Three-Dimensional Simulation of Focused Ion Beam Processing Using the Level Set Method

Otmar Ertl, Lado Filipović, and Siegfried Selberherr
Institute for Microlectronics, TU Wien, Gußhausstraße 27–29/E360, A-1040 Wien, Austria
Phone: +43 1 58801 36034, Fax: +43 1 58801 36099, Email: {ertl|filipovic|selberherr}@iue.tuwien.ac.at

***Abstract*— Three-dimensional simulations of focused ion beam milling, which use the level set method for surface evolution, are presented for the first time. This approach allows the inherent description of topological changes. The surface rates are calculated using Monte Carlo ray tracing in order to incorporate shadowing as well as redeposition. Parallelization is used to reduce the computation time.**

I. Introduction

Focused ion beams (FIBs) can be used for structuring, either by local removal or deposition of a material, for sample sectioning, mask repair, or fabrication of microelectromechanical systems [1]–[3]. Simulations are commonly performed in order to predict the shape of the final structure. Most simulators for FIB processing, which are capable to incorporate secondary effects, such as redeposition of sputtered particles, are restricted to two dimensions and can therefore only be applied to problems invariant to rotation or translation [4]–[6]. The only three-dimensional approach we are aware of, uses a segment-based method for surface tracking [7], which is not able to describe any topological changes of the surface. An improved approach is based on the level set (LS) method [8], which has already been successfully applied to two-dimensional FIB simulations [5]. This paper demonstrates the application of the LS method in three dimensions by means of a typical FIB process model.

II. Model

The ion beam is usually approximated to be collimated and its profile is considered to be Gaussian [3], [5], [7]. Hence, the ion flux density F obeys

$$F(r) = \frac{I}{e} \cdot \frac{1}{2\pi\sigma^2} \cdot \exp(-r^2/2\sigma^2), \quad \sigma = d/\sqrt{8\ln 2}, \tag{1}$$

where I is the beam current, $e = 1.602 \times 10^{-19}\,\mathrm{C}$ is the elementary charge, r denotes the distance to the center line of the beam, and d is the beam diameter, commonly defined as the full width at half maximum. For all simulations, presented later, d is chosen to be $50\,\mathrm{nm}$ and the beam current I is set to $50\,\mathrm{pA}$.

The removal rate on the surface is proportional to the sputter yield which describes the average number of removed particles per incident ion and depends strongly on the incident angle. The characteristic behavior of the sputter yield as a function of the incident angle θ can be approximated by Yamamura's semi-empirical formula [9]

$$Y(\theta) = C_1 \cdot (\cos\theta)^{-C_2} \cdot \exp(C_3 \cdot (1 - (\cos\theta)^{-1})), \tag{2}$$

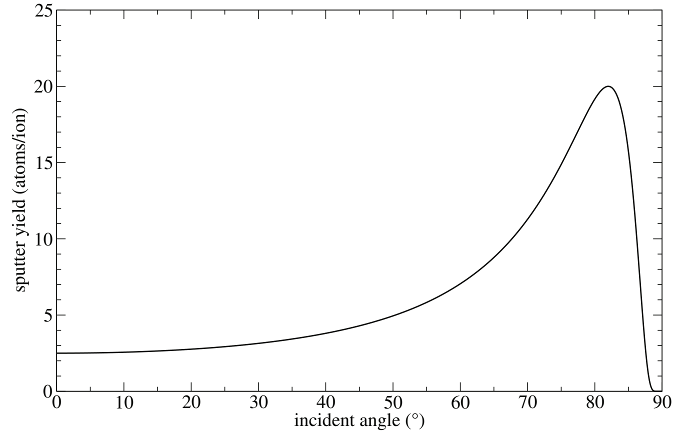


Fig. 1. The sputter yield vs. incident angle as used for all simulations presented in this work.

where C_1, C_2, and C_3 are fitting parameters. This function exhibits a maximum at inclined incidence for $C_2 > C_3$ and is decreasing towards zero for $\theta \to 90°$. In this work, the fitting parameters are chosen in such a way that the yield is 2.5 atoms per ion for normal incidence, while a maximum yield of 20 sputtered atoms per incident ion is obtained for $\theta = 82°$ ($C_1 = 2.5$, $C_2 = 1.87$, and $C_3 = 0.26$). Fig. 1 shows the corresponding yield function as used in this work.

Realistic simulations also require the incorporation of redeposition effects [5]–[7], [10]. Sputtered particles are able to stick to any other reachable parts of the surface. Since the mean free path of sputtered particles is much larger than the typical feature sizes, ballistic particle transport can be assumed. The angular distribution of sputtered particles is described by a cosine distribution [7].

The surface velocity V at a certain surface point $\vec{x}$ is given by the difference between the redeposition rate $\gamma \cdot F_{\mathrm{sp}}(\vec{x})$ and the sputter rate $Y(\theta) \cdot F(\vec{x})$

$$V(\vec{x}) = \frac{1}{n_{\mathrm{Si}}} \left(\gamma \cdot F_{\mathrm{sp}}(\vec{x}) - Y(\theta) \cdot F(\vec{x})\right). \tag{3}$$

Here $n_{\mathrm{Si}} = 5 \times 10^{28}\,\mathrm{m}^{-3}$ denotes the atomic density of silicon, γ is the sticking probability of sputtered particles on the surface, which is assumed to be unity ($\gamma = 1$) in this work, and F_{sp} is the arriving flux density of sputtered particles.

978-1-4244-7700-5/10/$26.00 ©2010 IEEE

III. NUMERICAL METHODS

Two different tasks must be consecutively carried out in order to calculate the altering wafer surface geometry over time. First, the surface rates must be determined, and second, the surface must be accordingly advanced.

A. Monte Carlo Ray Tracing

In order to obtain the surface velocities as given by (3), Monte Carlo ray tracing is applied. This technique has already been successfully demonstrated for other topography simulations [11], [12]. The idea is to simulate a large number of particles and to sum up their individual contributions to the local surface rates. Since ballistic transport is assumed for all particles, all trajectories are straight lines, which enables the application of advanced ray tracing algorithms.

Every simulated ion hitting the surface decrements the local surface rate by a value proportional to the sputter yield given by the incidence angle. A new particle representing all sputtered atoms, thus with a weight factor proportional to the sputter yield, is launched from the point of impact. If the particle trajectory reaches any other part of the surface, the local surface rate is incremented by a value proportional to the weight factor in order to account for the redeposition of sputtered particles.

The simple model presented in the previous section does not consider specular-like reflections of ions, secondary sputtering, non-cosine angular distributions of sputtered particles, or sticking probabilities less than unity. However, it was shown that the inclusion of such effects is necessary for some FIB processes to obtain appropriately accurate simulation results [13]. Apparently, these effects can be straightforwardly incorporated using the Monte Carlo approach. Another advantage is the inherent consideration of shadowing.

Since all particle trajectories are independent from each other within the ballistic transport regime, their calculation can be easily distributed over multiple threads, which makes this Monte Carlo technique very suitable for modern multi-core processor machines. As done in this work, the parallelization can easily be realized for shared memory architectures using OpenMP [14].

B. Level Set Method

For the description of the surface evolution the LS method was used, which is the most popular technique for three-dimensional topography simulations. It represents the surface S implicitly as the zero LS of a function $\Phi(\vec{x})$ [8]. In the corresponding equation of motion

$$\frac{\partial \Phi}{\partial t} + V(\vec{x})\|\nabla\Phi(\vec{x})\| = 0, \quad S = \{\vec{x} : \Phi(\vec{x}) = 0\}, \tag{4}$$

$V(\vec{x})$ is a velocity field obtained through extrapolation of the surface velocities (3). The LS equation (4) is commonly solved on regular grids using upwind finite difference schemes. In this work a first-order forward Euler method is used for time integration. To guarantee stability, the Courant-Friedrichs-Lewy (CFL) condition must be satisfied, which limits the advancement of the surface to a maximum of one grid spacing per time step.

For advanced topography simulation models which consider particle transport, the computation time of the surface velocities is usually negligible compared to that of the surface evolution [11], [12]. However, this is not true for FIB simulations. Since the particle incidence is concentrated within the ion beam, much less particles need to be simulated in order to obtain sufficiently accurate surface rates. As a consequence, the surface evolution plays a major role in the total computation time which needs to be reduced by fast and optimized algorithms.

We have recently developed [15] a fast LS framework which combines the sparse field method [16] and the hierarchical run-length encoded LS data structure [17] in order to reduce the computation time as well as the memory requirements. However, due to the parallel implementation of the Monte Carlo ray tracing algorithm, a parallelization of the LS framework was also necessary as described in [12] in order to avoid a critical bottleneck, when running the simulation on multi-core machines.

IV. RESULTS

The numerical techniques are demonstrated by means of some examples presented in the following. For all simulations one million trajectories of ions and sputtered particles are calculated at every time step. The grid spacing was $2\,\mathrm{nm}$ and the time step was automatically chosen in accordance with the CFL condition in such a way, that the maximum advancement of the surface was less than 0.1 grid spacings. All examples have been calculated on a machine with four six-core AMD Opteron 8435 processors clocked at $2.6\,\mathrm{GHz}$.

A. Dot Milling

Fig. 2 shows the effect of an ion beam on a step structure for inclined incidence with angle $45°$. The step height was $0.4\,\mu\mathrm{m}$ and the lateral extensions $0.8\,\mu\mathrm{m} \times 0.6\,\mu\mathrm{m}$, which corresponds to lateral grid extensions of 400×300.

Using the LS method, the topological change which occurs, when milling through the structure, can be treated appropriately. Furthermore, shadowing is inherently incorporated by the Monte Carlo ray tracing algorithm. 2050 time steps were necessary to simulate the first $80\,\mathrm{ms}$.

In order to test the entire simulation algorithm, the calculation was performed using different numbers of threads. Table I summarizes the corresponding average measured computation times for a single time step as well as the parallel efficiencies.

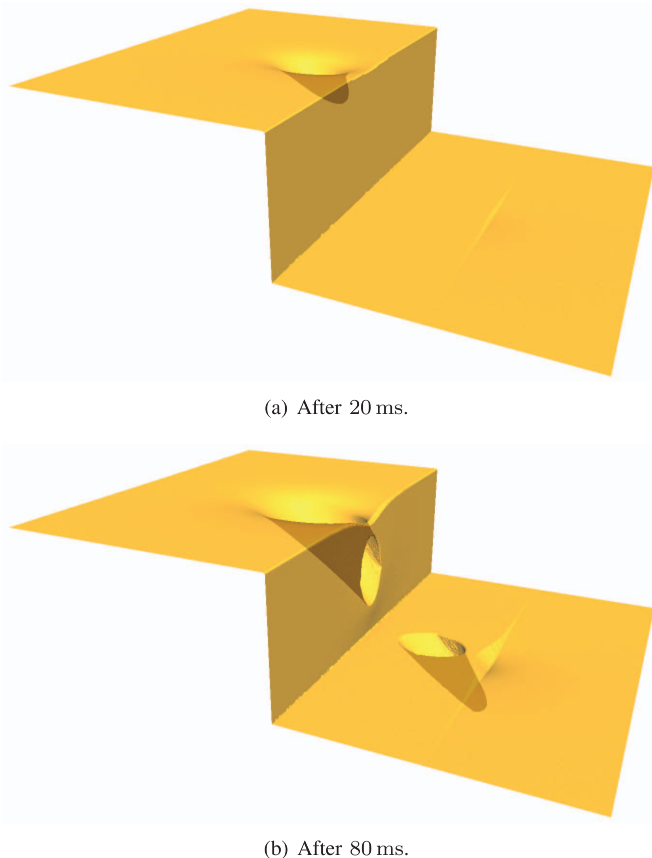
B. Box Milling

The creation of a certain structure requires the movement of the ion beam over the wafer surface. To obtain the desired structure an appropriate scanning strategy must be found. Due to redeposition of sputtered particles and the dependence of the sputter yield on the inclination angle of the surface the prediction of the final depth is not possible by just calculating the total ion dose at every surface point. The direct solution

TABLE I

PARALLEL BENCHMARKS OF OUR SIMULATION APPROACH.

Threads	Calculation time per time step	Parallel efficiency
1	15.14 s	100.0 %
2	8.48 s	89.2 %
4	4.77 s	79.3 %
8	2.47 s	76.5 %
16	1.28 s	73.8 %
24	0.92 s	68.5 %



(a) After 20 ms.

(b) After 80 ms.

Fig. 2. Ion beam milling of a step structure for an incident angle of 45°.

of this inverse problem is still unsolved for the general case [10]. Trial and error seems to be the only way to find a good scanning strategy, especially for three-dimensional problems. Fast simulation techniques are necessary to accelerate this procedure.

As an example, a box milling process as described in [7] is analyzed for two different scanning strategies. The time evolution of a plain surface, while serpentine scanning (see Fig. 3) over 30×6 pixels with a dwell time of $4\,\mathrm{ms}$ using a normal incident beam, is shown in Fig. 4. The pixel spacing, the distance between the center points of neighboring pixels, was $25\,\mathrm{nm}$. The simulation domain was $1\,\mu\mathrm{m} \times 0.4\,\mu\mathrm{m}$ corresponding to lateral grid extensions of 500×200. 8 threads were used for the calculation, resulting in a total computation time of about $13.6\,\mathrm{h}$. 32 300 time steps were necessary and the average computation time for a single time step was $1.52\,\mathrm{s}$.

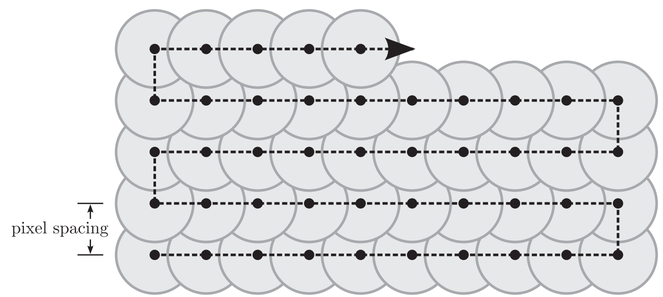


Fig. 3. Illustration of a serpentine scan strategy.

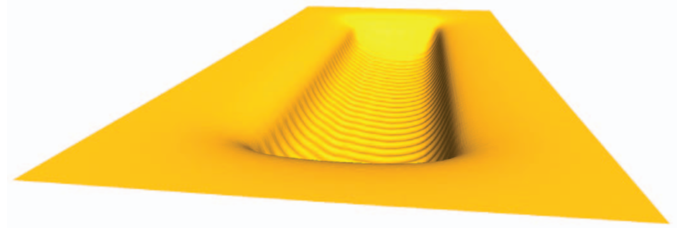


Fig. 5. Final surface profile after four passes of a serpentine scan over 30×6 pixels with a dwell time of $1\,\mathrm{ms}$.

Fig. 5 shows the result for another scanning strategy. Instead of scanning the surface once with a dwell time of $4\,\mathrm{ms}$, four passes with a dwell time of $1\,\mathrm{ms}$ are applied. Although the ion dose is the same for all pixels, the final structure is completely different. The simulation of the multi-pass box milling process required 25 600 time step with an average calculation time of $1.24\,\mathrm{s}$, which led to a total calculation time of $8.8\,\mathrm{h}$ using again 8 threads. The results of both scanning strategies are in good agreement with those presented in [7].

V. CONCLUSION

The use of the LS method for the surface evolution of FIB processes allows the inherent treatment of possible topological changes. Shadowing and redeposition of sputtered particles are accurately considered using a Monte Carlo ray tracing approach. This technique is also suitable to calculate with more detailed models which consider secondary sputtering or more accurate angular distributions of sputtered particles. In order to reduce the computation time, parallelization is used for the surface rate calculation and for the surface evolution.

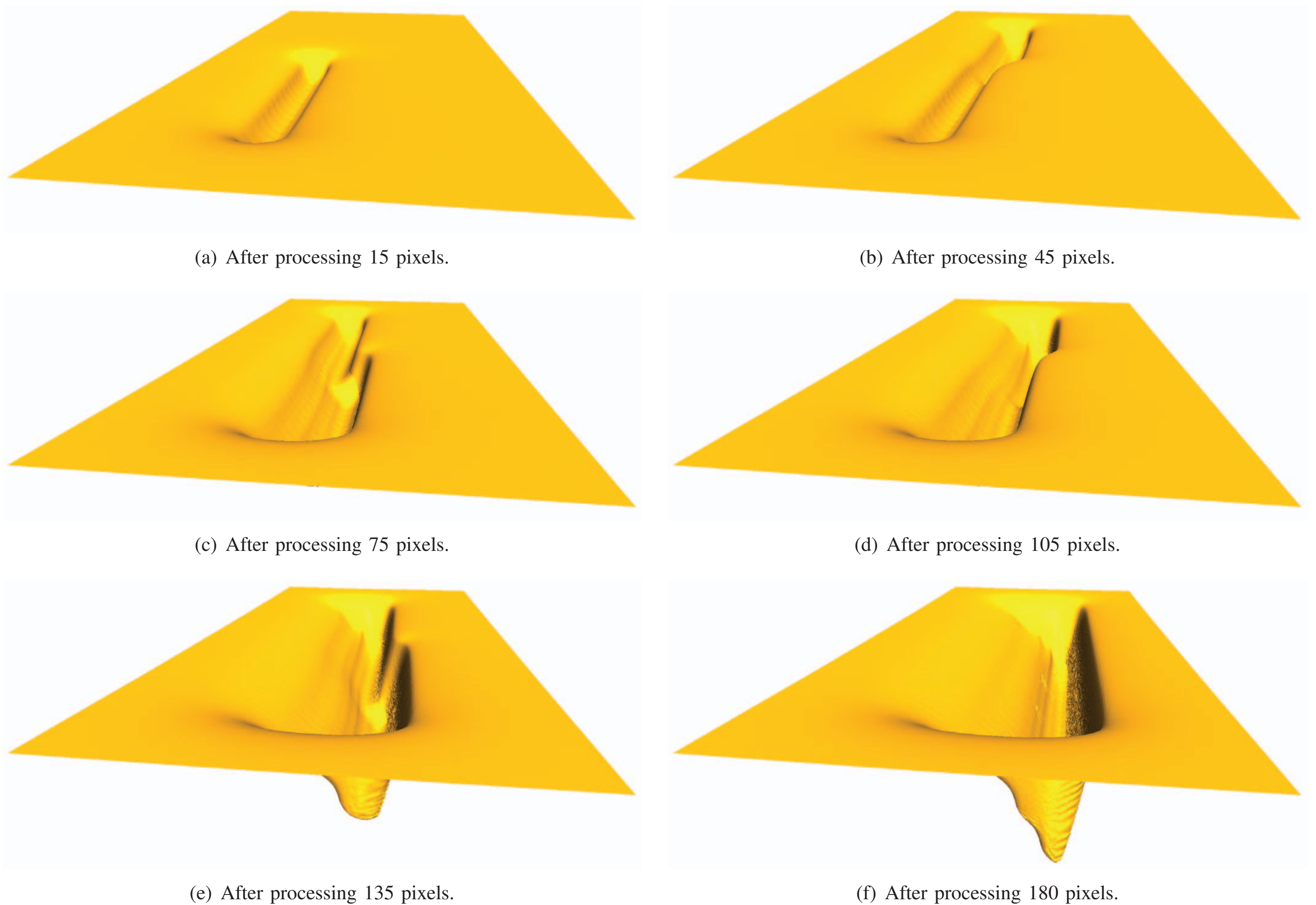
REFERENCES

[1] S. Reyntjens and R. Puers, "A review of focused ion beam applications in microsystem technology," *Journal of Micromechanics and Microengineering*, vol. 11, no. 4, pp. 287–300, 2001.

[2] A. A. Tseng, "Recent developments in micromilling using focused ion beam technology," *Journal of Micromechanics and Microengineering*, vol. 14, no. 4, pp. R15–R34, 2004.

[3] I. Utke, P. Hoffmann, and J. Melngailis, "Gas-assisted focused electron beam and ion beam processing and fabrication," *Journal of Vacuum Science and Technology B*, vol. 26, no. 4, pp. 1197–1276, 2008.

[4] E. Platzgummer, A. Biedermann, H. Langfischer, S. Eder-Kapl, M. Kuemmel, S. Cernusca, H. Loeschner, C. Lehrer, L. Frey, A. Lugstein, and E. Bertagnolli, "Simulation of ion beam direct structuring for 3D nanoimprint template fabrication," *Microelectronic Engineering*, vol. 83, no. 4-9, pp. 936–939, 2006.



(a) After processing 15 pixels.

(b) After processing 45 pixels.

(c) After processing 75 pixels.

(d) After processing 105 pixels.

(e) After processing 135 pixels.

(f) After processing 180 pixels.

Fig. 4. Box milling with a single serpentine scan over 30×6 pixels with dwell time of $4\,\mathrm{ms}$.

[5] H.-B. Kim, G. Hobler, A. Steiger, A. Lugstein, and E. Bertagnolli, "Level set approach for the simulation of focused ion beam processing on the micro/nano scale," *Nanotechnology*, vol. 18, no. 26, p. 265307, 2007.

[6] J. Han, H. Lee, B.-K. Min, and S. J. Lee, "Prediction of nanopattern topography using two-dimensional focused ion beam milling with beam irradiation intervals," *Microelectronic Engineering*, vol. 87, no. 1, pp. 1–9, 2010.

[7] H.-B. Kim, G. Hobler, A. Steiger, A. Lugstein, and E. Bertagnolli, "Full three-dimensional simulation of focused ion beam micro/nanofabrication," *Nanotechnology*, vol. 18, no. 24, p. 245303, 2007.

[8] S. Osher and J. A. Sethian, "Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations," *Journal of Computational Physics*, vol. 79, no. 1, pp. 12–49, 1988.

[9] Y. Yamamura, Itakawa, and N. Y., Itoh, "Angular dependence of sputtering yields of monatomic solids," Institute of Plasma Physics, Nagoya University, Tech. Rep. IPPJ-AM26, 1983.

[10] P. Nellen, V. Callegari, J. Hofmann, E. Platzgummer, and C. Klein, "FIB precise prototyping and simulation," *Materials Research Society Symposia Proceedings*, vol. 960, pp. 35–47, 2007.

[11] O. Ertl and S. Selberherr, "Three-dimensional level set based Bosch process simulations using ray tracing for flux calculation," *Microelectronic Engineering*, vol. 87, no. 1, pp. 20–29, 2010.

[12] O. Ertl, "Numerical methods for topography simulation," Dissertation, Vienna University of Technology, Institute for Microelectronics, 2010. [Online]. Available: http://www.iue.tuwien.ac.at/phd/ertl/

[13] C. Ebm and G. Hobler, "Assessment of approximations for efficient topography simulation of ion beam processes: 10 keV Ar on Si," *Nuclear Instruments and Methods in Physics Research B*, vol. 267, no. 18, pp. 2987–2990, 2009.

[14] "OpenMP C and C++ application program interface." [Online]. Available: http://www.openmp.org

[15] O. Ertl and S. Selberherr, "A fast level set framework for large three-dimensional topography simulations," *Computer Physics Communications*, vol. 180, no. 8, pp. 1242–1250, 2009.

[16] R. T. Whitaker, "A level-set approach to 3D reconstruction from range data," *International Journal of Computer Vision*, vol. 29, no. 3, pp. 203–231, 1998.

[17] B. Houston, M. B. Nielsen, C. Batty, O. Nilsson, and K. Museth, "Hierarchical RLE level set: A compact and versatile deformable surface representation," *ACM Transactions on Graphics*, vol. 25, no. 1, pp. 151–175, 2006.