

SANDIA REPORT

SAND2007-7755
Unlimited Release
Printed November 2007

Automated Mask Creation From A 3D Model Using Faethm

Richard L Schiek, Rodney C Schmidt
Department 1437: Electrical and Microsystem Modeling
PO Box 5800
Mail Stop 0316
Albuquerque, NM 87185-0316

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under Contract DE-AC04-94-AL85000.

Approved for public release; further dissemination unlimited.



Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government, nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, make any warranty, express or implied, or assume any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represent that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof, or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: <http://www.osti.gov/bridge>

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.fedworld.gov
Online ordering: <http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online>



Automated Mask Creation From A 3D Model Using Faethm

Richard L. Schiek & Rodney C. Schmidt

Abstract

We have developed and implemented a method which given a three-dimensional object can infer from topology the two-dimensional masks needed to produce that object with surface micro-machining. The masks produced by this design tool can be generic, process independent masks, or if given process constraints, specific for a target process. This design tool calculates the two-dimensional mask set required to produce a given three-dimensional model by investigating the vertical topology of the model.

Acknowledgment

The authors would like to thank the Computational Science Research Foundation for their support for this work. We would also like to thank Victor Yarberry for his technical guidance and advice.

Contents

Preface	8
1 Device Design and Topology	9
Topological Approach	10
2 Cross Sections to Masks	13
Topology Graph Analysis	14
Creating Production Masks	15
3 Algorithm and Implementation	17
Example	18
Implementation	20
Use	20
Performance	23
4 Conclusion	27
References	28

List of Figures

1.1	Surface Micromachining of a simple part. (a) Target polysilicon part on a silicon substrate. (b) A sacrificial oxide layer is deposited and a mask is placed. (c) The oxide is etched and the mask removed. (d) Polysilicon is deposited and a mask is placed. (e) Extraneous polysilicon is removed and the mask is removed. (f) The sacrificial silicon oxide is removed.	10
3.1	Locating the unique cross sections and building the topology tree for a hub which holds a gear in place.	18
3.2	This gear has five stubs or stand-offs on its lower face. Since they originate from the same topology node and end at the same location, these five, redundant nodes can be combined into one.	19
3.3	Analyzing the topology tree allows one to locate masks.	20
3.4	Graphical user input form for using Faethm.	22
3.5	Typical program run times for models of increasing complexity (higher vertex count for larger more complex models).	25
3.6	3D model rendering artifact's such as the long, thin <i>stringer</i> shown here can cause errors in faethm's mask calculations.	26

List of Tables

3.1	Models from Sandia's Standard Parts Library which were used to test Faethm. All models were <i>merged</i> prior to running through faethm. This combines all bodies that are in contact and of the same type of material into single bodies.	24
-----	---	----

Preface

We have developed and implemented a method which given a three-dimensional object can infer from topology the two-dimensional masks needed to produce that object with surface micro-machining. The masks produced by this design tool can be generic, process independent masks, or if given process constraints, specific for a target process. This design tool calculates the two-dimensional mask set required to produce a given three-dimensional model by investigating the vertical topology to the model. The 3D model is first separated into bodies that are non-intersecting, made from different materials or only linked through a ground plane. Next, for each body unique horizontal cross sections are located and arranged into a tree based on their topological relationship. A branch-wise search of the tree uncovers locations where deposition boundaries must lie and identifies candidate masks creating a generic mask set for the 3D model. Finally, in the last step specific process requirements are considered that may constrain the generic mask set. Constraints can include the thickness or number of deposition layers, specific ordering of masks as required by a process and type of material used in a given layer. Candidate masks are reconciled with the process constraints through a constrained optimization.

Chapter 1

Device Design and Topology

Designing a device for production by silicon micro-machining is very different from macro-scale mechanical design. In the macro-scale it is often sufficient for a designer to create a 3D model of their device, which a design program then translates into the tool paths needed for production. For a silicon micro-machined device however, the designer must create a set of process specific, lithographic masks needed to fabricate the device. Creating such masks is similar to requiring the macro-scale designer to design the tools needed to fabricate their product as well as the product itself. Because masks are dependent on the process in which they are used and can have complex dependency interactions within a production system, creation of the masks is a significant challenge to innovative device design and the manufacture of a device on multiple processes. Thus it is desirable to develop a tool for translating a designer's 3D model of a product directly into the masks needed to produce their product.

Earlier efforts on this problem have leveraged existing technology in process simulators, i.e. programs which, when supplied with a mask set for a given process, can simulate fabrication from those masks. Typically, this approach uses a trial mask set to produce a 3D object that is then compared to the desired object. Differences between the two objects are used to alter the trial mask set and then the process is repeated until a mask set is found which correctly produces the desired part. When coupled with a sophisticated optimization scheme, this approach works well for anisotropic etching processes. [1] Being computationally intensive however, optimization trial masks through a process simulator has yet to produce masks for complex, multi-layer surface micro-machined devices. Another approach starts from a 3D model that is annotated. This annotation describes when in the process each section of the model will be made and from which section a mask is derived. [2] More recently progress has been made on a geometric approach where a 3D model is interrogated for features that can be made via surface micro-machining, and a mask set is derived for these features. [3] While promising, these techniques cannot produce masks for specific processes nor handle isotropic etching processes such as wet etches.

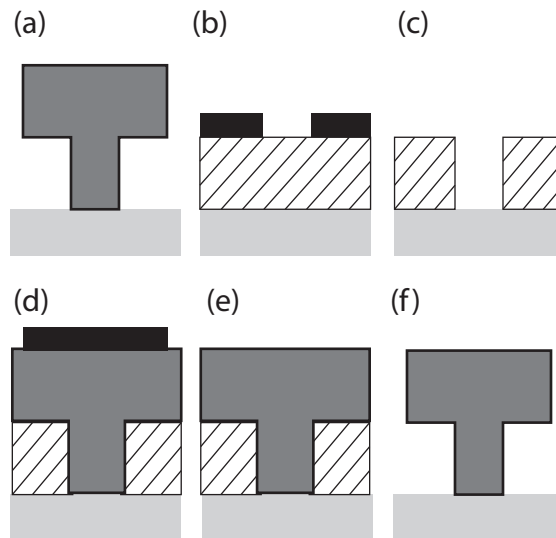


Figure 1.1. Surface Micromachining of a simple part. (a) Target polysilicon part on a silicon substrate. (b) A sacrificial oxide layer is deposited and a mask is placed. (c) The oxide is etched and the mask removed. (d) Polysilicon is deposited and a mask is placed. (e) Extraneous polysilicon is removed and the mask is removed. (f) The sacrificial silicon oxide is removed.

Topological Approach

Surface micro-machining builds a MEMS device with the successive deposition and controlled etching of materials on a silicon substrate. Motivation for an alternative approach can be found in consideration of the process steps typical of surface micro-machining. For example, to produce the simple part shown in figure 1.1a, two layers of deposited material and two masks are used. First, a layer of silicon dioxide is deposited (silicon dioxide is commonly used as a supporting material since it can easily be removed at the end of the process) and a mask is used to define the region of silicon dioxide to be retained as shown in figure 1.1b. Unmasked silicon dioxide is etched away resulting in the structure shown in figure 1.1c once the mask is removed. Next, a layer of polysilicon is deposited. As before, a mask is used to define the region of polysilicon to retain during the next etch; see figure 1.1d. Etching the extraneous polysilicon and removing the mask produces the part shown in figure 1.1e. Finally, removal of the sacrificial silicon via chemical dissolution reveals the final, desired part as depicted in figure 1.1f.

Considering the production of this simple device, one can identify two, horizontal

cross-sections in the 3D object which directly correlate to the masks used to manufacture the device. First, the narrow cross section of the post relates to the mask used to etch the sacrificial oxide. Second, the cross section of the larger top directly correlates to the mask used to produce the top section. Therefore, if important cross sections can be identified in a 3D model, then these cross sections can be used to create masks to manufacture the device.

Chapter 2

Cross Sections to Masks

Considering the example of figure 1.1, the horizontal cross sections of a device can be used to identify the masks. Given a body, let z represent the scalar distance from a reference ground plane and let $C(z)$ denote the cross section of a body at the height z . The function $C(z)$ is not necessarily a continuous function of height as a part with exactly vertical sides will create discontinuities in $C(z)$ when the cross section changes.

While a cross section itself is infinitely thin, one can identify a range of heights within which a given cross section is constant. Thus, if one defines \mathcal{C} as a constant cross section in some range of heights z_i, z_{i+1} , then one can define a constant cross section associated with that height range, P_i as:

$$P_i = C(z_i) : C(z) = \mathcal{C} \quad \forall z \in [z_i, z_{i+1}) \quad (2.1)$$

Specifically, P_i maps infinite sets of cross sections to a single valued, but not continuous function. In defining P_i , one has implicitly subdivided the z domain into intervals within which a given cross section is constant. Since $C(z)$ may not be continuous, the range of heights within which $C(z)$ is constant cannot easily be defined as closed. Specifically, if a discontinuity occurs at z_i , then the value of $C(z_i)$ depends on which side from which the discontinuity is approached. Thus a range of acceptable z values can be written as either $[z_i, z_{i+1})$ or $(z_i, z_{i+1}]$ and the sequence z_i may be increasing or decreasing. In this analysis, it is assumed that the range of allowed heights is traversed from top to bottom implying $z_i > z_{i+1}$ and that any discontinuities in $C(z)$ are placed at the lower height yielding the closure defined in equation 2.1. Now, given P_i a set can be defined as follows:

$$U = \{P_1, P_2, \dots, P_N\} \quad (2.2)$$

This defines U as the sequential set of all unique cross sections for a given body. Note that unique here only implies that P_i is not equivalent to either P_{i-1} or P_{i+1} , *i.e.* unique relative to one's neighbors. With a notation on hand to describe a body's cross sections and where they arise, attention next will be directed to organizing the cross sections into a useful topology tree.

Topology Graph Analysis

Considering again the example of figure 1.1a, one can find two unique cross sections for such a part yielding $U = \{P_1, P_2\}$. Since neither cross section is composed of multiple subcomponents, one can see that P_1 is connected to P_2 . This topological relationship can be represented by $P_1 \rightarrow P_2$. In general, a given cross section may contain multiple subcomponents, islands or lumps. To account for this one can expand the definition of P_i as:

$$P_i = L_{ij} : j = 1, J \quad (2.3)$$

where J is the number of subcomponents or lumps of cross section P_i . Using the notation L_{ij} to denote a lump of a given cross section, a graph or tree can be constructed relating the connectivity of the lumps of the various cross sections. For example, the following tree could relate three cross sections where the middle cross section has two lumps:



In the above tree diagram, each lump is a node. The next step in the analysis is to categorize the nodes as follows. For each node, its surface area is calculated and then compared to child and parent nodes to determine if the current node is a local maxima or local minima in cross sectional area. Nodes are categorized as: head nodes, end nodes, local maxima or local minima. Head nodes are nodes without parents. End nodes have no child nodes. Local-maxima or local minima nodes are extrema in the local cross sectional area, maxima or minima respectively. Local minima in particular are important as they typically indicate where one deposition layer of material joins a material deposited at a later time. If an extrema in cross sectional area occurs at a head or terminal node then special process masks may be required. No masking decisions are made at this stage; rather these nodes are just marked so that they can receive attention during the mask reconciliation stage.

Once the nodes are categorized, the tree is traversed to find the cross sections required to build the device. It is assumed at this stage that the surface micromachining process proceeds by depositing a layer of material, using a mask to etch away unwanted material, removing the mask and then repeating this process. This is a simplification that real processes do not necessarily follow which can be accounted for at a later stage in the analysis and will be covered in the next section.

While traversing each branch of the tree, the locations of local minima nodes are first recorded and between any pair of local minima on a given branch, a local maxima

is sought. Local maxima nodes are categorized as candidate *poly* masks as they typically represent how a structural layer like a polysilicon layer was masked before etching; *poly* is purely a name of convenience as this method would work for any material. Similarly, local minima nodes are categorized as candidate *sac-ox* masks as they typically correspond to masks used in etching sacrificial layers like the sacrificial oxide layers, SiO₂. Again, this nomenclature is for convenience. Terminating nodes that end in local extrema are categorized as *dimples* if they are local minima or *undercuts* if they are local maxima. These two mask types are almost equivalent to *sac-ox* and *poly* masks respectively, however their use in a fabrication process is different from *sac-ox* and *poly* masks so they are singled out at this stage.

The masks thus far identified have an additional attribute associated with them. Each mask has a *thickness* which corresponds to the difference in height between the node where the mask was identified and either the next extrema on a child branch or the end of the current branch. When attempting to match or reconcile the masks found from the topology analysis with masks required for given process, this thickness is used to determine if a given process step is compatible with a given mask. Next, these masks will be converted to production masks.

Creating Production Masks

The candidate masks found in the previous section apply only to an idealized version of surface micromachining as was assumed earlier. If one were only given a model of a part, and the part's designer did not have a specific production process in mind for that part, then the candidate masks together with their thicknesses and material types would define a new, idealized production scheme for this device.

However, if the designer of this part had a specific production process in mind then the candidate masks must be reconciled with process mask specifications to yield valid process masks. This is done as follows. First the process specification is searched for the materials and material thicknesses it uses, mask names and their locations in the process stream. Next, the target process is searched for places where the assumed deposition-mask-etch process order does not occur. With these parameters known, the candidate masks can be searched for masks that match the function of those used in a given process step. If a candidate mask corresponds to a layer which is thicker than layers in the target process, then that mask can be duplicated and used to produce two laminated layers in the actual process. If all of the candidate masks cannot be fit to the target process then the designer can be informed of what feature is blocking this fit.

Chapter 3

Algorithm and Implementation

The analysis described in the previous section forms the basis of the following algorithm, which successfully infers 2D mask sets from complex 3D models. Aspects of the algorithm that have not yet been discussed concern largely logistical points. For example, a given 3D model may have many non-intersecting bodies. If so, it is efficient to work on one body at a time; so initially the model is divided into its non-intersecting components or bodies. Figure 3.1 demonstrates this deconstruction.

Compensation for this division occurs later when the mask sets are summed. This summation is straightforward as the non-intersecting bodies will have non-overlapping masks. Finally, a simplification of the topology tree is conducted where redundant nodes are joined, a process where by nodes that topologically connect the same nodes are combined to one node.

Given a 3D model, the algorithm can be summarized as:

1. Locate independent bodies.
 - (a) find all non-intersecting bodies
 - (b) separate bodies made of different materials
 - (c) separate bodies only connected via. ground
2. For each body
 - (a) Generate a topology tree.
 - (b) Categorize the nodes of the tree.
 - (c) Combine redundant nodes.
 - (d) Locate deposition boundaries.
3. For each deposition domain
 - (a) Locate masks
 - (b) Save masks in candidate mask set
4. Sum all candidate masks

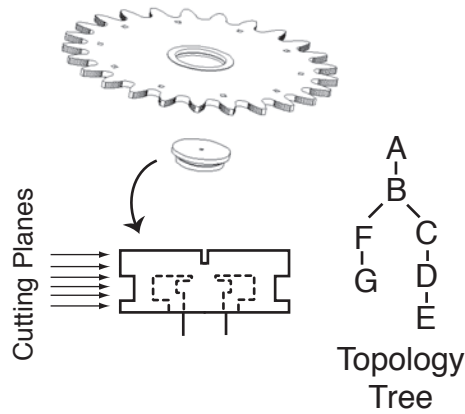


Figure 3.1. Locating the unique cross sections and building the topology tree for a hub which holds a gear in place.

5. Reconcile masks with the target process.

It is significant to note that specific process details do not enter the algorithm until the final step. Allowing most of the algorithm to operate independently of process details keeps the algorithm flexible to process changes. This technique of mask identification has been patented under U.S. Patent 7,065,736. [6]

Example

As an illustrative example of this method, figure 3.1 depicts a hub which is used to hold a gear in place. A single hub is an example of an independent, non-intersecting body found in step one of the method listed previously. The hub is cut into horizontal cross sections and the unique cross sections are assembled into a topology tree. The computational cost of cross sectioning and creating a topology tree typically scales with the number of vertical surface in a model squared. Thus identifying independent model bodies at the start of this method is an important aspect in method throughput.

Continuing, note that the topology tree developed for the hub is branched and non-symmetric as the center post has a different topology than the outer ring of the hub. Figure 3.3 demonstrates the analysis of the topology tree. After the area of each topology node is calculated, an area versus height graph is created where the vertical lines connect the nodes to indicate topological relationships. Since the hub's topology tree is branched, the branch for the outer part of the hub is drawn with a dashed line. Using the area data and the topological connectivity of the nodes, candidate masks can be selected.

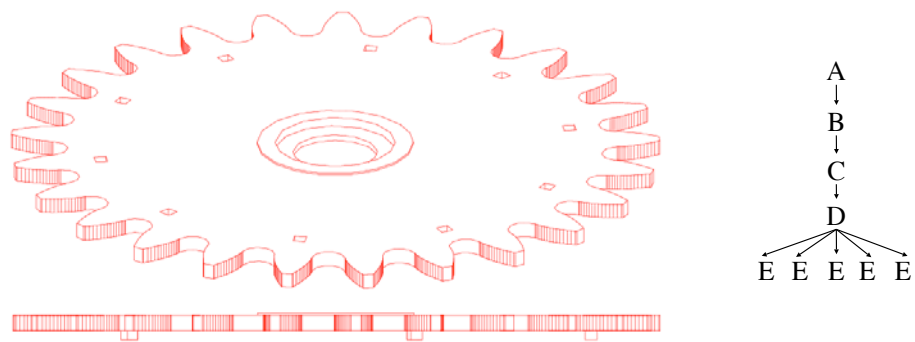


Figure 3.2. This gear has five stubs or stand-offs on its lower face. Since they originate from the same topology node and end at the same location, these five, redundant nodes can be combined into one.

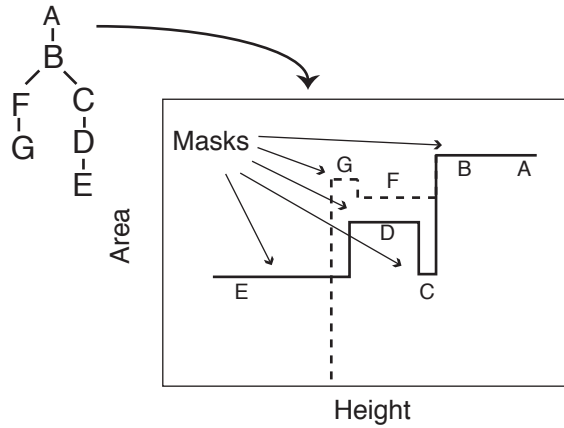


Figure 3.3. Analyzing the topology tree allows one to locate masks.

Reconciliation of the masks with process constraints produces a set of production masks for the hub. Operations on masks to convert candidate masks into process masks include combining masks that operate on the same material at the same point in a process, discarding redundant mask features and inserting masks required by a given process.

Implementation

The algorithm was implemented in a C++ program called `faethm` using the ACIS geometric modeling library version 16 (<http://www.spatial.com>) for import and manipulation of the 3D models. Sandia's 2D Boolean library was used to handle the 2D cross section and subsequent masks [4]. Models were both manually generated and provided by Sandia's 3D Modeler [5]. Note, Sandia's second generation modeler, SummitView, was not used in testing this program.

Use

Faethm is a command line driven program that takes input files and stores its results in output files. The syntax for invoking `faethm` is:

```
faethm <assembly.sat>
```

where `assembly.sat` is the name of an ACIS sat file describing a three dimensional assembly.

Optional switches:

- `--crossSections` Output cross sections derived from the 3D model. In the first stage of Faethm's analysis, a 3D model is broken down to cross sections. This switch forces faethm to save the cross sections as files called `crossSection[n].dxf` where `n` is $0 \dots N$ an increasing integer to make each file name unique.
- `--mergeMaterials` merge components of the same material type before analysis. This is useful when the original model has bodies of the same material type in contact which would effectively be a single body in a finished part. For example, Sandia's 3D Modeler [5] generates 3D models with bodies built up by each process step. To aid in the visualization of each production step, these bodies are not united even though the final, real part would have such bodies united.
- `--outputLevel` [`none,minimal,normal,moderate,verbose`] Instructs faethm to generate more, or less output. More output is useful to gauge faethm's progress on large models.
- `--process` `<process name>` The name of the process to be used for reconciliation. This process name and its full definition must be in the process description file.
- `--processFile` `<process file name>` The process description file which describes the steps for a given production process.
- `--trialMasks` Forces faethm to output trial masks in a file called `trialMasks.dxf` where each layer in this AutoCAD file specifies a candidate mask. While normally not generated, trial masks will be saved if faethm is started without a specified process.

To aid the used in running faethm, a simple, graphical user interface has been created to control faethm and select the runtime options. One can start the GUI by double clicking on the FaethmNetApp icon. Figure 3.4 shows a screen capture of the GUI in operation.

To convert a 3D part to 2D cross sections or masks using the GUI, do the following:

1. Select your 3D model in the Input File box. This model must be in ACIS SAT format.
2. Select a directory to hold the output in the Output Directory box.
3. Using the checkboxes, select options you require.

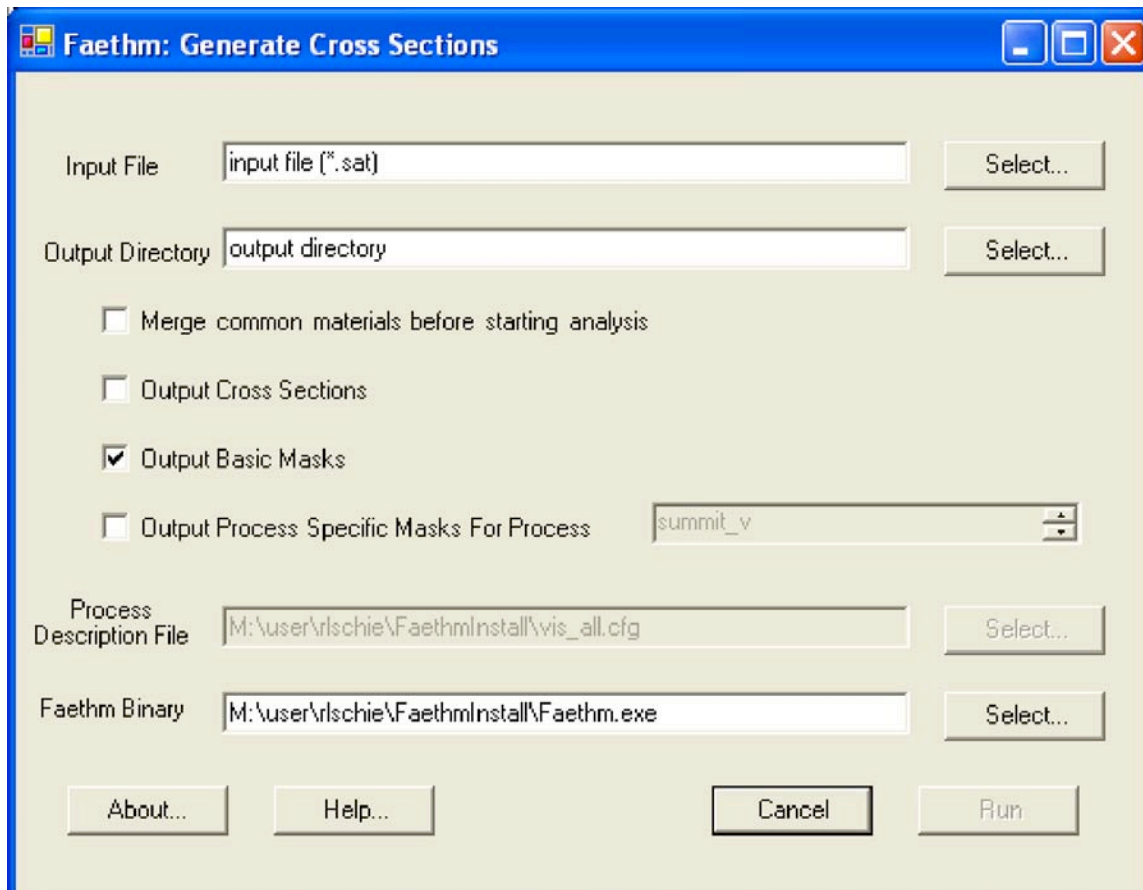


Figure 3.4. Graphical user input form for using Faethm

- a. Selecting Merge common materials before starting analysis will force Faethm to unite any bodies that are in direct contact and of the same material type before trying to find the masks for that device.
 - b. Selecting Output Cross Sections will output all of the horizontal cross sections found in the model
 - c. Selecting Output Basic Masks will output the cross sections that would likely be used as process masks.
 - d. Selecting Output Process Specific Masks will attempt to produce masks for a specific production process. Note, Summit V is the only currently supported process.
4. If needed, edit the Process Description File and Faethm Binary boxes to point to the current program and process file. Typically you will not need to edit these fields.
 5. Click on Run. Faethm will run in a new window and store the results in the output directory specified.

Performance

Three-dimensional models from Sandia's Standard Parts Library have been used to test the program. Testing and debugging of faethm was done with parts from Sandia's Standard Parts library. Table 3.1 lists the parts used and their size. Also listed is a count of the number of vertices in a given model which is a measure of the complexity of the geometrical construct.

On a typical workstation, models containing 3,000 vertices are reduced to masks in under 10 seconds as shown in figure 3.5. More complex models with 10,000 vertices can be analyzed in approximately 100 to 300 seconds. However, as can be seen from the log-log scale of figure 3.5 models with over 10,000 vertices are computationally intensive and take significantly more time to complete.

Several techniques have been used to improve the runtime to its current state. First, non-intersecting geometry operations are conducted at the same time – making the geometry calculations concurrent. For example if comparing two cross sections to see which lumps continue from one cross section to the next, all the intersection calculations are done at the same time. While this does not reduce the computation involved in actually computing an intersection, it pools all the overhead of manipulating the cross sections prior to and during the calculation reducing overhead time. As shown in figure 3.5 this technique dramatically improves performance for models under 10,000 vertices.

Additionally, faethm switches from the general 3D library of ACIS to an optimized 2D shape library once all the 3D models have been reduced to 2D cross sections.

Part File Name	Vertex Count	Model Size
3D_SNL_GEARCLIP_A00_merged.sat	128	73 KB
3D_SNL_MIRROREXTENSION_B00_merged.sat	296	168 KB
3D_SNL_MIRROREND_B00_merged.sat	368	177 KB
3D_SNL_INTC_BRIDGE_SINGLECROSS_A00_merged.sat	336	198 KB
3D_SNL_HINGE_PLATE2PLATE_A00_merged.sat	332	230 KB
3D_SNL_HINGEREGULARANCHOR_A00_merged.sat	312	236 KB
3D_SNL_P0TOSUBSTRATETHIN_B00_merged.sat	396	256 KB
3D_SNL_ELECSTC_BOND_PAD_A00_merged.sat	408	264 KB
3D_SNL_TOOTHCLAMP_100T_A00_merged.sat	504	279 KB
3D_SNL_HINGEREGULAR_A00_merged.sat	544	306 KB
3D_SNL_INTC_BRIDGE_FULLCROSS_A00_merged.sat	568	388 KB
3D_SNL_TOOTHCLAMP10038_A00_merged.sat	616	405 KB
3D_SNL_THUNDERBIRD_P2_A00_merged.sat	1,156	702 KB
3D_SNL_DISPLACEMENT_GAUGE_A00_merged.sat	1,198	731 KB
3D_SNL_SPRING_STOP_P4_A00_merged.sat	1,300	770 KB
3D_SNL_SUBMICRON_VERNIER_A00_merged.sat	1,624	872 KB
3D_SNL_THUNDERBIRD_P3_A00_merged.sat	2,174	964 KB
3D_SNL_THUNDERBIRD_P4_A00_merged.sat	1,818	1,186 KB
3D_SNL_FORCE_RING_A00_merged.sat	2,420	1,849 KB
3D_SNL_SINGLESIDERACK_MIRROR_A00_merged.sat	3,018	1,908 KB
3D_SNL_FLXLINK_P4_18T_A00_merged.sat	3,388	2,596 KB
3D_SNL_FLXLINK_FLEXURAL_P3_18T_A00_merged.sat	3,498	2,662 KB
3D_SNL_ELECSTC_BOND_PAD_SUB_A00_merged.sat	4,368	2,931 KB
3D_SNL_SINGLESIDERACK_A00_merged.sat	3,804	2,976 KB
3D_SNL_5X_DISP10UM_A00_merged.sat	4,160	3,739 KB
3D_SNL_12X_DISP40UM_A00_merged.sat	4,336	3,821 KB
3D_SNL_CANTI_ENG_STORAGE_A00_merged.sat	7,416	5,841 KB
3D_SNL_TRANSMISSION12_A00_merged.sat	9,886	7,722 KB
3D_SNL_INDEXDRIVE_A00_merged.sat	10,902	8,241 KB
3D_SNL_SPIRALSPRING1_B00_merged.sat	11,240	24,740 KB
3D_SNL_SPIRALSPRING1_C00_merged.sat	36,366	33,414 KB
3D_SNL_TRA_A00_merged.sat	43,356	40,618 KB

Table 3.1. Models from Sandia’s Standard Parts Library which were used to test Faethm. All models were *merged* prior to running through faethm. This combines all bodies that are in contact and of the same type of material into single bodies.

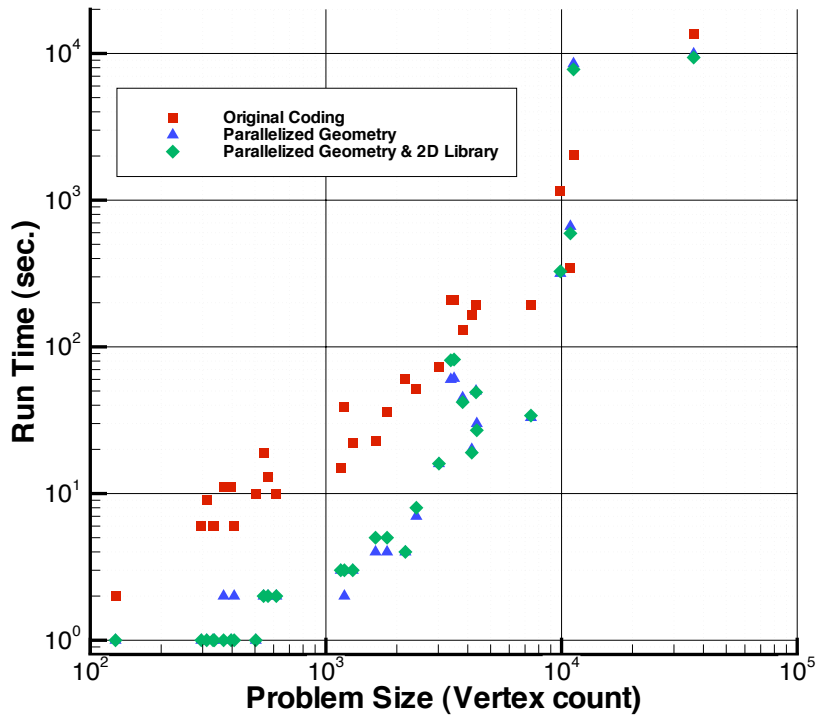


Figure 3.5. Typical program run times for models of increasing complexity (higher vertex count for larger more complex models).

This simplifies the computational work as the 2D shapes are easier to describe mathematically and store externally. In general, this change has had a small improvement on runtimes for models under 1,000 vertices and a small negative impact on larger models. This reduction in speed may be due to the overhead of re-constructing the 2d shape in the new library which takes time that is not made up by faster 2D operations later.

Despite the improvements in faethm’s runtime, there are some limitations in its current capabilities. First, since faethm was tested with models from Sandia’s 3D modeler, faethm assumes that material data (i.e. what material a given part is made from) are encoded in the model file in the same way that the 3D Modeler encodes this information. Thus, models developed in a 3D design package such as Solidworks cannot currently have material type information passed to Faethm. Work is continuing to resolve this issue.



Figure 3.6. 3D model rendering artifact's such as the long, thin *stringer* shown here can cause errors in faethm's mask calculations.

Second, if the 3D model being processed contains artifacts such as *stringers* shown in figure 3.6, these artifacts can cause errors in faethm's mask generation calculations. While such artifacts can be difficult to locate in 3D models, they typically are not manually added by a designer. Normally, such artifacts arise when the 3D models are automatically generated.

Finally, Faethm has been tested with models from Sandia's Summit V process and not other processes. Thus, mask reconciliation to other processes should be done with caution and the results carefully reviewed.

Chapter 4

Conclusion

The algorithm presented here and coded in `faethm` is capable of generating accurate mask sets for complex 3D devices. By focusing on a model's topology first, this work can identify masks for anisotropic and isotropic (dry and wet) etching processes. Fairly large scale models can be reduced to masks in a reasonable timeframe. For very large models (over 40,000 vertices) more work is needed to improve `faethm`'s mask generation speed.

References

- [1] L. Ma & E. Antonsson, “Robust mask-layout and process synthesis” *J. of Microelectromechanical Systems* vol. 12, no. 5, pp. 728–729 (2003).
- [2] V. Venkataraman, R. Sarma & S. Ananthasuresh, “Part to Art: Basis for a Systematic Geometric Design Tool for Surface Micromachined MEMS” *Proceedings of the ASME Design Engineering Technical Conferences 2000* pp. 1–14 (2000).
- [3] S. Cho, K. Lee & T. Kim, “Development of a geometry-based process planning system for surface micromachining” *Int. J. of Production Research* vol. 40, no. 5, pp. 1275–1293 (2002).
- [4] C. L. McBride, V. Yarberry, R. C. Schmidt & R. J. Meyers, “GBL-2D Version 1.0: A 2D Geometry Boolean Library” *Sandia National Labs, SAND Report SAND2006-6829* (2006).
- [5] C. Jorgensen & V. Yarberry, “A 3D Geometry Modeler for the SUMMiT V MEMS Designer” *Modeling and Simulation of Microsystems 2001* pp. 594–597 (2001).
- [6] R. Schiek, “System for generating two-dimensional masks from a three-dimensional model using topological analysis.” United States Patent 7,065,736 June 20 (2006).

DISTRIBUTION:

1	MS 0316	Scott Hutchinson, 1437
10	MS 0316	Richard Schiek, 1437
5	MS 0316	Rodney Schmidt, 1437
5	MS 1069	Victor Yarberry, 1737
1	MS 0899	Technical Library, 9536 (electronic copy)

