

Document downloaded from:

<http://hdl.handle.net/10251/65697>

This paper must be cited as:

Montoliu Álvaro, C.; Ferrando Jódar, N.; Cerdá Boluda, J.; Colom Palero, R.J. (2014). Application of the level set method for the visual representation of continuous cellular automata oriented to anisotropic wet etching. *International Journal of Computer Mathematics*. 91(1):124-134. doi:10.1080/00207160.2013.801464.



The final publication is available at

<http://dx.doi.org/10.1080/00207160.2013.801464>

Copyright Taylor & Francis (Routledge): STM, Behavioural Science and Public Health Titles

Additional Information

Application of the Level Set Method for the Visual Representation of Continuous Cellular Automata Oriented to Anisotropic Wet Etching

C. Montoliu^{†*}, N. Ferrando[‡], J. Cerdá[†] and R. J. Colom[†]

[†]*Instituto de Instrumentación para Imagen Molecular (I3M). Centro mixto CSIC Universitat Politècnica de València CIEMAT, camino de Vera s/n, 46022 Valencia, Spain*

[‡]*Centro de Física de Materiales, centro mixto CSIC-UPV/EHU and Donostia International Physics Center (DIPC) 20018 Donostia-San Sebastian, Spain*

Atomistic models are a very valuable simulation tool related with material engineering. This is the case of the Continuous Cellular Automata (CCA), that can simulate accurately the process of chemical etching used in Micro-Electro-Mechanical-Systems (MEMS) micromachining, which can create a wide range of three-dimensional structures. Due to CCA atomistic nature, simulation results are represented as a cloud of points, so data visualization has been usually problematic. When using these models as part of a CAD tool, good data visualization is very important. In this paper, a minimum energy model implemented with the Level Set (LS) method for improve the visual representation related to MEMS engineering is presented. In addition, the Sparse Field Method (SFM) has been applied to reduce the high computational cost of the original LS. Finally, some reconstructed surfaces with completely different topologies are presented, proving the effectiveness of our implementation and how is capable of produce any surface, flats and smooth ones.

Keywords: Level Set; Cellular Automata; Surface Reconstruction; MEMS

1. Introduction

Anisotropic wet chemical etching is one of the most popular bulk micromachining methods for the fabrication of Micro-Electro-Mechanical Systems (MEMS). By using this process it is possible to get suspended microstructures with both flats and smooth surfaces. Another interesting point of this process is its low cost. However, the final result of a particular etching process depends on many factors so it is hard to predict the resulting structure. Some of these factors are: the crystal orientation of the surface, the applied etchant (e.g., potassium hydroxide (KOH) or tetramethylammonium hydroxide (TMAH)), etchant concentration, temperature or the inclusion of additives, such as Triton X-100 [33] or isopropyl alcohol [41, 45, 46].

In order to ease the MEMS design, an important effort has been made through last years in order to accurately model and simulate the process for microengineering applications. First simulators of the wet etching process were based on geometrical models [8] which understand the etching process as a set of moving flats.

On the other hand, Cellular Automata (CA) models the moving surface as a collection of points that represent the substrate. The etching process is simulated

*Corresponding author. Email: carmonal@upv.es

by making surface atoms to disappear according to some microscopic rules, letting the neighbouring sites to emerge into the surface. The first CA models used to simulate the wet etching process are based on stochastic models [39], but many models based on deterministic CA have been presented in recent years [9, 10, 18, 24]. Specially, the Continuous CA (CCA) [44] has demonstrated to be the most accurate. This model takes into account several physical phenomena, such as the *step flow* process [17, 19], in order to accurately simulate wet etching process. Also, some accelerations of simulations have been presented [14, 15] and the CCA based simulator IntelliEtch [25] has the capability of execute the algorithm on Graphics Processing Units (GPU) reducing drastically the simulation times. The CA models are currently accepted as the most adequate models for simulating this process in terms of performance and accuracy.

Despite the accurate results obtained with CCA, the final result is a cloud of unconnected points, so it is hard to visualize correctly some details of the surface, specially at complicated topologies. In addition, the CCA approach introduces noise due to the calibration process required for obtaining the atomistic etch rates from the experimental macroscopic measurements [20]. The method currently used in [25] for improving the result visualization is to shade the points depending on its normal vector, but in complex morphologies of some structures the shading accuracy of the method is not good enough and adds too much noise, so the visualization quality decreases greatly. Thus, in order to improve the final visualization it is necessary to obtain a continuous surface from the information of unconnected points. Also, it is important to consider all the different kinds of topologies that can be obtained due to anisotropic wet etching. There are many methods for surface reconstruction [5, 11, 23, 30, 31]. Specially, triangulation methods such as Delaunay triangulations and Voronoi diagrams are very popular [6, 7, 13]. Although these techniques can provide suitable results in many applications, they are not adequate when the data is noisy. Furthermore, they generally require a constant density of points, and in some CCA results of a wet etching process there are complex parts that only are formed by a few atoms, decreasing the accuracy at those sensitive structures. On the other hand, the LS method has proven to be robust even when the data is noisy and is formed by different densities [42, 43]. Accordingly, in this study an implementation of the Level Set (LS) method is presented.

The rest of the paper is organized as follows. The fundamentals of LS method as well as numerical techniques needed are introduced in section 2. The optimization implemented to reduce computation cost is also explained in this section. In section 3 the algorithm developed to reconstruct surfaces is shown. This is followed by the reconstructed surfaces of some examples shown in section 4. The improvement obtained with our method is proved by comparing the results with the CCA simulator results, and also performance data is presented. Finally, conclusions of this study are shown.

2. Level Set method applied to surface reconstruction

LS method was introduced for capturing moving fronts [32], however this method has been used in many fields, such as: image segmentation [34], inverse problems [35], seismology [37] and also chemical etching [2–4]. The main idea of this method consists in embed the front $x(t)$ in a signed distance function ϕ such as $\phi(x(t), 0) = 0$. Therefore, the movement is applied to the function ϕ instead of the front itself. In the specific case where the movement is in the normal direction, the equation

movement is known as LS equation and turns out:

$$\phi_t + R(x(t), t) |\nabla\phi| = 0 \quad (1)$$

where $R(x(t), t)$ is the velocity of the front in its normal direction. This velocity at front points is determined by the physical laws of the process being simulated. We assume that this velocity is defined on the whole domain. The LS equation can also be written in Hamilton-Jacobi form:

$$\phi_t + H(\nabla\phi(x(t), t)) = 0 \quad (2)$$

where $H = R(x(t), t) |\nabla\phi|$ is the Hamiltonian. A Hamiltonian is convex if the following condition is satisfied:

$$\frac{\partial^2 H}{\partial\phi_x \partial\phi_y} \geq 0 \quad (3)$$

where ϕ_x and ϕ_y are the spatial derivatives of $\phi(x(t), t)$ respect of x and y dimensions respectively. In the case of a convex Hamiltonian, Osher *et al.* [32] presented an upwind difference scheme in order to choose appropriately spatial derivatives of ϕ to ensure the convergence of the front. On the other hand, when the Hamiltonian depends on the geometry of the front, usually condition (3) is not satisfied and this technique cannot be applied.

In the LS method, the zero-level of the ϕ function is the propagating front between both phases, and the main idea when LS is applied to surface reconstruction is to build an initial surface exterior to the cloud of points, embed it in the ϕ function and move it to the cloud until it is close enough. Despite several models for LS image reconstruction have been presented [28, 40], in this study the minimal energy model presented in [43] has been chosen due to this model does not need any additional information, only the data points (from now on referred as set S), and also because CCA resulting data set can present very different topologies. This energy model defines an energy function of a surface which is proportional to the sum of the distance values of those points that form the surface.

This energy function is defined as:

$$E(\Gamma) = \left[\int_{\Gamma} dist^m(\vec{x}) ds \right]^{1/m} \quad (4)$$

where Γ is a surface, ds the surface area, $m \in [1, +\infty)$ variates the smoothness of the surface and $dist(\vec{x})$ is the unsigned distance between surface points and the S points. In this study, we have determined that $m = 1$ is a good choice to obtain accurate results. In this model, a minimal energy surface is reached as a final solution, so initial surface is evolved following the gradient descent of the energy function. Thus, by extending this variation to the rest of the ϕ levels, the movement equation turns out:

$$\phi_t = \left[\nabla dist(\vec{x}) \cdot \vec{N} + dist(\vec{x})\kappa \right] |\nabla\phi| \quad (5)$$

where \vec{N} is the normal vector and κ is the local curvature. The first term of equation (5) needs to be solved with upwind differencing technique since it satis-

fies the equation (3). In this study first order derivatives has been used as they have demonstrated to be sufficiently accurate and they are the least expensive computationally. However, if more accuracy would be required, the order of these derivatives could be increased, e.g. using Weighted Essential Non-Oscillatory scheme [26].

On the other hand, when ϕ is a signed distance function, the curvature can be defined by $\kappa = \phi_{xx} + \phi_{yy} + \phi_{zz}$. Upwind differencing scheme cannot be applied to solve this term since it depends on local geometry of the surface, thus, due to the second derivatives of κ , we use central second order derivatives to ensure convergence. Because of this derivatives, it is necessary to take a time step proportional to grid resolution squared ($(\Delta x)^2$). We use a first order forward Euler time discretization since it is the least computationally expensive. Thus, equation (5) results:

$$\phi^{n+1} = \phi^n + \Delta t [\nabla D \nabla \phi^n + D \kappa^n |\nabla \phi^n|] \quad (6)$$

where D is the distance matrix between grid points and S set points.

2.1 Sparse Field Method

The LS method has a high computational cost ($O(N^3)$), where N is the number of grid points in each dimension, since every point of the three-dimensional grid needs to be updated by equation (6). In order to reduce this computational cost Adalsteinsson *et al.* introduced the Narrow Band (NB) optimization [1]. This technique consists of updating only those layers of points which are within a close region of the front itself. Applying this technique, computational cost is reduced to $O(kN^2)$, where k is the width of the close region. Despite advantages of NB method, it is still not optimum as more points than the strictly necessary are computed. Furthermore, a mechanism is necessary to build the signed distance function every time the front reaches the boundary of the close region, introducing an additional computational cost.

In order to solve these problems, the Sparse Field Method (SFM) was introduced [40]. SFM reduces the active region near to the front to only those points strictly necessary to update the front, resulting in a computational cost of only ($O(N^2)$). Furthermore, in each time step, a new signed distance function is built, thus, it is not necessary to apply any reinitialization technique [16].

In this study, only 3 sets of points need to be stored as lists to be updated, namely: the list containing the points of the front ($L0$), the list which has outer points at Δx distance from the front ($Lp1$), and another list with the inner points at Δx distance from the front ($Ln1$). Notice that in this study, the signed distance function ϕ has been taken to be positive outside the surface and negative inside it. The rest of the points of the lattice that are not included in any list are kept with constant values and do not need to be updated. Depending on the list where each point is stored, the active points are labelled with different states: 1, 0, -1 for $Lp1$, $L0$, $Ln1$ respectively. Figure 1 shows a two-dimensional example of a front and the necessary lists of active points.

For each time step, only the new values of ϕ for those points included in $L0$ are calculated with equation (6). On the other hand, for points of $Lp1$ ($Ln1$) the algorithm searches for the nearest point belonging to $L0$ and add (subtract) Δx to the distance value. Finally, for every point of the front, the neighbouring points that are not included in any list are added to the corresponding list $Lp1$ or $Ln1$ depending on if the value is positive or negative. Our LS implementation only needs backward and forward first order derivatives, thus only three voxels lists must be

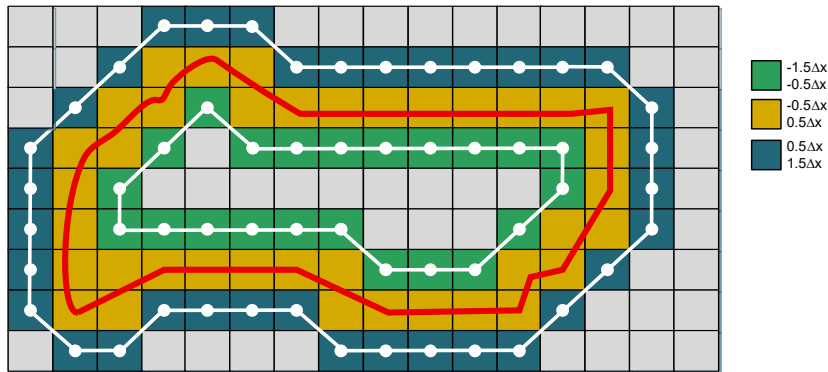


Figure 1. Two-dimensional example of a front. Green voxels represent the $Ln1$ list, blue voxels form the $Lp1$, orange voxels represent the $L0$ list used in SFM. Grey voxels are kept with a constant value. Also, respective distance value ranges of each list are shown.

maintained. In order to increase the accuracy, a higher order k of derivatives can be chosen, thus, the points at distance $\leq |\pm(k)\Delta x|$ need to be maintained in their corresponding lists.

3. Algorithm implementation

In this section the algorithm developed to reconstruct the surface from the S points obtained with the CCA simulator is presented. The pseudocode of our implementation based on SFM is the following:

- (1) Build a uniform grid over the whole space occupied by S . The resolution of the grid is chosen to be proportional to the minimal distance between S points: $\Delta x = n \cdot \min_dist$, where $n > 0$ depending on the desired accuracy of the final result. The value of \min_dist is already known since we know the crystallographic structure from the CCA model.
- (2) Calculate distance matrix D . The direct form to calculate this matrix has a computational cost $O(N^3 \cdot N_S)$ being N_S the number of S points and N is the number of grid points in one dimension, since for each grid point it is necessary to calculate the distance value with every S point and get the minimum one. In order to reduce the computational cost, a growing algorithm has been developed, which consists in:
 - 1) Discretize S points over the grid and obtain the closest grid points. Calculate exact distance value between those pairs of points.
 - 2) These grid points propagate the coordinates of its S closest point to its neighbours grid points.
 - 3) Those neighbouring points calculate the distance respect the S point that have received. If more than one S point are received, the one coming from the grid point with the smallest distance value is chosen.
 - 4) Go to 2) until every grid point has calculated its own distance value.
With this growing algorithm, each grid point only needs to calculate a single distance value and only accesses to its neighbouring points, thus, the computational cost is reduced to $O(N^3)$. Also, the mean relative error respect of the exact distance matrix introduced by this algorithm is only about 2%, and in our tests this approximation did not affect the final result.
- (3) Find an initial surface exterior to S . An initial surface far from the S set requires more steps to reach the final solution. We have found that a suitable initial surface is formed by those exterior grid points that have a distance

value $2\Delta x$. In order to find this initial surface the simple algorithm shown in [42] has been used.

- (4) Embed the initial surface into the signed distance function ϕ . According to SFM, it is only necessary to calculate distance for those points close to the front whereas the rest of the points are kept to a constant value.
- (5) Include points in its respective list depending on distance values.
- (6) Calculate second order central derivatives of ∇D .
- (7) Calculate time step Δt according to Courant-Friedrichs-Levy (CFL) condition [12] for parabolic PDE. We have found that the maximum possible value of time step in order to avoid surface divergence is: $\Delta t = \frac{\Delta x^2}{2 \cdot \max\{D\}}$.
- (8) Evolution surface loop:
 - 1) Calculate the term $\nabla\phi^n$ with upwind differencing and first order derivatives, and $\kappa|\nabla\phi^n|$ with central derivatives for every points included in list $L0$.
 - 2) Calculate new values of ϕ with (6) and check if those new values are still in $L0$ range, *i.e.*, for every point of this list:
 - If $\phi_{i,j,k}^{n+1} < -0.5\Delta x$ remove it from $L0$ and add it to changing status list point $Sn1$.
 - If $\phi_{i,j,k}^{n+1} > 0.5\Delta x$ remove it from $L0$ and add it to changing status list point $Sp1$.
 - 3) For every point (i, j, k) of $Lp1$ list:
 - i. Find the $L0$ point (i_n, j_n, k_n) , among its neighbours, with the smallest distance value ϕ_{i_n, j_n, k_n} and update its distance value $\phi_{i,j,k}$ by adding Δx to the value of this point, such as: $\phi_{i,j,k} = \phi_{i_n, j_n, k_n} + \Delta x$. If this new value is within $L0$ range, set it to $S0$.
 - ii. If no $L0$ neighbour point is found, the point (i, j, k) is removed from list $Lp1$.
 - 4) For every point (i, j, k) of $Ln1$ list:
 - i. Find the $L0$ point (i_n, j_n, k_n) , among its neighbours, with the maximum distance value ϕ_{i_n, j_n, k_n} and update its distance value $\phi_{i,j,k}$ by subtracting Δx to the value of this point, such as: $\phi_{i,j,k} = \phi_{i_n, j_n, k_n} - \Delta x$. If this new value is within $L0$ range, set it to $S0$.
 - ii. If no $L0$ neighbour point is found, the point (i, j, k) is removed from list $Ln1$.
 - 5) For every $Sp1$ point:
 - i. Add it to $Lp1$ list.
 - ii. Change state to 1.
 - iii. Remove it from $Sp1$ list.
 - 6) For every $Sn1$ point:
 - i. Add it to $Ln1$ list.
 - ii. Change state to -1 .
 - iii. Remove it from $Sn1$ list.
 - 7) For every $S0$ point:
 - i. Add it to $L0$ list.
 - ii. Change state to 0.
 - iii. Remove it from $S0$ list.
 - 8) For every point (i, j, k) of $L0$ list:
 - i. Search for neighbouring points with $state = 2$, (i_{np}, j_{np}, k_{np}) , and:
 - Change state to 1.
 - Update its distance value $\phi_{i_{np}, j_{np}, k_{np}} = \phi_{i,j,k} + \Delta x$.
 - Add it to $Lp1$ list.
 - ii. Search for neighbouring points with $state = -2$, (i_{nn}, j_{nn}, k_{nn}) ,

and:

- Change state to -1 .
- Update its distance value $\phi_{i_{nn},j_{nn},k_{nn}} = \phi_{i,j,k} - \Delta x$.
- Add it to $Ln1$ list.

9) Go to step (8) until the surface is close enough to the S set points.

(9) Extract final surface from signed distance function and represent it. We use marching cubes [29] method in order to solve this step.

The computational cost of our implementation is primarily due to step (8), *i.e.* the evolution loop of the front, which represents about 50-75% of the total execution time, depending on the morphology of the structure and the resolution of the grid. Step (2) is also relevant, representing 22-42% of the simulation time. Thus, steps (8) and (2) altogether represent about 95% of the simulation time. Table 1 shows the simulation time of these parts of the algorithm for several sizes of the grid when executing example a) of figure 2. The results show that the scaling behaviour of the whole algorithm is not exactly $O(N^2)$ due to the high computational cost of step (2). Nevertheless, the computational time of step (8) scales more accord to $O(N^2)$ behaviour.

Grid size	Execution time (s)	Step (2)	Step (8)
$143 \times 406 \times 19$	16.3	6.3(2)/38%	9.4(s)/57%
$274 \times 800 \times 26$	40.6	8.9(s)/22%	29.6(s)/73%
$535 \times 1588 \times 40$	175.1	45.5(s)/26%	45.5(s)/65%

Table 1. Computational time of the total execution, step (2) and step (8) of the algorithm, for different grid sizes of the experiment shown in figure 2. The data shown in third and fourth column are the time required by the corresponding step and the fraction of the total execution that represents.

4. Results

In this section some reconstructed surfaces with our implementation are shown, as well as performance results. The implementation is developed in pure Java and the testing machine consists of an Intel Core i7 at 2.8GHz with 8GB of RAM with 64 bit Windows-based server Java Virtual Machine version 1.7.0_03. In addition, results obtained with CCA method [25] are also shown in order to compare the improvement obtained with the proposed technique. Three examples of experiments are collected in figure 2. Notice that shading points technique has been applied to the direct CCA cloud of points in order to obtain the CCA results shown in blue. Figures represented in green show the results of the reconstructed surface with our LS method implementation. Gouraud technique [22] is applied to shade the surfaces.

The first example is a three-axis accelerometer presented in [36]. Second example are microneedles [38] used for drug delivery and the last example is a wagon wheel experiment [21] used to characterize the anisotropy of a particular etchant. Some figures show a closeup image of some parts of the main structures and they clearly show the improvement obtained respect to the point cloud of the CCA simulator.

Despite of the completely different topologies that present these three experiments, our LS implementation has been able to reconstruct all of them. In addition, the final surfaces are always continuous even in those complicated parts of the structures. Thus, details of the final structures can be easily visualized. This improvement is clearly shown in example c) of figure 2 since the spokes of the wheel are very thin and it is difficult to determine correctly the normal vector corresponding to each atom for the CCA model. In contrast, our LS implementation

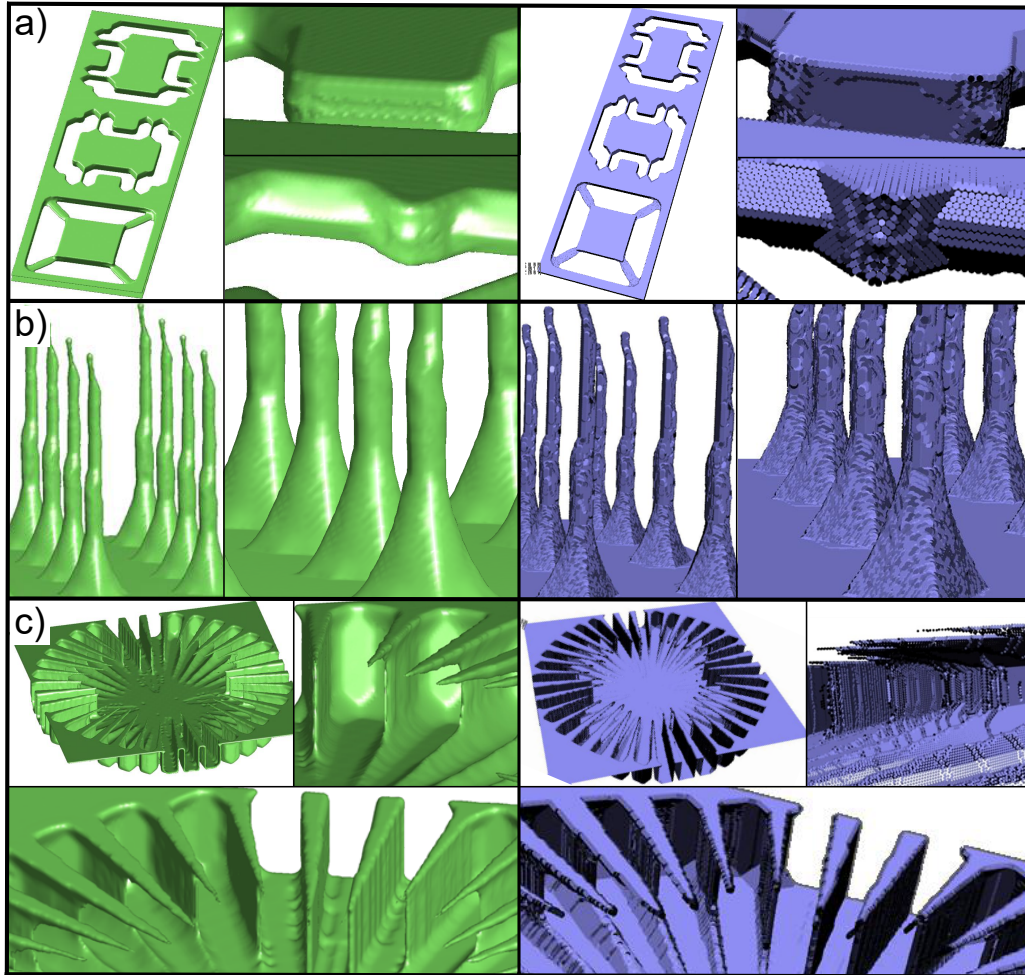


Figure 2. Surfaces reconstructed with our implementation represented in green. Figures represented in blue are the shaded point cloud solution obtained with CCA simulator [25]. a) Three-axis accelerometer [36], b) microneedles [38] and c) wagon wheel [21].

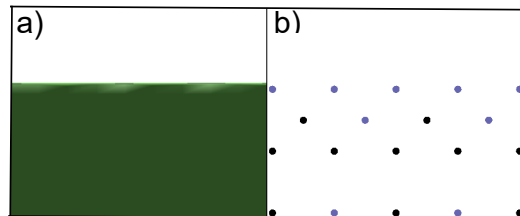


Figure 3. Comparison between the profile representation of the implemented SFM a) and the CCA method b). These profiles correspond to a plane of the example shown in figure 2 a).

is able to reconstruct the surface flawlessly.

Although the atoms of a plane follow the crystallographic structure of the silicon, our implementation represents the planes of the structures as continuous regular surfaces. This feature improves the visualization of the final results as can be observed explicitly in figure 3.

On the other hand, the curvature dependency of the SFM tends to soften the corners as shown in example a) of the figure 2. Depending on the fabricated MEMS purpose, flats intersections might be relevant, thus, in those cases a larger grid should be used to obtain better accuracy.

Performance data of these simulations is collected in table 2. The second column

shows the size of the built grid, the third one shows the number of the points produced by CCA simulator, the fourth column is the number of iterations needed to converge, the fifth one collects the time required by LS algorithm to converge and finally the time required by the CCA simulations are also shown. The CCA times are obtained when executing the method on a GPU Nvidia GeForce 260GTX.

	Grid size	Points of S	Iterations	LS time (s)	CCA time (s)
a)	274x800x26	1109390	73	40.6	10.7
b)	212x212x117	287034	70	19.4	58.2
c)	352x352x49	785471	60	35.1	26.3

Table 2. Summary of the parameters used to obtain simulations results shown in figure 2. Simulations times required by the CCA approach are also presented.

Due to SFM and the low order derivatives that have been used, time required by LS algorithm to converge is only about a few tens of seconds even at example c) which has a grid of more than 6 millions points. On the other hand, due to the optimized GPU-based CCA simulation times are in the same order of magnitude than the visualization process itself, our SFM visualization technique can be used as a final procedure to improve the representation of the MEMS structure. Nevertheless, our algorithm can also take advantage of the acceleration process offered by the massively parallel platforms GPUs as proved in other other studies [27, 34], leaving the possibility of further acceleration processes.

5. Conclusion

In this study it has been demonstrated that:

- LS method is capable to improve visual representation of CCA applied to anisotropic wet etching.
- Our implementation has been able to reconstruct all the tested topologies types.
- Due to SFM optimization, execution time is only about tens of seconds.
- First order derivatives are sufficient to get a good level of details.
- Since the optimized GPU-based CCA simulations requires similar time than the LS visualization technique, the presented implementation can be used as a final procedure to improve the representation of the MEMS structures.

By using our implementation of the LS method, the visual representation of CA based simulators is improved, so it can be very useful to ease the design of MEMS. In addition, execution time could be reduced even more if the algorithm is adapted to a many-core platform like GPUs, as it has been demonstrated previously on similar LS methods [27, 34].

Acknowledgements

We thank to Miguel Ángel Gosálvez for his collaboration in the early stages of this research. This work has been supported by the Spanish FPI-MICINN BES-2011-045940 grant. Also, we acknowledge support by the JAE-Doc grant from the Junta para la Ampliación de Estudios program co-funded by FSE.

References

- [1] D. Adalsteinsson and J.A. Sethian, *A fast level set method for propagating interfaces*, Journal of Computational Physics 118 (1994), pp. 269–277.
- [2] ———, *A level set approach to a unified model for etching, deposition, and lithography i: Algorithms and two-dimensional simulations*, Journal of Computational Physics 120 (1995), pp. 128–144.
- [3] ———, *A level set approach to a unified model for etching, deposition, and lithography ii: Three-dimensional simulations*, Journal of Computational Physics 122 (1995), pp. 348–366.
- [4] ———, *A level set approach to a unified model for etching, deposition, and lithography iii: Redeposition, reemission, surface diffusion, and complex simulations*, Journal of Computational Physics 138 (1997), pp. 193–223.
- [5] R. Allegre, R. Chaine, and S. Akkouché, *A flexible framework for surface reconstruction from large point sets*, Computer Graphics 31 (2007), pp. 190–204.
- [6] N. Amenta, M. Bern, and D. Eppstein, *The Crust and the Beta-Skeleton: Combinatorial Curve Reconstruction*, in *Graphical Models and Image Processing*, 1998, pp. 125–135.
- [7] N. Amenta, M. Bern, and M. Kamvyselis, *A new Voronoi-based surface reconstruction algorithm*, in *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '98, ACM, 1998, pp. 415–421.
- [8] K. Asaumi, Y. Iriye, and K. Sato, *Anisotropic-etching process simulation system microcad analyzing complete 3d etching profiles of single crystal silicon*, IEEE Micro Electro Mechanical Systems (1997), pp. 412–417.
- [9] S. Buttgenbach and O. Than, *Suzana: a 3d cad tool for anisotropically etched silicon microstructures*, ED&TC 96. Proceedings 9 (1996), pp. 454–458.
- [10] H. Camon, A.M. Gue, J.S. Daniel, and M. Djafari-Rouhani, *Modelling of anisotropic etching in silicon-based sensor application*, Sensors and Actuators A 33 (1992), pp. 103–105.
- [11] J.C. Carr, R.K. Beatson, B.C. McCallum, W.R. Fright, T.J. McLennan, and T.J. Mitchell, *Smooth surface reconstruction from noisy range data*, in *Proceedings of the 1st international conference on Computer graphics and interactive techniques in Australasia and South East Asia*, GRAPHITE '03, Melbourne, Australia, URL <http://doi.acm.org/10.1145/604471.604495>, ACM, New York, NY, USA, 2003, pp. 119–ff.
- [12] R. Courant, K. Friedrichs, and H. Lewy, *Über die partiellen differenzengleichungen der mathematischen physik*, Mathematische Annalen 100 (1928), pp. 32–74.
- [13] H. Edelsbrunner, *Shape Reconstruction with Delaunay Complex*, in *Proceedings of the Third Latin American Symposium on Theoretical Informatics*, LATIN '98, Springer-Verlag, 1998, pp. 119–132.
- [14] N. Ferrando, G.M. A., J. Cerd, R. Gadea, and K. Sato, *Faster and exact implementation of the continuous cellular automaton for anisotropic etching simulations*, Journal of Micromechanics and Microengineering 21 (2011).
- [15] N. Ferrando, M.A. Gosálvez, J. Cerda, R. Gadea, and K. Sato, *Octree-based, gpu implementation of a continuous cellular automaton for the simulation of complex, evolving surfaces*, Computer Physics Communications 182 (2011), pp. 628–640.
- [16] J. Gomes and O. Faugeras, *Reconciling distance functions and level sets*, Journal of Visual Communication and Image Representation 11 (1999), pp. 209–223.
- [17] M.A. Gosálvez, Y. Xing, and K. Sato, *Analytical solution of the continuous cellular automaton for anisotropic etching*, Journal of Micromechanical Systems 17 (2008), pp. 410–431.
- [18] M.A. Gosálvez, R.M. Nieminen, P. Kilpinen, E. Haimi, and V. Lindroos, *Anisotropic wet chemical etching of crystalline silicon: atomistic monte-carlo simulations and experiments*, Applied Surface Science 178 (2001), pp. 7–26.
- [19] M.A. Gosálvez, K. Sato, A.S. Foster, R.M. Nieminen, and H. Tanaka, *An atomistic introduction to anisotropic etching*, Journal of Micromechanics and Microengineering 17 (2007), pp. S1–S26.
- [20] M.A. Gosálvez, N. Ferrando, Y. Xing, P. Pal, K. Sato, J. Cerda, and R. Gadea, *Simulating anisotropic etching of silicon in any etchant: Evolutionary algorithm for the calibration of the continuous cellular automaton.*, Journal of Micromechanics and Microengineering 21 (2011).
- [21] M.A. Gosálvez, P. Pal, N. Ferrando, H. Hida, and K. Sato, *Experimental procurement of the complete 3d etch rate distribution of si in anisotropic etchants based on vertically micromachined wagon wheel samples*, J. Micromech. Microeng. 21 (2011), p. 125007.
- [22] H. Gouraud, *Continuous shading of curved surfaces*, Computers, IEEE Transactions on C-20 (1971), pp. 623 – 629.
- [23] H. Hoppe, T. DeRose, T. Duchamp, M. Halstead, H. Jin, J. McDonald, J. Schweitzer, and W. Stuetzle, *Piecewise smooth surface reconstruction*, in *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, SIGGRAPH '94, URL <http://doi.acm.org/10.1145/192161.192233>, ACM, New York, NY, USA, 1994, pp. 295–302.
- [24] T. Hubbard and E.K. Antonsson, *Cellular automata modeling in mems design*, Sensors and Materials 9 (1997).
- [25] IntelliSense-Corp., *Intellietch web page*: <http://www.intellisensesoftware.com/modules/IntelliEtch.html> (2010).
- [26] G.S. Jian and D. Peng, *Weighted eno schemes for hamilton-jacobi equations*, SIAM Journal on Scientific Computing 21 (1997), pp. 2126–2143.
- [27] A.E. Lefohn, J.E. Cates, and R.T. Whitaker, *Interactive gpu-based level sets for 3d segmentation*, Proc. of Medical Image Computing and Computer Assisted Intervention (2003), pp. 564–572.
- [28] H. Liu, X. Wang, and W. Qiang, *Implicit surface reconstruction from 3D scattered points based on variational level set method*, in *Systems and Control in Aerospace and Astronautics, 2008. ISSCAA 2008. 2nd International Symposium on*, dec., 2008, pp. 1 –5.
- [29] W.E. Lorensen and H.E. Cline, *Marching cubes: A high resolution 3d surface construction algorithm*, Computer Graphics 21, no. 4 (1987), pp. 163–169.
- [30] R. Mencl and H. Muller, *Graph-based surface reconstruction using structures in scattered point sets*,

- in *Computer Graphics International. Proceedings*, jun, 1998, pp. 298–311.
- [31] C. Oblonsek and N. Guid, *A fast surface-based procedure for object reconstruction from 3d scattered points*, *Computer Vision and Image Understanding* 69 (1998), pp. 185–195.
- [32] S. Osher and J.A. Sethian, *Fronts propagating with curvature dependent speed: Algorithms based on hamilton-jacobi formulations*, *Journal of Computational Physics* 79 (1988), pp. 12–49.
- [33] P. Pal and K. Sato, *Various shapes of silicon freestanding microfluidic channels and microstructures in one-step lithography*, *Journal Micromechanics and Microengineering* 19, no. 5 (2004).
- [34] M. Roberts, J. Packer, M. Costa-Sousa, and J.R. Mitchell, *A work-efficient gpu algorithm for level set segmentation*, *High Performance Graphics* (2010).
- [35] F. Santosa, *A level set approach for inverse problems involving obstacles*, *ESAIM: Control, Optimisation and Calculus of Variations* 1 (1996).
- [36] G. Schrpfer, M. de Labachellerie, S. Ballandras, and P. Blind, *Collective wet etching of a 3d monolithic silicon seismic mass system*, *J. Micromech. Microeng* 8 (1998), pp. 77–79.
- [37] J.A. Sethian and M. Popovici, *Fast marching methods applied to computation of seismic travel times*, *Geophysics* 64 (1999).
- [38] M. Shikida, M. Ando, Y. Ishihara, T. Ando, K. Sato, and K. Asaumi, *Non-photolithographic pattern transfer for fabricating pen-shaped microneedle structures*, *Journal of Micromechanics and Microengineering* 14 (2004), pp. 1462–1467.
- [39] O. Than and S. Buttgenbach, *Simulation of anisotropic chemical etching crystalline silicon using a cellular automata model*, *Sensors and Actuators A* 45 (1994), pp. 85–88.
- [40] R.T. Whitaker, *A level-set approach to 3d reconstruction from range data*, *International Journal of Computer Vision* 29 (1998), pp. 203–231.
- [41] R.A. Wind, H. Jones, and M.A. Hines, *Macroscopic etch anisotropies and microscopic reaction mechanisms: a micromachined structure for the rapid assay of etchant anisotropy*, *Surface Science* 460 (2000), pp. 21–38.
- [42] H.K. Zhao, S. Osher, and R. Fedkiw, *Fast surface reconstruction using the level set method*, *Proceedings. IEEE Workshop on Variational and Level Set Methods in Computer Vision* (2001), pp. 194–201.
- [43] H.K. Zhao, S. Osher, B. Merriman, and M. Kang, *Implicit and non-parametric shape reconstruction from unorganized data using a variational level set method*, *Computer Vision and Image Understanding* 80 (2000), pp. 295–319.
- [44] Z. Zhu and C. Liu, *Micromachining process simulation using a continuous cellular automata method*, *Journal of Micromechanical Systems* 9 (2000), pp. 252–261.
- [45] I. Zobel and M. Kramkowska, *The effect of alcohol additives on etching characteristics in koh solutions*, *Sensors and Actuators A* 101 (2002), pp. 255–261.
- [46] ———, *Etch rates and morphology of silicon (h k l) surfaces etched in koh and koh saturated with isopropanol solutions*, *Sensors and Actuators A* 115 (2004), pp. 549–556.