the exposure and post-exposure bake manufacturing steps. Because these materials do not diffuse by a noticeable amount inside the photoresist during the dissolution step, the etch rate varies only as a function of position and not as a function of time[9-14]. This property causes photolithography dissolution to be a useful application for computer simulation from a theoretical perspective, since the mathematics is simple enough that good comparisons can be made between alternate methods of topography simulation to understand the different advantages and disadvantages of various algorithms. In addition, the mathematical simplicity of photolithographic dissolution simulation allows many interesting and innovative algorithms to be constructed and tested before being applied to more complex problems.

Two new techniques are introduced in this manuscript for simulating photolithography dissolution. These techniques are the steady Level Set Method and the volume-of-fluid method[1]. Both techniques can be derived from the Hamilton-Jacobi equation[7]. Each method describes the advancement of a surface, in the direction of its surface normal, according to a specified etch rate that is defined as a function of position and is independent in time. The mathematics of the derivations of each method from the Hamilton-Jacobi equation will be shown, and implementation techniques will be described. The steady Level Set method solves the Hamilton-Jacobi equation as a transformation into the eikonal equation. It should be noted that J. Sethian has published a similar method that takes advantage of the same principles. Both his method and this one were invented independently[8]. The transformation of the Hamilton-Jacobi equation to the eikonal equation generates a function defined over the simulation space, whose contours represent the surface of the resist at different development times. By using a gradient operator that only computes the gradient with local values that are less than the value at the point that the gradient is being evaluated, and a scheduling algorithm, the equation can be solved for exactly with 0 residual. (I.e. the error is only dependent on the accuracy of the grid and the gradient operator. No diffusive smearing occurs as a result of advancement.) Because this method can use past etch times to accurately compute the gradient for any advancing flat surface, this method does not cause facets to occur.

The volume-of-fluid method operates in a manner like a cell method, in that it assigns each cell in the grid a volume fraction

of 0 to 1. The volume fraction represents the amount of volume in the cell that has been etched. Interface reconstruction routines are used at each time step to compute a surface normal for each cell that has a partial volume fraction. The volume fractions are then taken to represent the presence of an etching material, whose advancement under the volume-of-fluids equation represents the etching of the photoresist. As the fluid advances into the simulation region, this is taken to mean that the area that is dissolved by the dissolution solution is advancing into the area occupied by the photoresist.The etched material is then advanced according to the advection equation from fluid mechanics, with an additional source term that is dependent on curvature. The velocity of material advancement is determined by solving the Riemann condition[3] using the interpolated normals and multiplying the result by the etch rate, thereby giving the component of the surface velocity at each cell edge. The advancement is performed by a split advection scheme, thereby simplifying advancement to a number of one-dimensional cases. Because the methods used to reconstruct the interface in volume-of-fluids can exactly recapture any representation of a half plane from its associated volume fractions, this method also has no faceting.

2. HAMILTON-JACOBI EQUATION

2.1. Application of the Hamilton-Jacobi Equation to surface motion

The Hamilton-Jacobi equation is defined for some ϕ as:

$$\frac{\partial \phi}{\partial t} + R\|\nabla \phi\| = 0 \tag{1}$$

where R is the etch rate. For the photolithographic case, R is only a function of position and is not dependent on time. To describe motion of surfaces using this equation, the function ϕ must be a monotonic function with 3 additional conditions. First ϕ must acquire the value 0 on the surface, i.e. the 0 contour of ϕ represents the surface of the photoresist. Second, ϕ must be positively valued and monotonically increase along paths leading into the photoresist. Finally, ϕ in the dissolved region must be negatively valued and monotonically decrease along paths that lead away from the resist surface.

When equation (1) is evolved in time, the 0 contour moves in the direction of positive ϕ at the rate R. This motion may be more accurately demonstrated through the 1-dimensional version of equation (1):

$$\frac{\partial \phi}{\partial t} + R\frac{\partial \phi}{\partial x} = 0 \tag{2}$$

Assume that the equation $\phi(x)$ has the form:

$$\phi(x) = ax + b \tag{3}$$

then the intercept of the x-axis (i.e. the 0 point) is at:

$$x = \frac{-b}{a} \tag{4}$$

Given $R(x) = r$ where r is a constant, equation (2) evaluates to:

$$\frac{\partial \phi}{\partial t} = -r\frac{\partial \phi}{\partial x} \tag{5}$$

or:

$$\frac{\partial \phi}{\partial t} = -ra \tag{6}$$

giving:

$$\phi(x, t) = a(x - rt) + b \tag{7}$$

This demonstrates that the 0 point advances at a speed r for any choice of a. In two and three dimensions, the advancement occurs in the direction of the surface normal at the local etch rate. Therefore, in any small region about the surface, the advancement can be defined as:

$$\frac{\partial \phi}{\partial t} + R(x, y, z) \frac{D\phi}{D\vec{n}} = 0 \tag{8}$$

where $\vec{n}$ is the surface normal, but since the surface normal is the gradient of the equation, the equation can be rewritten as:

$$\frac{\partial \phi}{\partial t} + R(x, y, z) \| \nabla \phi \| = 0 \tag{9}$$

2.2. Boundary conditions

The most natural boundary condition for the Hamilton-Jacobi equation as applied to photolithography and other topography problems is the reflective boundary condition. For this condition, the plane where the boundary occurs is considered to be a "mirror" that reflects the values inside the simulation region. This is a useful condition, since it represents both a region of unetchable material, or the plane of symmetry that represents the boundary between the topographical evolution under consideration and another surface undergoing the same evolution in a symmetric manner.

The boundary condition on the part of the simulation region that represents the initial surface is handled in a different manner. To properly advance the zero contour of the level set, the initial surface is set to $\phi = -t$ where t is the elapsed simulation time. This allows the conditions on the monotonicity of ϕ to be preserved.

3. STEADY LEVEL-SETS

3.1. Mathematical Derivation

A great deal of freedom exists in picking appropriate ϕ to employ in equation (1) for values off the surface of the photoresist. Often, ϕ is chosen to be equivalent to the signed distance function, where ϕ represents the distance along the shortest path from the point to the surface. Another possible choice for ϕ is to set ϕ such that:

$$\frac{\partial \phi}{\partial t} = -1 \tag{10}$$

This causes equation (1) to be rewritten as:

$$R(x, y, z) \| \nabla \phi \| = 1 \tag{11}$$

or, assuming that $R(x, y, z)$ is not equal to 0 at any point (a reasonable assumption for photolithography):

$$\| \nabla \phi \| = \frac{1}{R(x, y, z)} \tag{12}$$

This is the eikonal equation from geometrical optics. Two important results have occurred due to this transformation. First, the time evolution aspect of the equation has been removed. This is relevant, since the removal of time dependence from the equations negates the need to pursue solutions over all spatial elements for many time steps. The non-existence of time dependence suggests that no divided differencing in time is necessary to find a solution, therefore the possibility that a very efficient algorithm to find values of ϕ that satisfy (12) probably exists. Second, the fact that ϕ and t are related by (10) means that a solution to (12) generates the location of the surface at different times as contours of ϕ. (Assume that a solution to (12) exists such that the initial surface location has the boundary condition $\phi = 0$, and the other conditions on ϕ expressed in section 2.1 are satisfied. The resulting ϕ will also satisfy equation (10). Therefore after evolution of (1) from $t = 0$ to $t = c$, the 0 contour of ϕ will be identical to the points that were described by the c contour of the initial ϕ). Therefore any technique that can solve (12) efficiently will also solve (1) for non-zero non-time varying $R(x, y, z)$.

3.2. Implementation

To solve the problem numerically, it is necessary to find solutions to the discretized form of (12):

$$\|\nabla \phi\|_{i,j,k} = \frac{1}{R_{i,j,k}} \tag{13}$$

To find solutions to (13) rapidly, a special discretization of the gradient operator that was described in [5] is employed. This operator was also used to rapidly compute eikonal functions. This operator is:

$$\|\nabla \phi\|_{i,j,k} = \sqrt{(max(D^{plus}{}_x, D^{minus}{}_x))^2 + (max(D^{plus}{}_y, D^{minus}{}_y))^2 + (max(D^{plus}{}_z, D^{minus}{}_z))^2} \tag{14}$$

where:

$$D_x^{plus} = min\left(\frac{\phi_{i+1,j,k} - \phi_{i,j,k}}{\Delta x}, 0\right) \tag{15}$$

$$D_x^{minus} = max\left(\frac{\phi_{i,j,k} - \phi_{i-1,j,k}}{\Delta x}, 0\right) \tag{16}$$

$$D_y^{plus} = min\left(\frac{\phi_{i,j+1,k} - \phi_{i,j,k}}{\Delta y}, 0\right) \tag{17}$$

$$D_y^{minus} = max\left(\frac{\phi_{i,j,k} - \phi_{i,j-1,k}}{\Delta y}, 0\right) \tag{18}$$

$$D_z^{plus} = min\left(\frac{\phi_{i,j,k+1} - \phi_{i,j,k}}{\Delta z}, 0\right) \tag{19}$$

$$D_z^{minus} = max\left(\frac{\phi_{i,j,k} - \phi_{i,j,k-1}}{\Delta z}, 0\right) \tag{20}$$

This operator has two properties that will be taken advantage of. First, the computation of $\|\nabla \phi\|_{i,j,k}$ is only dependent values of ϕ in neighboring cells that are strictly less than $\phi_{i,j,k}$. Second, given advance knowledge of the values of ϕ in neighboring cells that are known to be less than $\phi_{i,j,k}$, the value of $\phi_{i,j,k}$ necessary to satisfy equations (13-20) can be computed by solving:

$$(max(D^{plus}{}_x, D^{minus}{}_x))^2 + (max(D^{plus}{}_y, D^{minus}{}_y))^2 + (max(D^{plus}{}_z, D^{minus}{}_z))^2 - \frac{1}{R_{i,j,k}^2} = 0 \tag{21}$$

since foreknowledge of the neighboring values of ϕ that are less than $\phi_{i,j,k}$ allows the max and min operators to be evaluated. Equation (21) therefore reduces to either a quadratic or linear equation with $\phi_{i,j,k}$ as an unknown. If this equation is a quadratic, then only the largest root will satisfy the max and min operators, therefore the solution is unique. Because solutions to ϕ are found which satisfy equation (13) to nearly the limit imposed by roundoff error, it can be seen that this method is extremely accurate. It is also clear that this method is extremely fast, since the set-up and evaluation of the quadratic equation is a computationally undemanding operation.

In the previous paragraph, it was remarked that equation (13) can be solved using equation (21) provided that the values of all smaller neighboring values ϕ are known for each cell. This can be performed by evaluating ϕ in each cell in ascending order

through the use of a scheduling algorithm. The cells are divided into 3 groups. First, the frozen cells that have computed values of ϕ permanently assigned. The second group of cells is the group of cells with preliminary values of ϕ, these cells are also those cells that are adjacent to a frozen cell. The preliminary values of ϕ are generated through the use of equation (21) and the assumption that Finally there are the untouched cells that have no value of ϕ assigned. None of the untouched cells can be adjacent to a frozen cell. In addition, a heap data structure is maintained with entries for all the preliminary cells sorted by the preliminary values of ϕ.

The algorithm is initialized by setting all of the cells along the top boundary of the simulation region, which represents the top of the photoresist, to the value $\phi = 0$ and placing each of these cells in the frozen list. All cells which neighbor the frozen cells are placed in the preliminary list. The preliminary values of ϕ are generated by assuming that only cells in the frozen list have values of ϕ that are strictly less than the values of in the cells in the preliminary list. The preliminary values are inserted into the heap, and all other cells in the simulation region are placed in the list of untouched cells.

Execution of the algorithm proceeds by removing the lowest valued cell in the preliminary list and placing it in the frozen list, and assigning the preliminary value to be the true ϕ value of that cell. The preliminary cells that neighbor it have their preliminary values recomputed via equation (21) using the recently transferred cell as a cell of lesser value. Neighboring untouched cells are placed in the preliminary list. The algorithm now repeats by removing the next lowest cell from the heap. Termination of the algorithm occurs when preliminary list is empty.

4. VOLUME OF FLUID

4.1. Mathematical Derivation

The volume-of-fluid method is based on a mathematical transformation of equation (1). The function ϕ is replaced by a function f. f is defined as having the value 1 when ϕ is positive and the value 0 when ϕ is negative. This allows equation (1) to be rewritten as:

$$\frac{\partial f}{\partial t} + R \cdot \|\nabla f\| = 0 \tag{25}$$

we now define:

$$\vec{n} = \frac{\nabla f}{\|\nabla f\|} \tag{26}$$

giving:

$$\frac{\partial f}{\partial t} + \vec{n} R \bullet \nabla f = 0 \tag{27}$$

Transformation via vector identities yields:

$$\frac{\partial f}{\partial t} + \nabla \bullet (f R \vec{n}) = f \nabla \bullet (R \vec{n}) \tag{28}$$

Equation (28) is an advection equation of a material f according to a velocity field given by $R\vec{n}$. The value f can be represents the ratio of the volume of the cell occupied by the material to the total volume of the cell. Methods of solving this problem given $R\vec{n}$ have been employed in a wide variety of problems. Therefore, the challenge is to accurately determine $R\vec{n}$. Since R is given as input and is a simple scalar, the real question is solving for the surface normal $\vec{n}$.

4.2. Implementation

Advection methods require velocities on the cell faces. To determine these velocities it is necessary to determine the normal component to the face of the surface normal on these cell faces. This is performed be first approximating the surface normal inside each cell by employing interface reconstruction techniques. Once this has been performed, it is possible to find the value of the normal at each cell face by solving the Riemann problem using Godunov techniques. The advection routine used is an

X-Y-Z, Z-Y-X split advection scheme. In this method, the X component velocities are solved for, and the volume fractions transported only across the cell faces that are perpendicular to the X direction. This is known as an X-sweep. For each sweep, the interface and velocities are completely recalculated. The X-Y-Z, Z-Y-X split advection method performs an X-sweep, followed by a Y-sweep, followed by a Z-sweep on odd time steps, and reverses the order for even time steps.

4.2.1. Interface reconstruction

To determine the normal to the surface resident in each cell, the interface is calculated using Pilliod's fast interface calculation scheme. This is performed by employing a 5x5x5 cube of cells with the cell under consideration at the center. A 5xs5x5 cube of cells is necessary since there are cases where a 3x3x3 cube generates ambiguous results. The center of mass of the fluid is first located in the 5x5x5 cube. This center of mass is then used to rotate the cube so that the center of mass sits as low as possible in the cube. This is the rotation in which the normal of the surface will be found to point into the first octant. Once this is performed, the central column and four neighboring columns have their volumes summed. There are now four possible ways to determine the orientation of the plane, since there are two ways to take divided differences off the central column in each coordinate direction. The divided differences of the column volumes are used to determine the surface normal, and an interface for each normal is generated. The interface that generates the least error is the one selected. This method can regenerate the volumes that represent a flat surface exactly.

4.2.2. The Riemann problem

The Riemann problem is solved by employing Godunov techniques to solve for the jump condition suggested by Hamaguchi. The Hamaguchi technique requires that a characteristic equation be solved that is written in a form requiring conservation of slope. More specifically, if given a discontinuity in the surface normal at a cell edge, the surface normal that persists at the cell edge when the surface is advanced is the true normal at that edge. The velocity is then given as the component of the normal perpendicular to the cell edge times the etch rate in the center of the cell. The surface normal at the edge is computed as follows, assuming a unit etch rate and that the face between two cells is perpendicular to the x direction The normal components on either side of the cell face are labeled $\vec{n}_L$ and $\vec{n}_R$, where the left hand cell is in the direction of lower x:

Table 1: Riemann Problem

Condition	Velocity
$\vec{n}_L > 0$ and $\vec{n}_R > 0$	$\vec{n}_L$
$\vec{n}_L < 0$ and $\vec{n}_R < 0$	$\vec{n}_R$
$\vec{n}_L < 0 < \vec{n}_R$	0
$\vec{n}_L > 0$ and $\vec{n}_R < 0$ and $\vec{n}_L < -\vec{n}_R$	$\vec{n}_R$
$\vec{n}_L > 0$ and $\vec{n}_R < 0$ and $\vec{n}_L > -\vec{n}_R$	$\vec{n}_L$

4.2.3. Flux calculation and volume fraction update

Once the velocities are calculated, the flux of the material across the faces must be calculated. This is performed by finding the volume of the intersection of the interpolated surface with the rectangular prism that extends into the cell a distance $\vec{n}_x \Delta t$. Once all the fluxes of material are computed across the faces (in only one coordinate direction at a time), the updated values are calculated by the formula:

$$f_{i,j,k}^{n+1} = \frac{f_{i,j,k} + \frac{\Delta t}{\Delta x}(flux_i - flux_{i+1})}{1 - \frac{\Delta t}{\Delta x}((\vec{n}R)_{i+1,j,k} - (\vec{n}R)_{i,j,k})} \tag{29}$$

The numerator of the expression solves the advection part of equation (28) while the denominator solves the right hand side.

5.COMPARISON

The steady level-set algorithm has been compared against the Develop, Crater and Advect programs from the SAMPLE-3D suite of tools. The example on which the comparison was performed is from Section 7.3. Table 1 of the thesis of J. Helmsen. This example is a nearly radially symmetric contact cut with standing waves. Each program was compared on two criteria. First, the accuracy of the methods was compared by examining the time at which the surface of the photoresist encountered the substrate, otherwise known as the "break time". This value is compared against an exact value which was computed using analytic methods. The second criteria is the amount of CPU time on a Sun Sparcstation 10 to execute 13 simulated seconds of development.

Table 2: Real Break Time: 12.79

Method	Grid	Break Time	Variation	CPU Sec
Advect	81x81x51	10.13	+0.080	1151
Advect	81x81x101	10.90	+0.040	4080
Advect	81x81x201	11.49	+0.020	16369
Advect Iter.	81x81x51	12.52	+0.040	2065
Advect Iter.	81x81x101	12.54	+0.020	7830
Advect Iter.	81x81x201	12.70	+0.010	30866
Crater	80x80x50	12.045	+0.033	367
Crater	80x80x100	12.542	+0.017	791
Crater	80x80x200	12.549	+0.008	2174
Develop	21x21	12.827	+0.097	202
Develop	31x31	12.89	+0.065	587
Develop	41x41	12.865	+0.049	1344
steady	81x81x51	12.53	N/A	42
steady	81x81x101	12.55	N/A	72
steady	81x81x201	12.70	N/A	134

The variation column represents the size of the time step. Break Time values are given for the beginning of the time step when the break through occurred. It is clear from this example that the steady level-set method is both accurate and fast. It is also known to require a reasonable amount of memory, is easy to program and maintain, and has no difficulty with topological changes or faceting. It is sufficiently fast that a major bottleneck in the steady level-set method is the time that the computer requires to load the rate file from the hard drive and write the computed etch times. Only approximately half of the CPU seconds used above for the steady level-set method were for actual etching, the rest were for bookkeeping functions. J. Sethian reports similar results[8].

The 3D volume-of-fluid method cannot be employed in the comparison, because at the time of this writing, full functionality has not been achieved. Full functionality has been achieved, however, in two dimensions. It has been noted that the present implementation of the three-dimensional volume-of-fluid executes at a rate of approximately 1000 border cells per minute. To do the above example, therefore would require a time of about 10 CPU hours. Because diffusive error accumulates as well, it is expected to be somewhat less accurate than the steady level-set method. Two items, however, must be noted. The slowness of interface calculation in volume-of-fluid is a problem which is expected to be extremely tractable, since the existing interface calculation method is a first past implementation and was not created with efficiency in mind. Second, volume-of-fluid techniques in fluid mechanics typically come into their own when more complicated physics exists, such as simulating the transport of dissolved photoresist through the developer. It is expected that more complicated topography problems than the photoresist problem will be more appropriate for volume-of-fluid, such as problems with simultaneous deposition and etching, where the eikonal transformation that leads to the steady level-set method is wholly inappropriate.

6. CONCLUSIONS

Two new methods have been introduced for simulation of photoresist dissolution. The first, the steady level-set method has shown excellent performance in every category when applied to this problem, and has demonstrated that it can have significant advantages over other photolithography dissolution methods. The steady level-set method, however, is optimized for photolithography dissolution and it is unclear that it's concepts can be applied to other physical models. The second method is the volume-of-fluid technique, which, while being slower when applied to photolithography dissolution, is intended to be a more general method. The interface calculation routine also has significant room for improvement in terms of CPU cycles consumed. Volume-of-fluid is expected to have greater performance when applied to other topography problems, such as simultaneous deposition and etching, and problems involving non-convex behavior, such as RIE etching and ion-milling.

7. REFERENCES

[1] Pilliod, J. and Puckett, E. "Second-Order Volume-of-Fluid Algorithms for Tracking Material Interfaces", Preprint, 1994

[2] Helmsen, J. "A Comparison of Three Dimensional Photolithography Simulators", *Ph.D. Dissertation*, University of California, Berkeley, Dec. 1994.

[3] LeVeque, R. J., *Numerical Methods for Conservation Laws*, p. 202, Birkhauser Verlag, 1990.

[4] Sethian, J. A. "Curvature flow and entropy conditions applied to grid generation." *Journal of Computational Physics*, vol. 115, pp 440-54, 1994.

[5] Adalsteinsson, D. and Sethian, J. "A Fast Level Set Method for Propagating Interfaces", *Journal of Computational Physics*, vol. 118, pp. 269-277, 1995.

[6] Rouy, E. and Tourin, A. "A Viscosity Solutions Approach to Shape-From-Shading", *SIAM Journal of Numerical Analysis*, vol. 29, no. 3, pp. 867-884, June 1992.

[7] Sethian, J. A. "A Review of the Theory, Algorithms, and Applications of Level Set Methods for Propagating Interfaces" in press, *Acta Numerica*, 1995.

[8] Sethian, J. A. "A Fast Marching Level Set Method for Monotonically Advancing Fronts" in press, *Proc. Nat. Acad. Sci.*, 1995.

[9] Dill, F.H., Hornberger, W.P., Hauge, P.S., and Shaw, J.M., "Characterization of Positive Photoresist," *IEEE Transactions on Electron Devices*, vol. ED-22, pp. 456-464, July 1975.

[10] Ferguson, R., Hutchinson, J.M., Spence, C.A., and Neureuther, A.R.,"Modeling and Simulation of a Deep-UV Acid Hardening Resist," *Journal of Vacuum Science and Technology B*, vol. 8, pp. 1423-1427.

[11] Tam, N. N., "Resist Mechanisms and Models in Electron-Beam Lithography," *Ph.D. Dissertation*, University of California at Berkeley, 1991.

[12] Mack, C., "Development of Positive Photoresists," *Journal of Electro-Chemical Society*, vol. 134, no. 1, pp. 148-152, 1987.

[13] Kim, D., Oldham, W., and Neureuther, A., "Development of Positive Photoresist," *IEEE Transactions on Electronic Devices*, vol. 31, pp. 1730-5, Dec. 1984.

[14] Hirai, Y., Sasago, M., Endo, M., Tsuji, K., and Mano, Y., "Process Modeling for Photoresist Development and Design of DLR/sd (Double-Layer Resist by a Single Development) Process," *IEEE Transactions on Computer-Aided Design*, vol. CAD-6, no. 3, pp. 403-9, 1987.

[15] Hamaguchi, S., Dalvie, M., Farouki, R. T., and Sethuraman, S., "A Shock-Tracking Algorithm for Surface Evolution Under

Reactive-Ion Etching," *IBM Research Report*, RC 18283 (80168), Aug., 1992.