CFD-Micromesh: A Fast Geometric Modeling and Mesh Generation Tool for 3D Microsystem Simulations

Z. Tan, M. Furmanczyk, M. Turowski, and A. Przekwas

CFD Research Corp., 215 Wynn Drive, Huntsville, Alabama, 35805, USA
[zt|mf|mt|ajp]@cfdrc.com

ABSTRACT

In this work, a new fully automated geometrical modeling and meshing tool is described. It imports standard layout formats (CIF, GDSII, DXF), images (GIF, JPG), and 3D boundary representations (STL). A 3D model is then generated by simulating 3D operations (etching, deposition, insertion, etc.) specified by the process data or the user. A 3D finite element mesh with tagged boundary and volume conditions is then automatically created. The automatic generation of 3D model and mesh takes typically a couple of minutes on a current PC machine. The paper will present the geometry/meshing engines, user interfaces, and will demonstrate them on a range of microsystem applications.

Keywords: TCAD, mesh generation, 3D modeling.

1 INTRODUCTION

Design of Microelectromechanical Systems (MEMS) is not a straightforward task as there is no fully integrated, robust, easy to use CAD tool available. Design software vendors face major challenges integrating range of physical disciplines such as: fluidics, mechanics, heat transfer, electostatics, electronics, electomagnetics, controls, and other into a single design environment. The first challenge facing the microdevice designer is to setup the problem in terms of geometry, mesh, material properties, boundary and volume conditions, and other parameters needed to simulate the device with high fidelity field solvers. Since geometrical configurations of microsystems are often very complex, typically the grid generation task is performed manually with mechanical CAD tools, which is a tedious and time-consuming effort. Initial design of MEMS is routinely performed with the ECAD tools and stored in the form of layouts (in CIF or GDSII format) and process specifications and there is no straightforward way to transfer such data into electomechanical CAD tools. There is a great need for fully automated mesh generation from 2D layouts and fabrication process specifications.

In this work, a new fully automated geometrical modeling and meshing tool is described. It can import standard layout formats (CIF, GDSII, DXF), or even images (GIF, JPG), and generate a 3D model using operations specified by the process data or the user. A 3D finite element mesh with tagged boundary and volume conditions is then automatically generated. The automatic generation of 3D model and mesh takes typically a couple of minutes on a current PC machine. The paper will present the geometry/meshing engines, user interfaces, and will demonstrate them on a range of microsystem applications.

2 VOXEL REPRESENTAGTION AND MODELING

The voxel model. In this work, the solid model is represented by voxels. A one-byte space is assigned to each voxel. This allows up to 256 materials in a single model.

In most previous works, a regular $N_x \times N_y \times N_z$ was used to represent the voxel array [1, 2]. This representation is simple but memory consuming. Octree and "segment walls" have been used [3, 4], but it is slow to access an element. Access time is very important in our case because the model is non-static (i.e. we have to simulate the etching and deposition operations to get the final model).

In this work, we store the voxel data set as an $N_x \times N_y$ array of pointers. Each pointer points to a 1D voxel column of N_z height. They columns are shared whenever possible. The sharing is automatically managed by reference counting and insulated from the outside of the data structure; so the interface of the voxel data structure still behaves like a regular 3D array. This strategy is appropriate for many TCAD applications where many columns are the same.

Operations on the voxel data set. 3D models are generated starting from 2D masks and other 3D representations. They include: 1) deposition, 2) etching, 3) diffusion (which replace a material of given thickness by a new material), 4) dropping (which drops a 3D object from $z = \infty$ till it touches a solid voxel), 5) insertion of a 3D solid, 6) removing of a 3D solid, and 7) replacing a material inside a 3D mask by a new material. Note that our 3D operations can be used to emulate the standard CSG operations.

2D masks. The 2D masks are specified in either vector format (polygons, disks, rectangles, etc.) or image format (GIF, JPEG, BMP, PPM, etc.). They can be imported directly from a layout file (CIF, GDSII, DXF). By also specifying the processing technology (e.g. MUMPS for MEMS applications), the model is uniquely defined.

Etching/deposition/diffusion distance function. The corner shapes of the operations are controlled by a two-parameter distance function $d(\mathbf{u}, \mathbf{v}; \beta, n) \equiv \{[(u_x - v_x)/\beta]^n + [(u_y - v_y)/\beta]^n + [u_z - v_z]^n\}^{1/n}$. When $n = 2$ and $\beta = 1$, the usual Euclidean distance is obtained, which yields rounded corners. When $n = \infty$, we have the Manhattan distance which gives square corners. The parameter β controls the ratio of horizontal growth and vertical growth.

Efficient simulation of etching/deposition/diffusion operations. In a naïve simulation, all voxels within the thickness under the specified distance function are visited for possible replacement. The cost is $O(At^3)$ and very expensive when the thickness is much larger than one voxel, where A is the initial wet area and t is the thickness. We can exploit the spatial coherence to reduce the operation time. For example, if a wet face element has a neighboring wet face element on the same plane, we don't need to check the voxels on the side of its neighbor. This strategy reduce the average time to $O(At)$, which is optimal.

Conversion of 3D data. Surface triangulations in STL format can be imported into the modeler. An extension of the standard polygon scanline conversion algorithm is implemented. To reduce the searching time for candidate triangles intersecting the scanline, a sweep plane orthogonal to the scanline is maintained. Some standard primitives (ellipsoid, cylinders and bricks) are directly converted into voxels. Extrusions, heightmaps and voxel data are also converted.

Visualization. The model is visualized through ray-traced images. On input, the user specifies the material, light and camera properties. The user can also control which materials (including the air) to visualize.

3 MESH GENERATION

The meshing module generates finite element grid with prismatic or hexahedral elements. The voxel data are projected onto the xy-plane to obtain a 2D image. A 2D triangular grid is then generated. The triangles are then extruded to get volumetric grid. So we can decompose our task into the following sub-problems: 1) given a 2D image, generate a high-quality triangular mesh, 2) extrude, and 3) export the grid in proper format.

Triangulation of 2D image. The pixels on material boundaries are first identified. A quadtree mesh is then generated and triangulated without altering the material boundaries. The triangles are then modified to improve the qualities. The material boundaries are smoothed (to remove the jiggles due to finite voxel size) during the quality improving process. The maximum boundary movement in the normal direction is limited within a user specified number. The mesh quality improvement process consists of lower level operations such as edge swapping and vertex smoothing.

Mesh size control. The user can control the mesh size distribution by providing a bitmap. The bitmap is then smoothed inside the mesher to ensure smooth transitions.

Extrusions. The z-planes are automatically located by detecting the largest material changes between neighboring planes. The user can also specify the planes manually. When requested, the extruded prisms can be further processed to obtain grids of mixed prismatic-hexahedral, pure tetrahedral, or pure hexahedral type.

Grid export. The grids are exported into DTF (Data Transfer Facility) format. Care was taken to ensure that the boundary conditions, volume conditions, and interfaces are properly tagged. This eases user in setting up physical simulations.

4 USER INTERFACE

The CFD-Micromesh graphical editor has been designed for structures built from separate layers. A two-dimensional (2D) layout and information about the 3rd (z-) dimension define each layer. The editor space for a project is divided into two parts: the left part is used for viewing and editing 2D layouts of masks in particular layers, and the right part allows changing layer properties and their order.

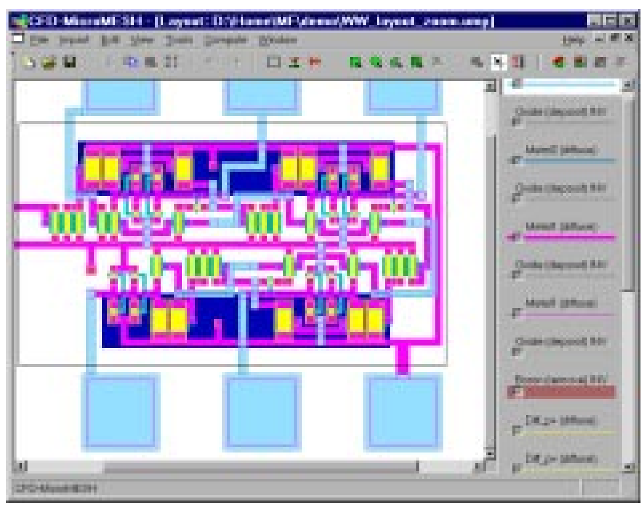


Figure 1: User Interface Main Window.

A layer contains information about material, type of operation, and relative parameters. The program displays layers in the order of operations necessary to build a 3D model. The layer displayed at the bottom of the list will be inserted into the model first; the others follow from the bottom to the top of the list. This is similar to the layers placed (deposited or etched), one on the top of another, as a result of a manufacturing process in microelectronics. Possible operations are: deposit, diffuse, etch, remove, insert 3D, drop 3D, replace 3D, and remove 3D. It is possible to copy layers, or individual elements, between different projects.

The layout editor displays two-dimensional top view of the project. The window displays figures of the currently selected layer and all the layers positioned below the selected one in Layer List.

The triangular mesh of the projected 2D image can be distributed uniformly on the modeled box ("project area"), or according to chosen criteria. There is a special type of layers used for mesh-density control. They define relative size of triangles in selected areas.

The 2D layout of each layer can be edited by the user. Layout can be built from basic geometrical primitives and bitmaps such as GIF or BMP images. The editor makes possible to move, resize and copy/paste figure or a group of figures. There are also zooming and unlimited undo/redo operations.

A 2D layout can also be imported from two popular electronic design formats GDSII and CIF. It is possible to define for given technology the layer types and properties and to use this definition to import data for several projects.

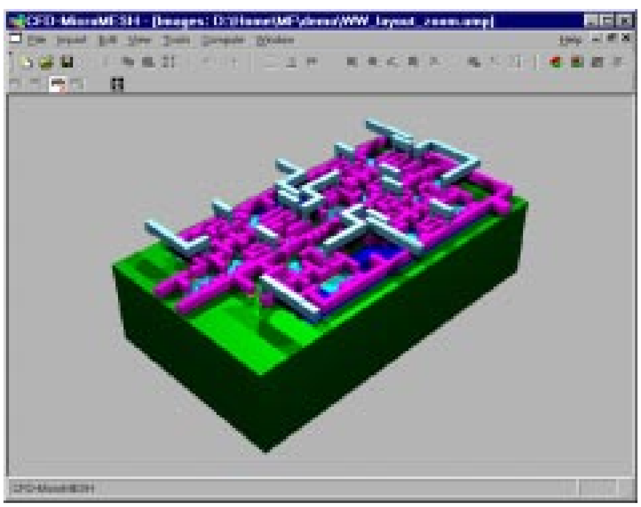


Figure 2: Generated 3D solid model.

The solid modeling and mesh generation programs are integrated with graphical editor. Therefore, it is possible to perform all building steps from the graphical environment and view the created 3D solid model (Fig. 2). The final computational mesh is written to DTF format used by CFD-ACE+ simulation program (`http://www.cfdrc.com`).

5 EXAMPLES APPLICATIONS

The CFD-Micromesh software has been tested and applied to many practical problems ranging from microelectronics to automobile industry. In this paper, only applications in microelectronics are demonstrated.

In Fig. 3 the layout (from GIF), ray-traced solid and the prismatic grid are shown. The air surrounding the model is removed.

Figure 4 shows the automatic modeling and meshing of VLSI interconnects. The 3D layout in the GUI is also shown. In this case, many prisms have been merged into hexahedra.

Figure 5 shows an application to a DC converter. The layouts of lead frame and other parts are from GIF files. The computed temperature distribution is also shown.

6 CONCLUSION

In this paper, a new fully automated geometrical modeling and meshing tool has been described and demonstrated on several examples. The tool imports 2D layouts (CIF, GDSII, DXF, GIF, JPG, …) as well as 3D boundary representations. A 3D model is then generated by simulating 3D operations (etching, deposition, insertion, etc.) specified by the process data or the user. The solid is ray-traced to provide visual feedback. The 3D finite element mesh generation is also automatically generated. The mesh quality is high, and the user can have full control of the mesh size distribution. The whole process is automatic, and the user controls it at high level.

The graphic user interface enables the user to edit and setup control parameters easily. It also integrates the modeling, meshing, visualization, file conversion and editing into a single environment.

Work is in progress on generating fully 3D tetrahedral mesh. Another improvement in progress is to use higher order voxels similar to level set functions. Preliminary results show that the model accuracy can be significantly improved.

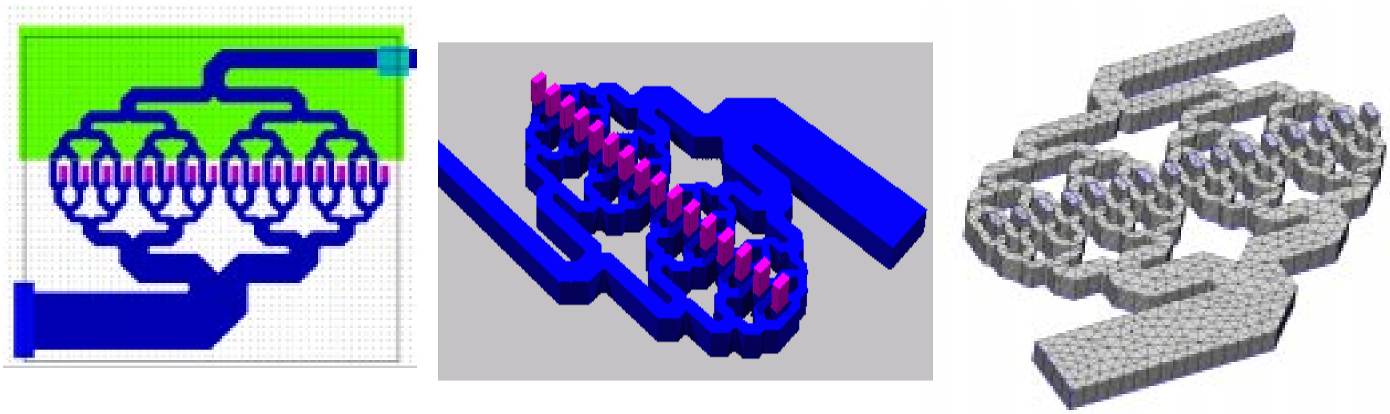
REFERENCES

[1] Kaufman, A., Cohen, D., and Yagel, R., "Volume Graphics," *IEEE Computer*, Vol. 26, 51-64, 1993.

[2] Pyka W., Martins R., Selberherr S., "Efficient Algorithms for Three-Dimensional Etching and Deposition Simulation," *Proc. SISPAD 98 Conf.*, 16-19, 1998.

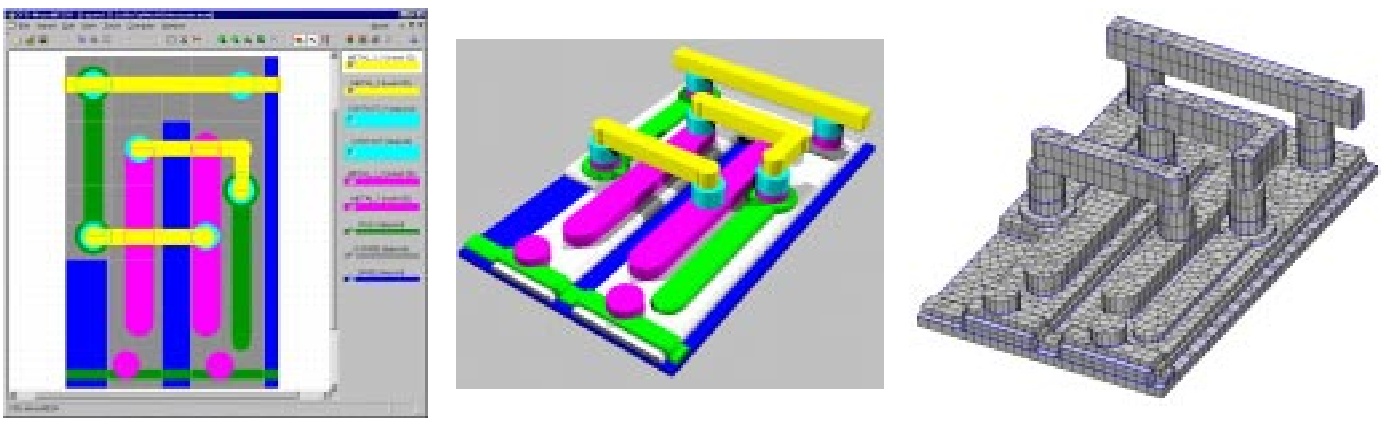
[3] Meagher, D.J., "Efficient Synthetic Image Generation of Arbitrary 3-D Objects," *Proc. IEEE Computer Soc. Conf. On Pattern Recognition and Image Processing,* 1982.

[4] Shareef, N. and Yagel, R., "Rapid Previewing via Volume-based Solid Modeling," *3rd Symp. Solid Modeling and Applications, SOLID MODELING `95*, 281-292, 1995.



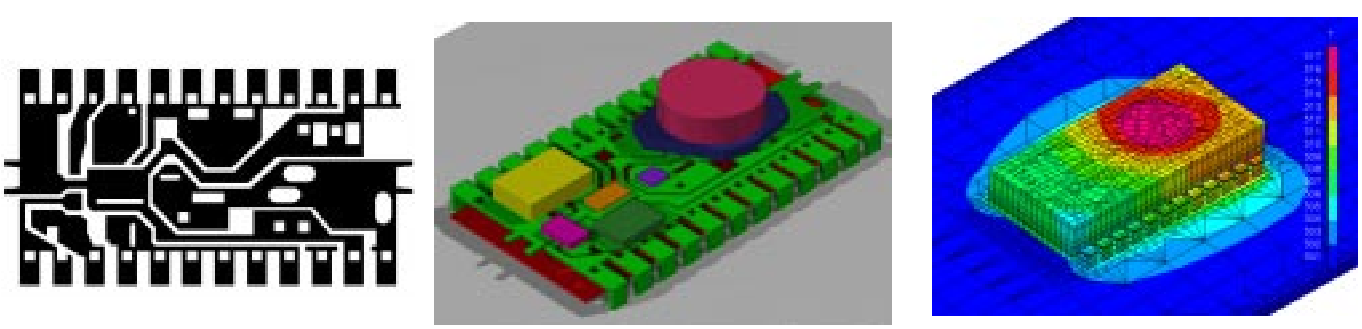
Layout **3D Solid Model** **Computational Mesh**

Fig. 3. Automatic 3D model and mesh generation for fluidic micromixer.



Layout in CFD-Micromesh GUI **3D Solid Model** **3D Mesh**

Fig. 4. Three-dimensional modeling of IC's and electronic interconnects.



Bitmap Image (GIF) **3D Solid Model** **Computed Temperature Distribution**

Fig. 5. Automatic 3D model and mesh generation of a DC/DC converter