3D Mesh Generation for the Results of Anisotropic Etch Simulation

Elena V. Zudilova, Maxim O Borisov

Integration Technologies Department
Corning Scientific Center
4 Birzhevaya Linia, St. Petersburg 199034, RUSSIA
tel.: +7-812-32292060 fax: +7-812-3292061
e-mail: {ZudilovaEV; BorisovMO}@corning.com
http://www.corning.com

Abstract. The paper is devoted to the development of 3D mesh generator that provides the interface between etch simulation tool of IntelliSuite™ CAD for MEMS™ and its analyses compounds. Paper gives a brief introduction to IntelliSuite and its anisotropic etch simulator. The rest of the paper is devoted to the algorithm of 3D mesh generation based on special requirements to Finite Element Mesh. This algorithm can be divided into two main stages: manual 2D meshing of the upper surface and automatic 3D mesh generation for simplified object obtained from the previous stage. Paper contains the detailed description of both stages of the algorithm.

1. Introduction

The paper is devoted to the development of mesh generation algorithm for 3D object which geometry is defined by the unstructured set of triangular and quadrangular polygons that form its surface.

The algorithm has been developed for building the interface between anisotropic etch simulator of IntelliSuite™ CAD for MEMS™ and its performance analysis components. This interface will allow to use 3D model of MEMS (Micro Electro Mechanical Systems) generated during etch simulation process for further mechanical, electrostatic, electromagnetic and other types of analyses provided by IntelliSuite, unfortunately, not available today.

To solve this task a stand-alone application has been developing. It permits a user to simplify manually the shape of initial 3D geometry of the object by smoothing slopes of etching pits via 2D surface meshing. Then this simplified 3D object is processed and meshed automatically by 3D mesh generator.

The introduction to IntelliSuite and its mesh requirements are given in section 2 of the paper. Section 3 is devoted to the first stage of the algorithm - manual 2D surface meshing. Section 4 contains the description of the second stage - automatic 3D mesh generation.

P.M.A. Sloot et al. (Eds.): ICCS 2002, LNCS 2329, pp. 316–323, 2002.

2. IntelliSuite Anisotropic Etch Simulation

IntelliSuite CAD for MEMS provides the ability to simulate with high accuracy different classes of MEMS devices induced mechanically, electrostatically and electromagnetically and then to obtain the graphical presentation of the appearance of each simulated device. It is an integrated software complex which assists designers in optimizing MEMS devices by providing them access to manufacturing databases and by allowing them to model the entire device manufacturing sequence, to simulate behavior and to see obtained results visually without having to enter a manufacturing facility. [1, 4]

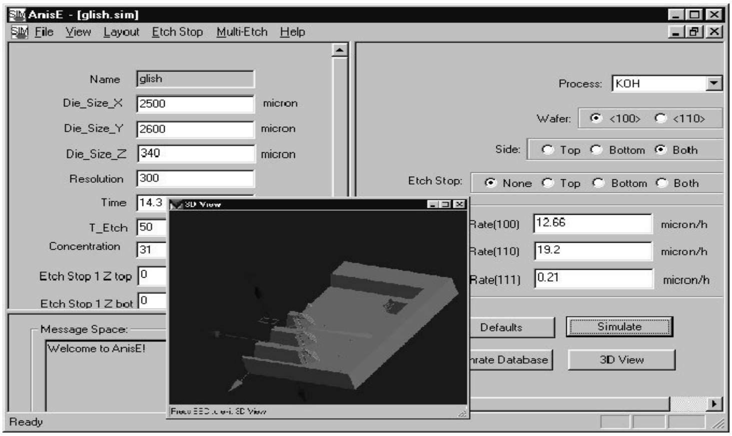

Fig. 1 IntelliSuite example of anisotropic etch simulation results

Etch simulation tool of IntelliSuite AnisE® permits to generate 3D model for anisotropic etching of silicon. With AnisE user can layout a microstructure, view 3D representation, access information about the etch rates of different etchants and simulate automatically the etching under different time, temperature, and concentration parameters [6]. Fig. 1 contains a screen-shot of AnisE GUI with a sample of 3D visualization.

3D geometry of MEMS device generated by AnisE can not be processed further by performance analyses components of IntelliSuite. It occurred because the geometry of MEMS device generated by AnisE, as a set of triangles and quadrangles defining its surface, does not satisfy Finite Element Mesh [5] requirements on which IntelliSuite is based.

A stand-alone application has been developing for building interface between AnisE and performance analyses of IntelliSuite. This component generates solid mesh from initial MEMS device geometry of AnisE in accordance with FEM requirements of IntelliSuite. These requirements are as follows [4, 7]:

- Mesh elements are hexahedrons;
- All edges are uninterrupted;
- Limits for edges ratio (1:1; 1:10);

- Lower and higher angle bounds ($\alpha_{min} = 30°$; $\alpha_{max} = 120°$);
- Optimum number of finite meshed elements.

All mesh requirements listed above are determined by ABAQUS on which IntelliSuite simulation routines are based.

The algorithm of 3D mesh generation is divided into two main stages - manual 2D surface meshing and automatic solid meshing.

First, the upper surface of the object is to be meshed by user. As a result, the new surface formed by quadrilateral polygons that satisfy angle and ratio requirements is generated. User also has the possibility to simplify the geometry of initial object by smoothing slopes or deleting insignificant pits. So at the end of the following stage a new simplified 3D object is generated. Its geometry is defined by the meshed upper surface, lower and front facets. Then this new geometry is processed by automatic 3D mesh generator and final solid mesh that satisfies all the requirements above is generated. The detailed description of each stage of the algorithm is provided in the next two sections of the paper.

3. Manual Surface Meshing

2D meshing of the upper surface of the object is conducted manually. Fig.2 represents the screen-shot of the graphical editor intended for this purpose.

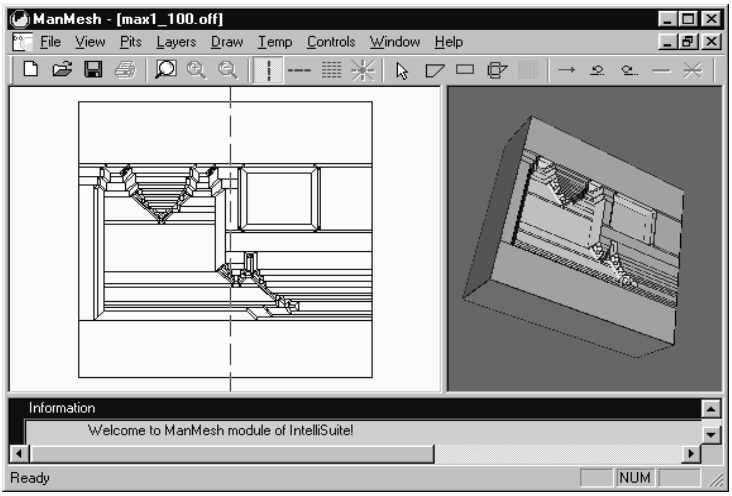

Fig. 2 Current GUI of 2D layer dependent editor (Win-platform: MFC, Vtk 3.1.2)

2D surface meshing and smoothing slopes of the object is conducted through 2D layer dependent editor [9] - the upper left zone on Fig. 2. Any projection of the object can be selected to be active for meshing [1, 3]. User can work with a concrete etching pit or with the object as a whole. At the beginning of surface meshing user detects the significant layers on the vertical or horizontal section of the object (Fig. 3).

User can draw mesh elements - generic quadrilateral polygons and rectangles. As soon as a new mesh element is added, the automatic checker runs. This checker verifies overlapping of elements, whether angles and edges ratios satisfy mesh requirements or not. If the obtained quadrangle satisfies all mesh requirements, it becomes colored as it is shown on Fig. 4.

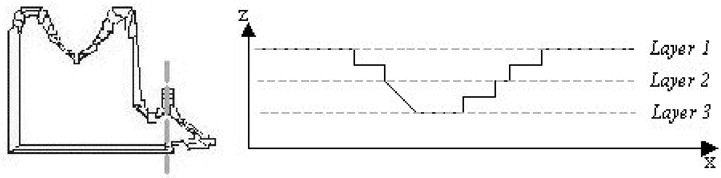

Fig. 3 Sample of definition of significant layers

The upper right zone on Fig. 2 contains interactive 3D viewer that represents geometry of MEMS device. As soon as any type of editing is conducted, viewer updates its screen. Special routine for etching pits recognition has been developed. It gives a user a possibility to delete insignificant pits from the object. This routine scans the active surface of the object and selects all connected faces that form a separate pit. As a result etching pits on 3D view are colored differently.

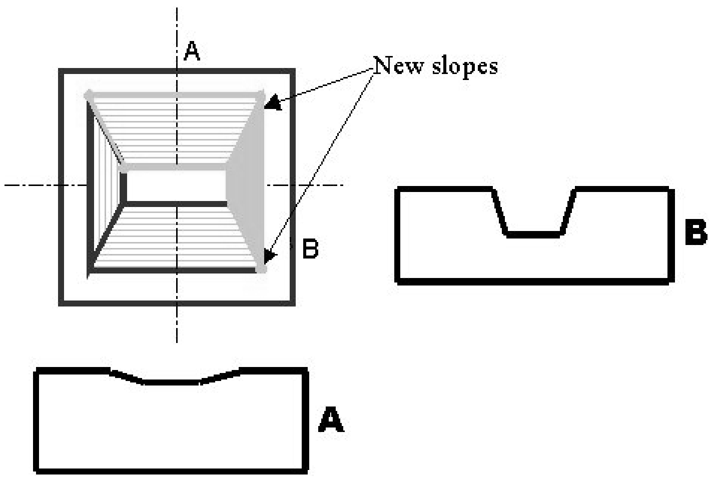

Fig. 4 Sample of smoothing slopes of a simple etching pit

The main goal of working with 2D layout editor is to get a new 3D object which upper surface will be meshed by quadrilateral polygons, convex and unstructured.

Special routine checks whether all final edges are uninterrupted or not. If it is true then the manual part of meshing is completed.

4. Automatic Generation of Solid Mesh

The next part of the algorithm is aimed to the generation of 3D mesh by processing data obtained from the previous stage. When the manual part of the algorithm is completed a new object is generated. Its surface is completely defined by a set of quadrilateral polygons. The next task is to mesh 3D object entirely on the base of obtained 2D surface mesh.

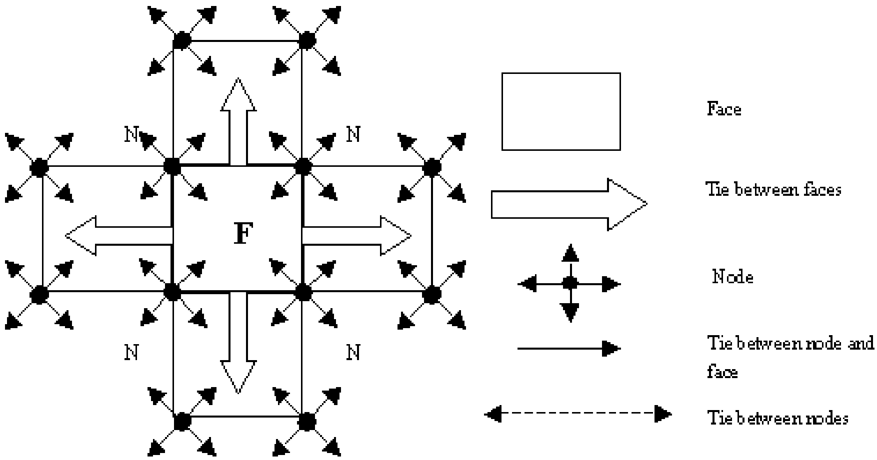

Fig.5 Data structure: faces, nodes and ties

There are lots of methods of 3D mesh generation, e.g., tetrahedral mesh [8], 20-node brick mesh [2], etc., that, unfortunately, do not satisfy in our case. The final solid model that is necessary to obtain has to be formed by hexahedrons satisfying several mesh requirements. Internal angles of each hexahedron's faces are between α_{min} and α_{max}, all faces of hexahedron have to be plane and some others (see section 2). Moreover, several extra conditions exist. There is no undercutting [6]. The geometry is defined by the meshed upper surface of the object, i.e. 2D surface mesh is to be transformed to 3D solid mesh [3].

All said above caused us to develop a new mesh algorithm. A common algorithm of solid meshing for AnisE results after 2D surface meshing is as follows.

(1) Represent the object surface as a set of clusters, where *cluster* is a set of faces, nodes and ties between them (see Fig. 5); so each face knows its neighbors and their ties (sizes, angles, etc.).

(2) Build the adjacency matrix for all faces, so that each face knows the shortest way to space, where *space* is the empty region that surrounds the surface of an object.

CYCLE (until the whole object is meshed):

(3) Surface analysis - finding the cluster with the slope of face plane less than α_{min} or the cluster that completely defines a hexahedron (see Fig. 6);

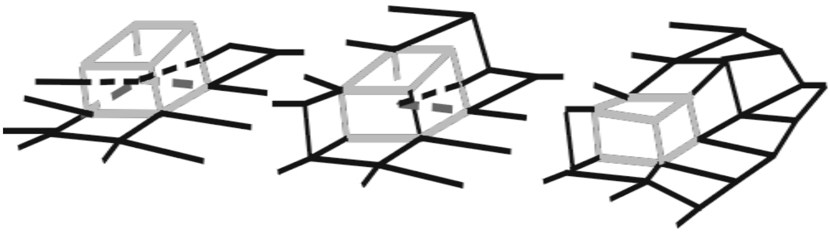

Fig. 6 Samples of clusters defining hexahedrons completely

(4) If there are no such faces, then cut the object vertically by planes formed from surface quadrangles.

(5) Build a hexahedron from the current cluster if this cluster completely defines hexahedron (Fig. 6):
   - If obtained hexahedron satisfies mesh requirements, then add missing faces if necessary and remove this hexahedron from the object with replacing obtained emptiness by appropriate faces;
   - If obtained hexahedron does not satisfy requirements, then divide a cluster into several ones trying to locate all the changes inside.

(6) If cluster does not define hexahedron completely, then build missing faces with tendency to horizontal planes.

(7) Return to step 3.

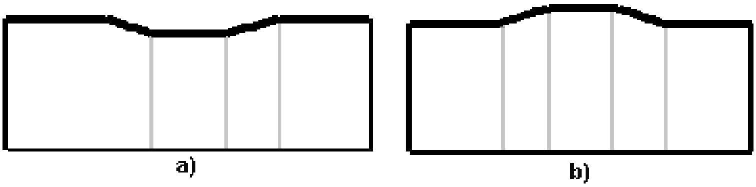

Fig. 7 Sample of 3D meshing: faces are horizontal or nearly horizontal

After data restructuring surface can be analyzed as a set of clusters. One of the approaches of how it can be done is shown on Fig. 7. If all angles between each face and the vertical axis Z is greater than α_{min}, then the object should be cut vertically by planes defined by 2D surface mesh elements (quadrangles) obtained from the previous stage.

If several faces of a cluster are "too vertical" (the angle between the face and axis Z is less than α_{min}), then it can be tried to build a hexahedron from this cluster.

The definition of a hexahedron depends on the number of ties in cluster and spatial location of faces. A cluster can completely define a hexahedron, e.g. if a cluster has 8 nodes and only 24 node-to-node ties.

In the case when a cluster does not completely define a hexahedron and several faces are not "nearly horizontal" (see Fig. 7), one another approach can be used. We can generate all missing parts, faces or edges, from 2D surface mesh elements with tendency to optimizing (construction of plane intersections, point-wise construction, etc.), trying to locate them so that angles being formed with axis Z were greater than α_{min}. After that this completely defined polygon has to be cut from the object and the emptiness obtained should be filled with appropriate faces and nodes as it is shown on Fig. 8.

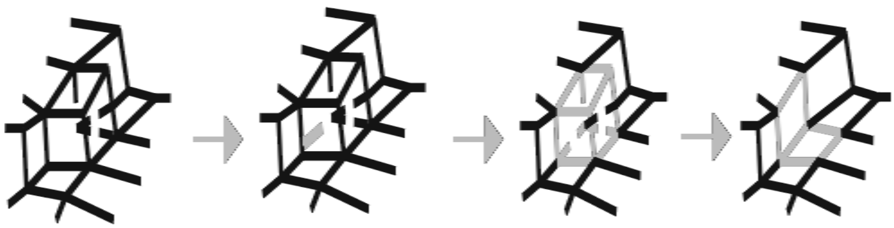

Fig. 8 Finding a cluster that defines a hexahedrons and removing it

Some clusters may completely define elements but obtained hexahedrons do not satisfy requirements (facets are not plane or angles are not correct). It can happen because of the characteristics of faces and their spatial location. Thus this completely defined cluster should be divided into several clusters that form correct hexahedrons. It is desirable that all changes should be localized inside initial cluster. New generated hexahedrons should be removed from the object as it was described above.

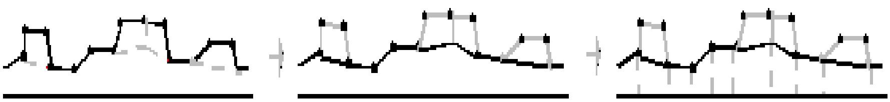

Fig. 9 Making surface "nearly horizontal"

Then we find projected clusters or clusters having “too vertical” faces (see Fig. 9), construct hexahedrons and cut them off till we get "nearly horizontal" surface or mesh the object as a whole.

5. Conclusion

The uniqueness of the algorithm described in the paper consists in the combination of manual simplification of 3D geometry with automatic mesh generation. It will give a user the ability to control the process of generating mesh. The following algorithm being implemented now is the first step of the long work aimed to the development of the interface between anisotropic etch simulator of IntelliSuite CAD for MEMS and its performance analyses components. The final goal of this work is to develop automatic 3D Finite Element Mesh generator that can be corrected by user if necessary [9].

These algorithms both elaborated and projected can be also used for solution of various mesh generation tasks when 3D object is defined by a set of triangles and/or quadrangles and mesh requirements are similar to those listed above.

Adding to IntelliSuite a new component based on the approach represented in the paper will permit to serve better users needs. IntelliSuite users will be able to go on processing anisotropic etch simulation results via electrostatic, mechanic, electromagnetic and other types of analyses within a uniform design environment that they can not do today.

References

1. Bogdanov A.V., Stankova E.N., Zudilova E.V.: Visualization Environment for 3D Modeling of Numerical Simulation Results, Proceedings of the 1st SGI Users' Conference, pp. 487-494 (2000).
2. Dhondt G.D.: Unstructured 20-node brick element meshing, Computer Aided Design N 33, pp. 233-249 (2001).
3. Farin G.: Curves and Surfaces for Computer Aided Geometric Design: A Practical Guide, Academic Press, Boston (1993).
4. He Y., Marchetti J., Maseeh F.: MEMS Computer-aided Design. Proc. of the 1997 European Design &Test Conference and Exhibition Microfabrication (1997).
5. Joun M.S., Lee M.C.: Quadrilateral Finite-Element Generation for Mesh Quality Control for Metal Forming Simulation, Int. J. Num. Meth. Engng., 40(21), pp.4059-4075 (1997).
6. Marchetti J., He Y., Than O., Akkaraju S.: Process Development for Bulk Silicon Etching using Cellular Automata Simulation Techniques. Proc. of SPIE's 1998 Symposium on Micromachining and Microfabrication, Micromachined Devices and Components, Santa Clara, CA (1998).
7. Maza S., Noel F., Leon J.C.: Generation of quadrilateral meshes on free-form surfaces, Computers and Structures, 71, pp.505-524 (1999).
8. O'Rourke J.: Computational Geometry in C (second edition), Cambridge University Press, 377 p. (1998)
9. Upson C., Faulhaber J., Kamins D.: The Application Visualization System: A Computational Environment for Scientific Visualization, IEE Computer Graphics and Applications. 9(4), pp. 30-42 (1989).