

Comparison of Different Approaches for the Simulation of Topography Evolution during Lithography Development

T. Schnattinger, E. Bär

Fraunhofer Institute of Integrated Systems and Device Technology (IISB),

Schottkystrasse 10, 91058 Erlangen, Germany

thomas.schnattinger@iisb.fraunhofer.de

Abstract – Different algorithms for simulating topography evolution are compared in 2D and 3D, using rates for a lithography development process as a benchmarking example. The methods studied are the cell removal, the string, and the fast-marching algorithm. Issues considered are the convergence of the extracted critical dimensions of resist layers with increasing resolution of the simulation grid and the computation time and its dependence on the resolution. Furthermore, it is shown that a slicewise 2D simulation of topography evolution for a 3D structure is not capable of correctly representing the evolving 3D shape of the resist.

I. INTRODUCTION

Modeling of topography evolution is an important part of simulation flows used in semiconductor technology for process characterization and optimization. This particularly holds for the 3D case where at present still a number of problems exist, e.g. with respect to generality, stability and performance. For the simulation of feature evolution, e.g. during lithography, etching, or deposition steps, different algorithms are currently in use. In this paper we compare the cell removal (CR) algorithm, the fast marching (FM) algorithm, and the string algorithm, using the lithography development process as an example. Issues addressed are the convergence with increasing resolution, the performance, and the applicability of 2D evolution to 3D problems.

The resist development process starts from results obtained from the internal lithography simulator of IISB [1] for a process with an illumination wavelength of 248 nm and a square mask pattern, including modeling of optics as well as of resist behavior. The information passed to the 2D and 3D surface evolution modules is the inhibitor concentration discretized on an equidistant 3D tensor grid for the entire resist. Based on these concentrations, the local development rates are determined using the Enhanced Mack model [2].

II. THE DIFFERENT TOPOGRAPHY EVOLUTION ALGORITHMS

Cell removal (CR) algorithm

The basic idea of this method is to represent the geometry as an array of cells and to assign to each cell a counter

value between 0 and 1. Cells that have been already passed completely by the front are assigned a value of 0, cells not yet touched by the front a value of 1. To cells in which the front resides, a value between 0 and 1 is assigned, depending on the position of the front which means how much of the cell has been removed. The counters are updated at each time step according to the local development rate [3].

Fast marching (FM) algorithm [4]

It computes the time value t at which the resist front passes a grid point by solving a partial differential equation where the time t is computed based on the time values and development rates of its neighboring grid points. It is made computationally efficient by (iteratively) computing the time values only for points in a small region around the current front.

String algorithm

The geometry is represented by polygonal segments (2D) or triangular facets (3D). For the lithography development process studied here, nodewise rates are obtained by determining the local inhibitor concentration from the discretized data provided, and by employing the development model to get the local rates. The shift direction of the node is calculated based on averaging the normal of the segments adjacent to the node. In order to allow sufficient accuracy and stability during surface update, short segments (2D)/thin triangles (3D) are deleted before they collapse, and long segments/large triangles are refined to provide sufficient topography resolution [5].

III. COMPARISON OF 2D ALGORITHMS

The criteria for using either the CR, FM, or string algorithm are the required computing time and the accuracy. Both factors depend on the spatial resolution of the simulation grid. As reference discretization (refinement factor = 1) we define a number of cells in x, y, and z direction of 65, 65, and 50, respectively. The refinement factor gives the resolution relative to this reference (the larger this factor, the smaller the cell size). For the 2D investigations, we used one cell layer corresponding to a center cut through the structure. The accuracy is evaluated in terms of critical dimension (CD) and sidewall angle.

As shown in Figures 1 and 2, all methods have reasonable convergence behavior with respect to bottom CD (differences are smaller than 0.0025 μm) and sidewall angle

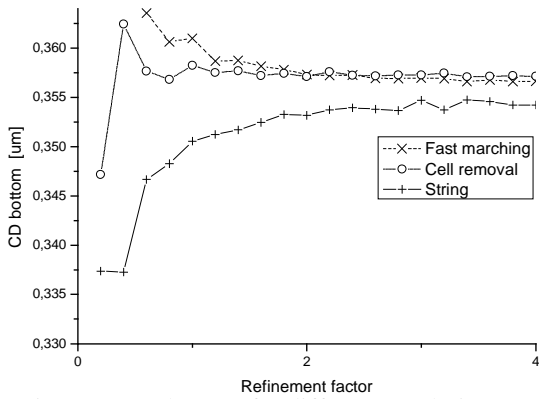


Figure 1: CD bottom for different resolutions.

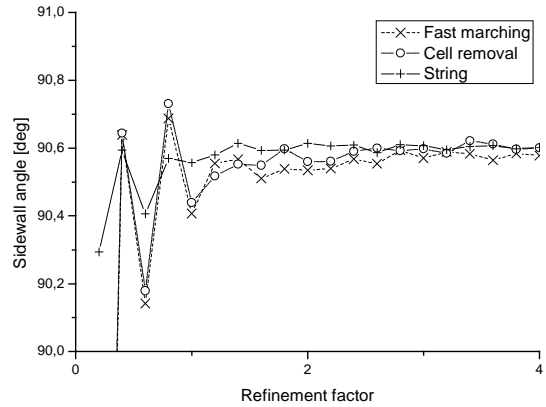


Figure 2: Average sidewall angle for different resolutions.

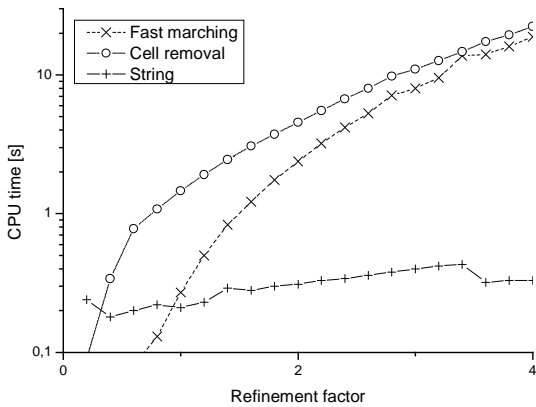


Figure 3: 2D simulation times for different resolutions.

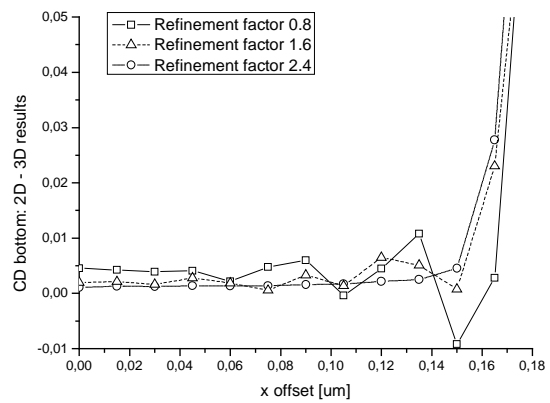


Figure 4: Comparison between 2D and 3D FM development simulation with varying offset from center.

(differences are smaller than 0.02°) for the highest resolution studied. For obtaining stable results (for both CD and sidewall angle), the required resolution (refinement factor 1.5 for CR and FM, 2 for the string) and the corresponding development simulation times (Figure 3) suggest that the string method is most suitable. However, if the additional simulation time for the other lithography steps (see Table 1, columns 1 and 2) is taken into account, the string falls behind CR and FM.

IV. THE NEED FOR FULL 3D DEVELOPMENT SIMULATION

When comparing 2D and 3D FM development simulations, there is very good agreement for x- or y-axis parallel 2D cuts through the center of the structure. For cuts made with an offset from the center, a difference between 2D and 3D simulation is observed (see Figure 4). In general, the 2D results are not accurate if the normal vector of the developing front is not parallel to the 2D cut direction, a phenomenon that can also be explained theoretically and that is

Resolution in x, y, z direction, refinement factor	Lithography simulation without development [s]	String algorithm [s]	Fast marching algorithm [s]	Cell removal algorithm [s]
32, 32, 25 0.5	9	320	1	0.2
65, 65, 50 1.0	20	224	7	1
130, 130, 100 2.0	138	214	56	10
260, 260, 200 4.0	2220	219	480	80

Table 1: 3D simulation times for the development process for different resolutions. To put the results into the context of the entire simulation run, also the simulation times required for the preceding lithography steps are shown.

independent of the development algorithm used. Therefore, 3D development simulations are required to capture the shape of the developed resist for general layouts and cannot be replaced by 3D rate data coupled to slicewise 2D development.

V. COMPARISON OF 3D MODULES

In Figures 5, 6, and 7, results for low and high resolution are shown for the string, the FM, and the CR algorithm, respectively. As one will expect, with increasing resolution the surface becomes smoother in all cases. The distortion at lower resolution is more randomized for the string than for the FM and CR algorithm, which is due to the unstructured surface mesh used by the string algorithm. The roughness of the surface for the CR algorithm (Figure 7) is due to the algorithm used for extracting the triangulation (as needed for the visualization) from the cells.

As can be seen in Figure 8, the CD values converge with increasing resolution, the remaining difference at high resolution is smaller than half of the cell size. In Table 1, the simulation times are compared. The dependence of the simulation times on the resolution is similar to the 2D case. The slightly increased time for the string at low resolution is due to some additional (unwanted) surface refinement caused by the surface noise.

Compared to the 2D case, in 3D the level of the string algorithm simulation times is higher relative to the times for the FM and the CR algorithms. For the 2D simulations, the almost constant time for the string algorithm equals the times for the FM and the CR algorithm for refinement factors of about or below 0.5, whereas for the 3D case the level of the string simulation times corresponds to a refinement factor between 2 and 4 (for the FM algorithm) and above 4 (for the CR algorithm). In other words, the additional dimension is computationally more expensive for the string than for the FM and the CR algorithm. This

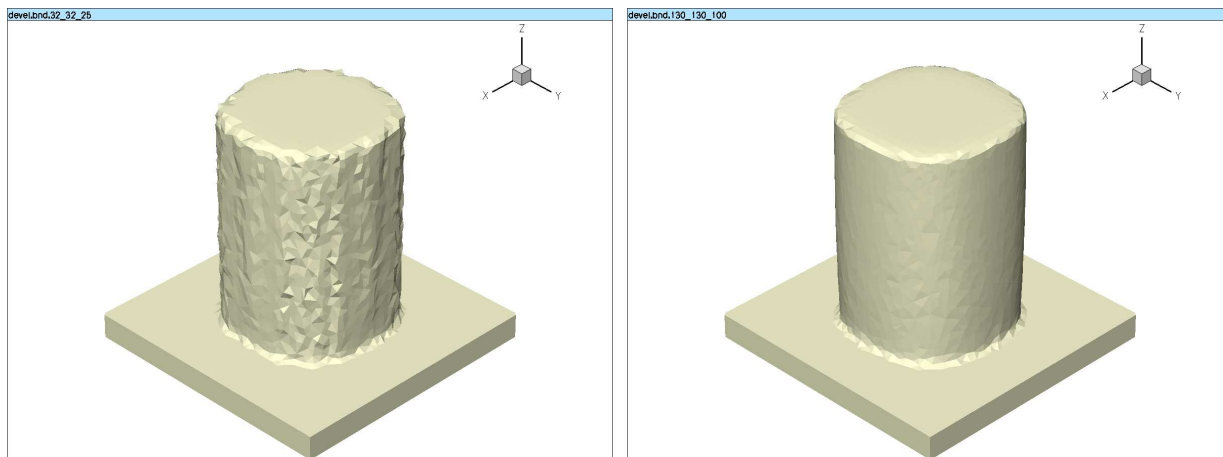


Figure 5: 3D resist topography obtained with the string algorithm for a resolution of (32,32,25) cells (left), and for (130,130,100) cells (right).

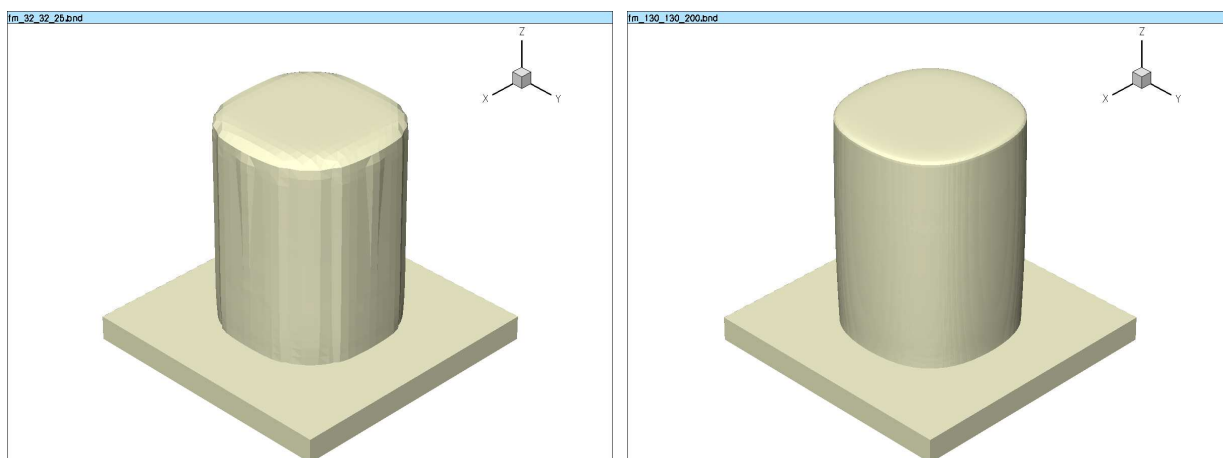


Figure 6: 3D resist topography obtained with fast marching algorithm for a resolution of (32,32,25) cells (left), and for (130,130,100) cells (right).

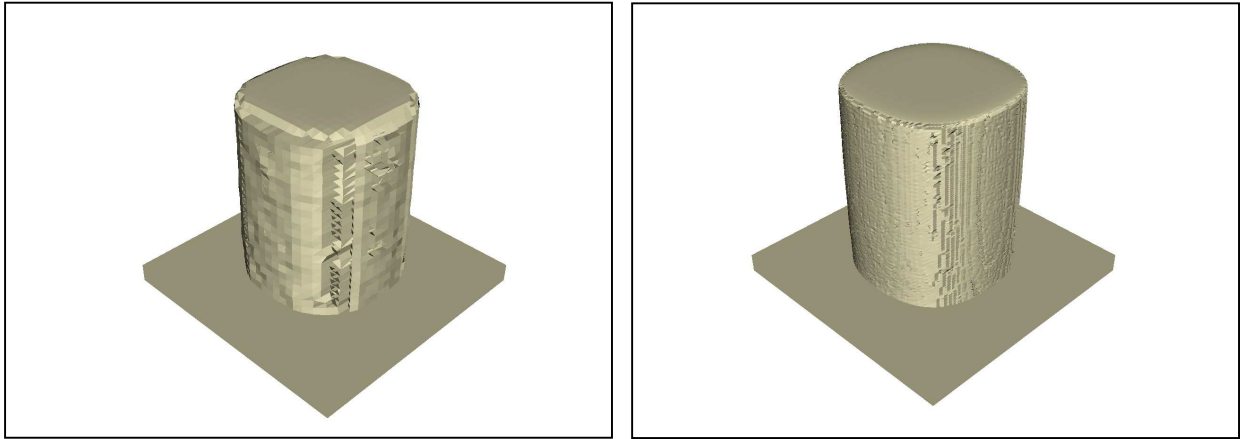


Figure 7: 3D resist topography obtained with cell removal algorithm for a resolution of (32,32,25) cells (left), and for (130,130,100) cells (right).

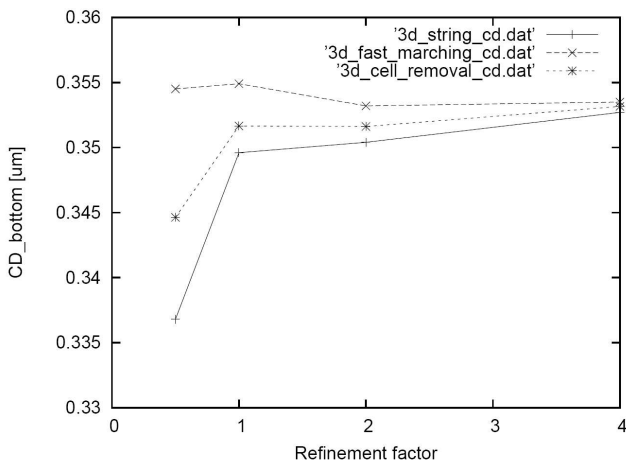


Figure 8: Comparison of the bottom CD determined by the 3D string, fast marching, and cell removal algorithm.

observation can be explained by the fact that the extension of the FM and the CR algorithm by one dimension is more straightforward than it is for the string algorithm. In particular, this is due to the triangular surface grid for the string algorithm in 3D which needs to be refined and coarsened many times during the simulation run, whereas the underlying cell discretization for the FM and CR algorithm remains constant throughout the entire simulation.

In consequence, for the implementations of the different algorithms and for the benchmark case studied in this work, in 3D the CR and the FM algorithm outperform the string algorithm. However, an additional aspect to be considered could be the usage of the simulation result for the simulation of subsequent process steps such as etching, or the integration with data formats available in TCAD environments. In case a conversion from the FM or CR structures (which are both based on a cellular representation) to a polygonal format (as it is used for the string algorithm) is needed, problems may arise due to this conversion [6].

VI. CONCLUSIONS

To gain confidence in simulation results, comparison of different independent simulation approaches is mandatory. This holds for the modeling part as well as (when regarding topography simulation) for the routines used to treat evolving geometries. We have shown the consistence of three different surface evolution methods for the application to lithography development. When passing these results to etching simulation, the number of discretization elements can be an additional important criterion, and reduction techniques will be necessary to allow acceptable computation times for the etching simulation.

ACKNOWLEDGEMENTS

Part of this work has been carried out within the project ABBILD, funded by the Bundesministerium für Bildung und Forschung.

REFERENCES

- [1] A. Erdmann, Modeling and Simulation, in: Handbook of Mask Making Technology, Ed. S. Rizvi, Marcel Dekker, 2005.
- [2] S. Robertson, C. Mack, M. Maslow, Proc. SPIE 4404 (2001) 111.
- [3] W. Henke, M. Weiss, in: Proc. KTI Microlithography Seminar, INTERFACE'91, San Jose, 1991, p. 257.
- [4] J. A. Sethian, Level Set Methods and Fast Marching Methods, Cambridge University Press, 1999.
- [5] E. Bär, J. Lorenz, IEEE Trans. Semicond. Manufact. **9** (1996) 67.
- [6] E. Bär, J. Lorenz, IEICE Trans. Electronics E83-C (2000) 1338.