Proceedings of the IASTED International Conference
Applied Modelling and Simulation
August 12-14, 1998, Honolulu, Hawaii - USA

OPTIMIZED GEOMETRY PREPROCESSING FOR THREE-DIMENSIONAL SEMICONDUCTOR PROCESS SIMULATION

P. FLEISCHMANN, E. LEITNER, and S. SELBERHERR

Institute for Microelectronics, TU Vienna
Gusshausstr. 27–29, A-1040 Vienna, Austria
Phone +43-1-58801-3859, FAX +43-1-5059224
e-mail: fleischmann@iue.tuwien.ac.at

ABSTRACT

The growing importance to model three-dimensional effects has made geometry preprocessing the bottle neck to efficient three-dimensional simulation in many areas. Especially, for semiconductor process and device simulation, as well as interconnect applications two-dimensional methods which have worked well in the past have become obsolete. More efficient algorithms have to be employed for the vast amount of data in three dimensions. We present a new approach to deal with complex structures that can be automated and allows an optimal volume decomposition adapted to the simulation needs.

KEYWORDS

Simulation, Finite Element mesh generation,
Technology Computer Aided Design (TCAD)

INTRODUCTION

The preprocessing and tessellation of input structures is known to play the critical role in semiconductor device and process simulation. The stiff and highly nonlinear equations governing the behavior of a semiconductor device and the moving boundary and interface situation during oxidation in process simulation require a powerful and efficient preprocessor tool. Proper decomposition of the complex semiconductor structures with multiple thin layers and extreme ratios between smallest and largest feature sizes is a necessary step to achieve good convergence with the typical algorithms applied in the field of semiconductor process and device simulation. The often used optimality criterion which maximizes the minimum angle in two dimensions (Delaunay triangulation [1]) has to be extended to allow additional criteria, e.g. to avoid typical sliver elements in three dimensions. The suitability for efficient local modifications is important to cope with structure deformations due to moving boundaries and interfaces.

The presented approach consists of a Delaunay tetrahedralization as the dual of the Voronoi graph [2] and a surface preprocessor. The tetrahedralization module uses a modified advancing front algorithm. It is provided with the initial seeding front by the surface preprocessor. The modified advancing front technique offers some advantages. The tetrahedralization of the convex hull of the vertices and internal nodes is at no time needed. It can be efficiently restricted to the domain to be tessellated. This can considerably save computation time for extremely non-convex structures. The resulting Delaunay tessellation possesses the flexibility possible within the scope of a tetrahedral representation not like octree based or other cartesian tessellations which are commonly employed in Technology CAD (TCAD). Furthermore, it can be optimized to adapt to other criteria by local modifications of undesirable elements. The combination of efficient Delaunay methods with advancing front techniques is a fairly new development [3].

SURFACE PREPROCESSING

Preprocessing the polygonal surface description of the three-dimensional structure is required prior to the volume processing. The volume decomposition performed by the tetrahedralization module uses a modified advancing front algorithm. The adapted surface description of the geometry gives the optimized seeding front. A general polygonal description of the boundary and interfaces is the required input. After triangulating each polygon the triangles of the resulting surface triangulation are processed and modified to fulfill the following criterion:

Criterion: Let D be a finite set of points in a sub-domain Ω^n of the n-dimensional space R^n. Two points d_i and d_j are connected by a Delaunay edge e if and only if there exists a point $x \in \Omega^n$ which is equally close to d_i and d_j and closer to d_i, d_j than to any other $d_k \in \Omega^n$.

$$e_{Delaunay}(d_i, d_j) \Longleftrightarrow \exists x \quad x \in \Omega^n \ \wedge \ \|x - d_i\| = \|x - d_j\| \ \wedge \ \|x - d_i\| < \|x - d_k\| \ \forall\, k \neq i, j$$

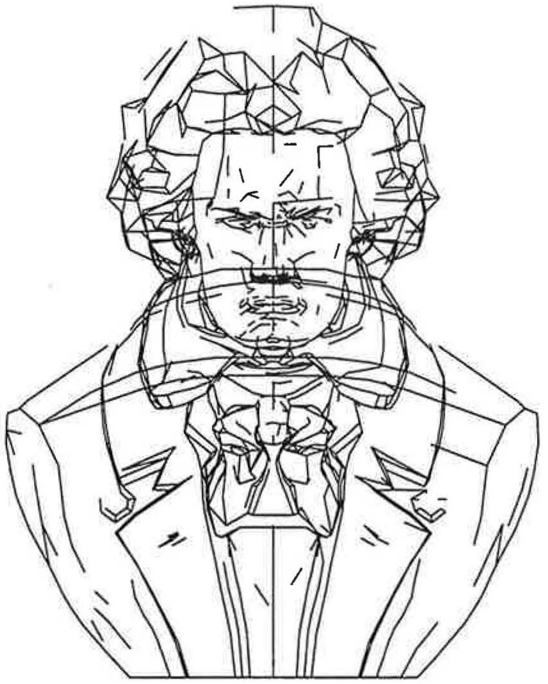

FIGURE 1: BEETHOVEN BUST: EXTRACTION OF VITAL GEOMETRIC INFORMATION

A special approach to adapt the surface representation to this criterion has been implemented:

1. Extraction of edges which hold vital geometric information. Fig. 1 shows such structural edges for the example of a Beethoven statue. A feature edge parameter is used to determine the structural edges. This parameter affects the degree as to how much the original geometry is allowed to be transformed.

2. Intelligent refinement of structural edges by point projection. Fig. 2 shows the necessary cases where *R* denotes "rotational" projections as opposed to *P* for normal projections. The dashed-line circles are used to ensure that the resulting edge has a minimal length. The solid-line circles illustrate additional geometric tests to avoid the generation of edges which are not conform with the above stated criterion. Note that these modifications are performed in three dimensions inspite of the two-dimensional appearance of the figure. In Fig. 3 the result of such a refinement is shown for the example of a complex polygon.

3. Applying local transformations [4] to edges which do not form structural edges. These also include the newly generated edges from the second step.

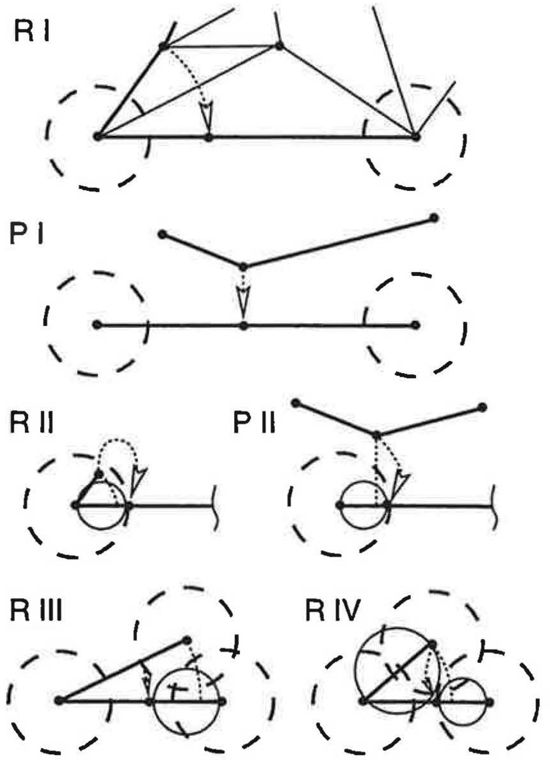

FIGURE 2: INTELLIGENT REFINEMENT OF STRUCTURAL EDGES

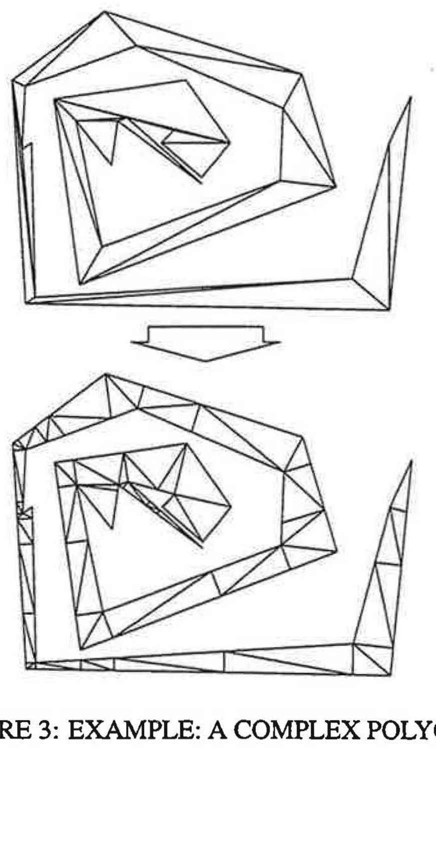

FIGURE 3: EXAMPLE: A COMPLEX POLYGON

MODIFIED ADVANCING FRONT ALGORITHM

The triangles derived from the surface preprocessor represent the boundaries and interfaces and form an oriented initial front. These triangles can be imagined as seeds which are inserted into a queue to "grow" tetrahedra. At the start, the algorithm needs a non-empty queue. It does not require the queue to hold all surface triangles. One triangle per enclosed segment is sufficient. The surface triangulation has two purposes:

1. Provide the initial front for the advancing front algorithm to start with.
2. Provide a border for the advancing front algorithm which cannot be passed.

The triangles of the initial front and all later generated triangles of the advancing front have a well defined orientation depending on the order of their vertices. They "face" the half-space to which their normal vector points. Given a seed triangle (taken from the queue) a tetrahedron is attached which contains a fourth point that has a positive distance to the triangle relative to the normal vector. In other words, the tetrahedron will only be attached to that side of the triangle which faces the half-space to which the normal vector points. In this way, one can distinguish a "front side" and a "back side" of each triangle.

Repeatedly attaching tetrahedra to the front sides of the triangles of the queue, removing them from the queue when they have been processed, and inserting newly generated triangles into the queue leads to a growth process of tetrahedra. Note, that the triangles of the queue form the advancing front at all times. It advances when a new tetrahedron (attached to a triangle which is removed from the queue) results in new triangles which are inserted into the queue. Generally, a created tetrahedron can produce any number between 0 and 3 new triangles. At the start of the tetrahedralization process with the given seed triangles each created tetrahedron will more likely produce 3 new triangles and the queue will increase its size rapidly. Later on, the advancing front will close in and merge with itself or parts of the surface triangulation. A tetrahedron consists of n new triangles, $(3 - n)$ previously generated triangles, and the triangle to which it is attached. The $(3 - n)$ previously generated triangles must have been already inserted into the queue or belong to the initial surface triangulation. They are part of the advancing front. When they are encountered during the creation of a new tetrahedron, they are removed from the queue and the advancing front is stopped. The front cannot pass through the boundary or itself. In these cases the creation of the tetrahedron results in a decrease of the size of the queue. When the queue is empty and all its triangles have been merged, the tetrahedralization process is finished. Fig. 4 summarizes the algorithm flow.

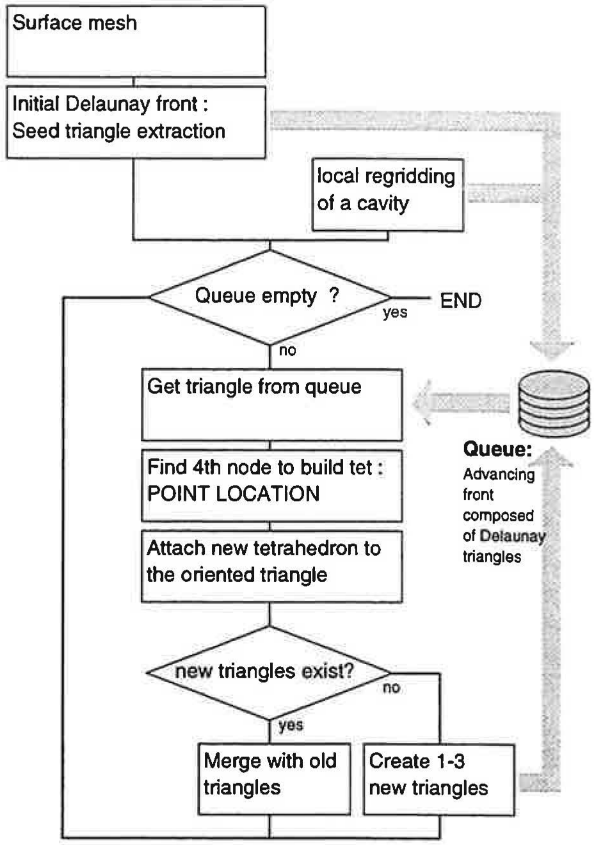

FIGURE 4: ALGORITHM

How the fourth vertex is determined to complete the active base triangle to form a tetrahedron is essential. The elements should fulfill the criterion stated above (optimality in a Delaunay sense). Hence, they are derived in a special way: A sphere can be defined by the active triangle and a radius. The radius will be increased as long as the sphere does not contain other points. In this way the sphere is blown up until its perimeter reaches another point. This point is the seeked fourth vertex of the tetrahedron. The table shows the overall performance on a HP 9000-735/100 workstation.

quantity of ...			CPU time
points	tetrahedra	triangles	(in sec)
103	535	1098	0.2
503	3016	6112	1.4
703	4323	8739	2.1
1003	6268	12635	3.3
1503	9494	19107	5.1
2003	12713	25557	6.8
2503	16076	32300	8.8
3003	19394	38967	11.1
4003	25969	52140	15.6
5003	32691	65605	19.7
10003	65927	132160	46.5
20003	132854	266133	92.0
30003	199613	399756	145.0
40003	266899	534405	207.0

During an optimization loop over all elements additional criteria can be satisfied and the volume decomposition further adapted to specific simulation needs. The well known technique to insert Steiner Points [5] can for instance be easily implemented, because the modified advancing front algorithm offers a very convenient way to apply local modifications. In such a manner the aspect ratio of the elements is improved [6].

1. Mesh update: deletion or insertion of a node, element deformation by moving nodes
2. Checking the Delaunay criterion for all connected elements. Removing non-Delaunay elements.
3. Recursion: Removing non-Delaunay elements which are connected to already removed elements
4. Reprocessing the combination of the cavity and the internal nodes. No surface preprocessing of the cavity surface is required. The modified advancing front algorithm performs the tetrahedralization of the local region in a straightforward manner by queuing the unattached triangles.

Fig. 5 shows a two-dimensional example where the circumcenter of a triangle [5] is inserted and local adaptation is performed.

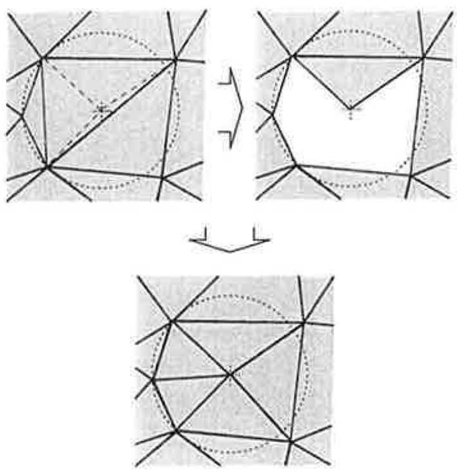

FIGURE 5: STEINER POINT INSERTION FOR A BAD ELEMENT.

APPLICATION EXAMPLE

The presented example shows a typical structure evolving during semiconductor process simulation of an NMOS transistor. This so called $0.18\mu m$ technology including a thin oxide layer ($30nm$) poses a challenge to most existing preprocessors. Only with an efficient and fully flexible approach is it feasible to deal with such structures. Fig. 6, Fig. 7, Fig. 8, and Fig. 9 show the input, the extracted geometry information, and two simulation profiles.

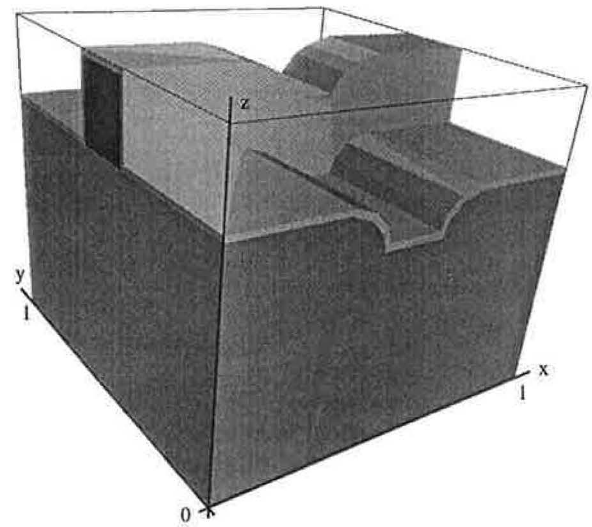

FIGURE 6: INPUT STRUCTURE DESCRIPTION WITH THIN LAYERS

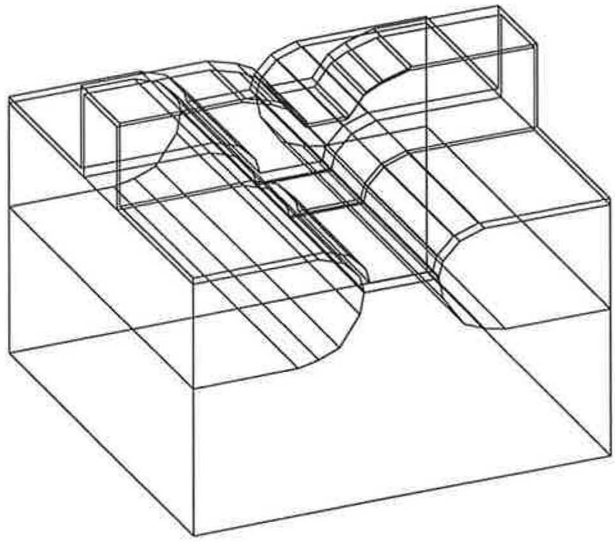

FIGURE 7: STRUCTURAL EDGES

CONCLUSION

We have presented an approach which deals with the complexity typically exposed by semiconductor structures. It has been shown how to optimize the surface representation of the input model and how to derive a volume decomposition satisfying various criteria. The implemented algorithm is ideal for local modifications to adapt to certain simulation needs. No application specific requirements were necessary, hence the method can be employed for a wider range of simulations. Most importantly it can be automated to be embedded into a framework for TCAD applications.

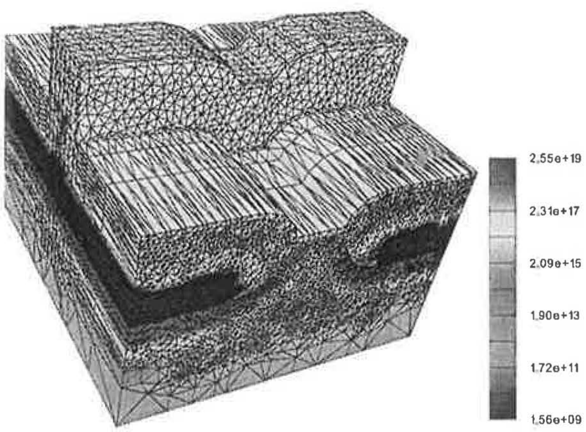

FIGURE 8: PREPROCESSED VOLUME REPRESENTATION WITH BOR IMPLANTATION PROFILE

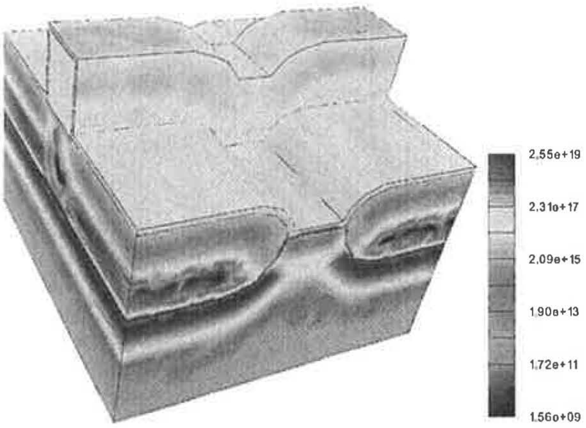

FIGURE 9: BOR PROFILE AFTER DIFFUSION (30 MINUTES AT $875°C$)

REFERENCES

[1] M. Bern and D. Eppstein, Mesh Generation and Optimal Triangulation, *Computing in Euclidean Geometry*, pp. 201–204, 1992.

[2] A. Okabe, B. Boots, and K. Sugihara, *Spatial Tessellations — Concepts and Applications of Voronoi Diagrams* (Wiley, 1992)

[3] P.J. Frey, H. Borouchaki, and P.L.George, Delaunay Tetrahedralization Using an Advancing-Front Approach, *Proceedings 5th International Meshing Roundtable*, Sandia National Labs, Pittsburgh, 1996.

[4] B. Joe, Construction of Three-dimensional Improved-Quality Triangulations Using Local Transformations, *SIAM: J. Sci. Comput.*, 16(6), pp. 1292–1307, 1995.

[5] L.P. Chew, Guaranteed-quality triangular meshes, *Tech. Rep. TR-89-983*, Cornell University, 1989.

[6] J. Ruppert, A Delaunay Refinement Algorithm for Quality 2-Dimensional Mesh Generation, *Journal of Algorithms*, 18, pp. 548–585, 1995.

BIOGRAPHY

Peter Fleischmann was born in Kabul, Afghanistan, in 1969. He studied electrical engineering at the Technical University of Vienna, where he received the degree of 'Diplomingenieur' in 1994. He joined the 'Institut für Mikroelektronik' in December 1994. In December 1997 he was with NEC in Sagamihara, Japan. He is currently working for his doctoral degree. His research interests include mesh generation as well as algorithms and data structures in computational geometry.