Computer Aided Mask-Layout for Bulk Etch Fabrication

Mark K. Long, Joel W. Burdick, and Erik K. Antonsson

Abstract—This paper presents a method to synthesize the mask layout geometry for a MEMS wet etching process. Given a desired part geometry, the method determines a candidate mask geometry that will etch to the final desired shape, even in the case of highly anisotropic etchants. It will also compute compensation structures for difficult to etch features. In cases where there does not exist a mask geometry that will etch to the desired feature, an approximate shape is produced. Conceptually, the algorithm is based on the use of a forward etch simulation in reverse time. Since the forward etch process is a many-to-one map, the reverse time simulation is augmented to include the set of valid preimages. While the methods are not inherently restricted to planar geometry, our discussion is limited to the case of planar polygonal feature geometries.

Keywords— Anisotropic Etching, Mask-layout, bulk etching, wet etching, KOH, EDP, TMAH, photo-lithography.

I. Introduction

There is a significant need for engineering tools to support MEMS development by automating routine engineering functions. This paper describes an engineering tool for automated mask-layout for wet etch micro-machining.

At present, a designer conceives of a MEMS function, then (informally) creates a mask-layout that the designer believes will etch into a shape that will exhibit the desired function. This process is based largely on the engineers intuition and experience, and can be difficult for complex geometries or for novice engineers. A prototype device is created from the candidate mask, and its actual function is tested. This process may result in many design iterations and prototypes. In some cases, manufacturing issues dominate the design cycle.

Recently, researchers have developed algorithms and procedures to model and simulate the etching process. While these simulators help reduce the number of prototypes, current design procedures still rely heavily upon the designer's intuitive understanding of the etching process. Additionally, the wide variety of micro-machining processes in use today make this intuitive work even more difficult.

There has been diverse work in the general area of MEMS-CAD. Many of these tools deal with developing devices for a specific function or analyzing the devices functional or mechanical properties [1], [2], [3]. Often manufacturing issues are not addressed. The work in this paper is complimentary to these prior developments in that only the effect of fabrication issues on the design is being addressed. Thus, this work has an analogous role to that of traditional computer-aided-machining packages.

Traditional computer aided (macro scale) machining has two basic parts. Computer aided machining (CAM) software packages are used to set up the part geometry and generate the method to machine the part. The second part is the actual physical machining of the part which is handled by a CNC (computer numerically controlled) machine tool. In this paper we will only look at the software planning (CAM) part and how some similar benefits may be provided in the micro-machining process.

We can draw functional analogies between macro CAM and micro-scale mask layout to give us a place to start in developing tools for computer aided micro-machining (μCAM).The capabilities of traditional macro-scale CAM tools suggests the following goals for a μCAM tool:

- Automate the mask synthesis procedure.
- Aid in selecting the proper etchant and process parameters(temperature, concentrations, etc.).
- Reduce Design Iterations.
- Increase Quality, Tolerance, Finish, and Repeatability.
- Shift focus to part geometry from machining procedure.

This paper develops a method to synthesize the mask layout for a given etching process. That is, given a desired part geometry and process characteristics, the computer aided micro-machining algorithm determines a candidate mask geometry that will etch to the final desired shape, even in the case of highly anisotropic etchants. *In certain cases, there does not exist a mask geometry that will yield the desired shape. The algorithm produces an approximate shape in this case.* While our methods are not inherently restricted to planar geometry, this paper considers only the case of 2-dimensions. We also restrict our discussion to polygonal geometry. Curvilinear shapes can be well approximated by faceted polygons.

The mask synthesis procedure can be roughly thought of as a reverse simulation process. The evolution of the part shape during the etch process is a many-to-one map. The challenge in the mask synthesis algorithm is to preserve this feature.

Polygonal shapes decompose into edges and vertices, vertices are categorized as either concave or convex. Further we categorize vertices in which new planes appear during the etching process as *compound* and those vertices in which no planes appear as *simple*. This taxonomy will be very useful in our computation of mask shapes for desired output geometry. Mask geometry that produces a desired feature of the output is its *preimage*.

In this paper we will describe algorithms and procedures for computing the preimage of each of the four types of polygonal vertices, *simple-concave, compound-concave, simple-convex*, and *compound-convex*. While simple vertices have a single *direct preimage* which is easily computed,

The authors are with the Department of Mechanical Engineering, California Institute of Technology, U.S.A., e-mail: long@caltech.edu

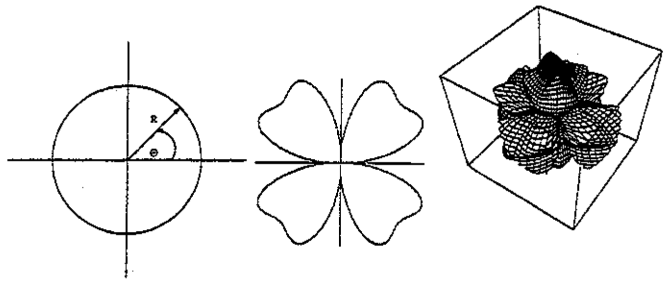

Fig. 1. Polar Etch Rate Diagrams (etch rate R vs. crystal orientation θ: 2D-Isotropic, 2D-Anisotropic, 3D-Anisotropic

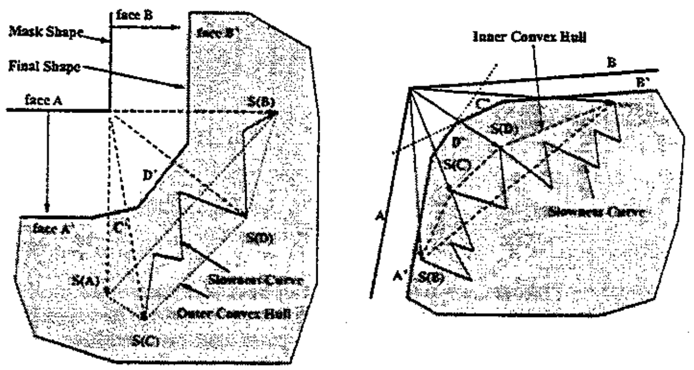

Fig. 2. Convex Hull Construction Over Slowness Diagram

they also have multiple *indirect preimages*.

We also show that *compound convex* vertices have no *direct preimage* and that we can algorithmically construct a valid *indirect preimage*, also called a *compensation* structure. Finally, we will discuss the lack of any valid exact preimages for *compound concave* vertices and methods to approximate the geometry.

II. Approach

Wulf-Jaccodine [4], E-Shape [5], Slowness [6], [7], [8], and Cellular Automata [5], [9] are a few of the proposed methods used to simulate etchant based micro-machining process. Because of the complexities of anisotropic micro-machining, the anisotropic case has received much of the attention. This paper also focuses on the anisotropic case.

Simplisticly, our approach is based on reverse time simulation of the etch process and this want to identify a forward simulation technique that can be inverted and can also handle isotropic and anisotropic processes. The slowness method satisfies these requirements and is based on some well known tools in computational geometry that lend to its ease of inversion. The slowness method also is extensible from two to three dimensions and does not preclude any etchants, that is, a continuum of etching characteristics from isotropic to highly anisotropic are handled. We now consider the forward etch process to point out those features that are required for our inversion algorithm.

A. Etch Forward Simulation

The forward simulation method developed by Sequin [8] and Foote [7], uses convex hulls to determine the rate of etching of plane sections and also the change in topology of the etched geometry.

The rate of etching of a plane section is given by the etch rate diagram [10], [11], [5], Figure 1. The etch rate diagram for a particular process is a polar or spherical plot of the angle of the 'etch face normal' plotted against the rate of etch per unit time in that direction. The inverse diagram, sometimes called the "slowness" or "reciprocal rate" diagram [6], [7], is also useful. This diagram is a polar plot of the inverse of the etch rate against the etch face normal angle, Figure 3.

As two adjacent planes advance during the etch process, the vertex that joins them also advances. Since the two planes may be etched at different rates, the path of the vertex may not be straight. Additionally, these plane edges will increase or decrease in length during etching due to this effect. Modeling and predicting the disappearance of a plane or the appearance of new planes at a vertex is fundamental to an etch simulation and ultimately to our mask layout process.

Given a planar polygonal object consisting of n faces and n vertices, the propagation of a *simple* vertex i under the etching process can be described by a single *vertex velocity vector* $\mathbf{V}_i$, which describes the direction and rate at which the corner propagates:

$$\mathbf{V}_i = r_i\hat{\mathbf{n}}_i + \frac{r_{i+1} - r_i(\hat{\mathbf{n}}_i \cdot \hat{\mathbf{n}}_{i+1})}{\hat{\mathbf{n}}_{i+1} \cdot \hat{\tau}_i} \cdot \hat{\tau}_i \tag{1}$$

where r_j is the etch rate associated with edge e_j, and $\hat{\tau}_j$ and $\hat{\mathbf{n}}_j$ respectively denote the unit vectors tangent and normal to edge e_j. Note that r_j is determined from the etch rate diagram.

A convex hull construction is used to determine if new planes will appear as a vertex is etched forward [6], [7], [8]. In Figure 2, face A and face B meet at a vertex. Each face has a corresponding etch slowness vector S(A) and S(B). We define two different convex hull constructions, one for each of the convex and concave class of vertices. The "outer" convex hull, Figure 2A, is the hull formed by the S(A)→S(B) line and the vertices of the slowness diagram outside this line. The "inner" convex hull, Figure 2B, is the hull formed by the S(A)→S(B) line and the vertices of the slowness diagram inside this line. If there exist hull vertices outside the S(A)→S(B) line, then the *concave* vertex splits into multiple vertices, each with its own unique velocity vector. Likewise, if there exists vertices inside the S(A)→S(B) line for a *convex* vertex, then the vertex splits and new plane(s) are formed. The vertices of the inner and outer convex hulls correspond to the slowness vectors (reciprocal rates) for the appearing planes. Note the compound/simple characterization is a function of the local geometry, its relative orientation to the crystal, etchant properties, and process parameters.

In the case of pseudo-isotropic processes, many, if not infinite, planes appear when certain vertices are etched.

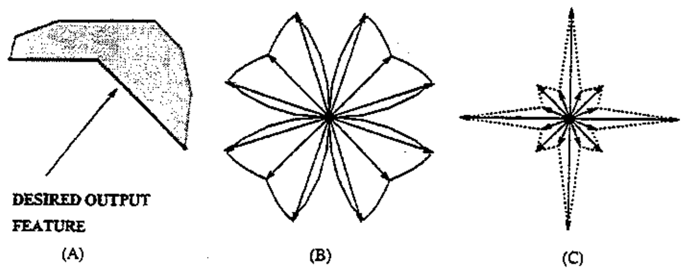

Fig. 3. (A) Concave Vertex, (B) Etch Rate Diagram, (C) Slowness Diagram

The outer or inner convex hull may follow a curved section of the slowness diagram. This emerging curved shape is then approximated with facets at a specified resolution.

As discussed in [8] this method extends to 3 dimensions.

III. μCAM Procedure

We now present a method to compute the preimage for each of the four types of polygonal vertices ("features"), *simple-concave, simple-convex, compound-concave,* and *compound-convex.* The "feature preimage" is a local mask shape, that when etched for the specified time, will produce the desired single vertex in the specified position and orientation. In some cases the preimages for two or more vertices may have conflicts or overlaps. Computing a set of preimages for each feature will allow global interactions to be handled if necessary.

Given a description of the desired output geometry, the first task is to classify each vertex into one of the four categories given above. Each vertex can be determined to be concave or convex by computing the included angle and determining if it is less than or greater than π.

To determine if a vertex is simple or compound we compute the convex hull of slowness vectors of the faces which form the vertex and the included portion of the slowness diagram.

A. Simple Concave Vertices

Simple-concave vertices have multiple preimages which will etch to the desired shape. Therefore techniques for computing multiple preimages are developed. Secondary considerations may then be used to select a given preimage from the range of possible preimages.

A.1 Simple Concave Direct Preimage

For the *simple-concave* vertex there exists a *direct preimage.* The direct preimage is formed by running the forward simulation in reverse-time without consideration for potential topology changes (additional or deletion of planes/vertices during the etch).

The procedure for *simple-concave* vertices is demonstrated by example. A concave corner and the etch rate and slowness diagrams for an EDP-like process is shown in Figure 3. Figure 4 shows the slowness vectors for the two faces forming the corner, the included portion of the slowness diagram and the convex hull of the slowness vectors. This diagram verifies that the slowness diagram does not form any vertices in the outer convex hull, thus the hull collapses to a line and this vertex is *simple.*

The propagation of vertex i during the etching process can be described by a single *vertex velocity vector* $\mathbf{V}_i$,(eq. 1), which describes the direction and rate at which the corner propagates. The *direct preimage* for a given etch time, τ, is computed by simply reversing the sign of $\mathbf{V}_i$ as shown in Figure 4. While the direct preimage provides a single solution, there exists other valid preimages which etch the desired vertex.

A.2 Simple-Concave Inner-Hull Preimage

To compute an alternative preimage, vertices contributed by the inner convex hull are used. These vertices represent the geometry which have the highest etch rates, yet will still converge to the vertex at the desired etch time, τ. Since for some plane that exists in the preimage but disappears at some time $t_1, 0 < t_1 < \tau$, there exists a plane further from the vertex that disappears at some time t_2, $t_1 < t_2 < \tau$, the plane geometry "furthest" from the vertex must disappear exactly at $t = \tau$.

By forming the "inner" convex hull of the face slowness vectors and the slowness diagram, the vertices of the convex hull give the fastest plane(s) in the included angle as shown in Figure 4. If the slowness diagram does not contribute a vertex to the convex hull, then this method does not contribute an additional preimage solution. Therefore we wish to find the preimage of the geometry in which these planes disappear exactly at $t = \tau$.

To visualize the approach to compute the preimage for these disappearing planes, consider the planes on the verge of disappearing at $t = \tau-$ having size ϵ, but still existing in the topology of the part geometry. This new geometric topology is composed of two or more *simple-concave* vertices that have a direct preimage which are computed as described above.

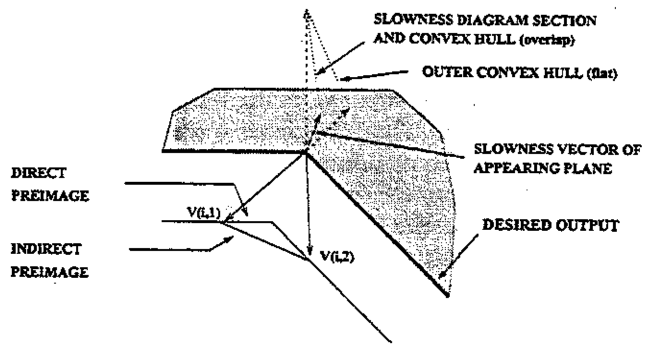
SLOWNESS DIAGRAM SECTION AND CONVEX HULL (overlap)
OUTER CONVEX HULL (flat)
SLOWNESS VECTOR OF APPEARING PLANE
DIRECT PREIMAGE
V(i,1)
INDIRECT PREIMAGE
V(i,2)
DESIRED OUTPUT

Fig. 4. Indirect Preimage Construction and Inner and Outer Preimage Bounds.

Another way to conceptualize this preimage construction is to consider these inner hull vertices as guides to compute multiple inverse vertex velocity vectors.

For our current example, Equation (2) is used to compute the multiple vertex velocity vectors, denoted by $\mathbf{V}_{i,j}$ in Figure 4, where i is the vertex number and $j \in 1, k$ refers

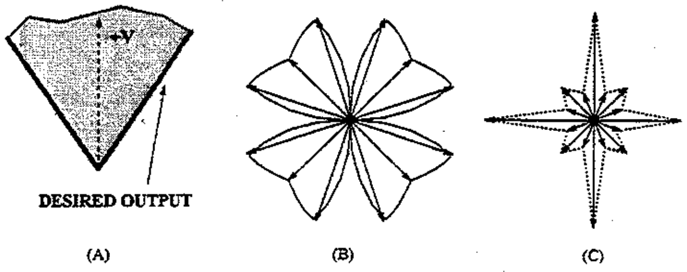

Fig. 5. (A) Convex Vertex, (B) Etch Rate Diagram, (C) Slowness Diagram

to one of the k inverse vertex velocity vectors. Here, there are two inverse vertex velocity vectors computed by:

$$\mathbf{V}_{i,1} = r_i \hat{\mathbf{n}}_i + \frac{r'_1 - r_i(\hat{\mathbf{n}}_i \cdot \hat{\mathbf{n}}'_1)}{\hat{\mathbf{n}}'_1 \cdot \hat{\tau}_i} \cdot \hat{\tau}_i \qquad (2)$$

$$\mathbf{V}_{i,2} = r'_1 \hat{\mathbf{n}}'_1 + \frac{r_{i+1} - r'_1(\hat{\mathbf{n}}'_1 \cdot \hat{\mathbf{n}}_{i+1})}{\hat{\mathbf{n}}_{i+1} \cdot \hat{\tau}'_1} \cdot \hat{\tau}'_1 \qquad (3)$$

where r_j is the etch rate associated with edge e_j, and $\hat{\tau}_j$ and $\hat{\mathbf{n}}_j$ respectively denote the unit vectors tangent and normal to e_j. r'_1,$\hat{\tau}'_1$, and $\hat{\mathbf{n}}'_1$ are the respective values for the new appearing plane (vertices of the inner convex hull).

Notice that the method presented in this section examines only those planes which lie in the inner convex hull as candidates for inclusion. Thus the turning angle of the local topology will not change sign. Also, these planes can be inserted at intermediate points, not just at $t = 0-$. In future work this restriction is lifted an examples of other candidate preimages are developed.

B. Simple Convex

Computing the preimage for the *simple-convex* vertex is very similar to the *simple-concave* case.

Figure 5 shows the convex vertex with the etch rate diagram and slowness diagram. In Figure 6 the slowness vectors for the two planes are shown with the included portion of the slowness diagram and the convex hull. Notice that in the case of a convex vertex we check the "inner" convex hull and see that there are no vertices of the hull contributed by the slowness diagram. Thus the vertex is indeed simple.

Equation (1) is used to compute the vertex velocity vector. It's sign is reversed and the direct preimage is computed in the usual way. See Figure 6.

The computation of an outer-hull preimage is similar to the inner-hull preimage procedure for the *simple-concave* vertex. The "outer" convex hull is used to determine which planes move the slowest from the vertex. These planes are used to compute the multiple inverse vertex velocity vectors $\mathbf{V}_{i,j}$ using Equation (2) as shown in Figure 6.

C. Compound Convex

Unfortunately the simple techniques that were employed for the *simple* vertices cannot be applied to compound ones. A *compound* vertex, by definition, splits into two or more vertices during etch progression, and thus has multiple forward vertex velocity vectors and an appearing plane(s).

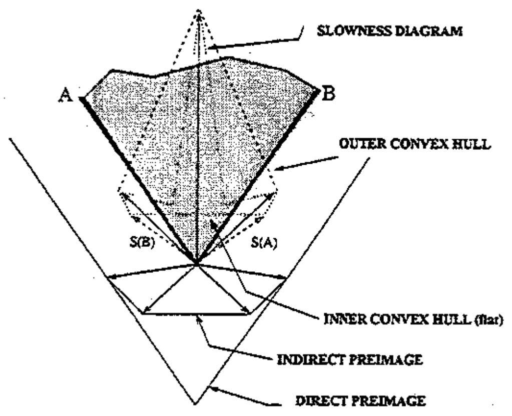

Fig. 6. Indirect Preimage Construction.

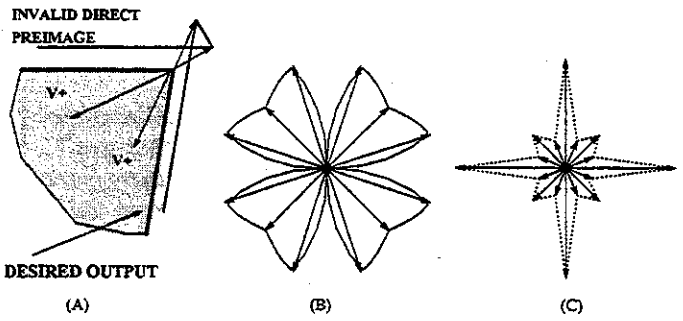

Fig. 7. (A) Compound Convex Vertex, (B) Etch Rate Diagram, (C) Slowness Diagram

The complexity here is that the desired resultant topology does not contain this appearing plane. However, any direct preimage for this vertex will create this plane at time $t = 0^+$. Thus the proper vertex could not be etched.

Figure 7 shows a convex vertex with the associated etch rate and slowness diagrams. The slowness vectors of the two faces are shown in Figure 8 with the included portion of the slowness diagram and the "inner" convex hull. The vertices in the hull due to the slowness curve correspond to the new planes that will appear in the forward etch of this shape. Thus, this is indeed a *compound* vertex.

This situation can also be examined by looking at the vertex velocity. If the simple direct preimage method were used, the vectors would cross each other at $t = 0^-$ and the resulting preimage geometry would be physically unre-

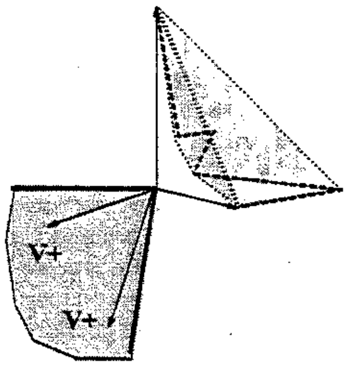

Fig. 8. Inner and Outer Hull Construction

alizable (Figure 7(a)). Therefore, in the same way that an indirect preimage for simple vertices was created, a physically realizable indirect preimage that will form the desired *compound-concave* vertex must be computed.

Preimages for etching *compound-convex* vertices have been called *compensation structures* in the literature. There are many potential topologies for compensation structures. The primary focus will be to obtain *any* valid preimage, then later discuss options for choosing from the space of potential solutions. Therefore, simple three vertex topologies will be examined first.

In order to avoid recursive compensation structures (a compensation structure with a compound vertex) only those structures with three *simple* vertices will be considered. Parameters for the description of a *compound-convex* vertex (ϕ, α_0) and a generic three vertex compensation structure $(\beta_2, \beta_3, \alpha_1)$ are show in Figure 9.

To compute a compensation structure for a given ϕ, α_0, the "$\beta_2 beta_3$" diagram is computed (Figure 10). Every point of this diagram corresponds to a possible three-point compensation structure. Those points that are white in Figure 10, are inadmissible solutions in that at least one vertex is not *simple* or the structure is not closed. The admissible solutions (in grey), are normalized to indicate the distance from the preimage vertex at α_1 to the original *compound-convex* vertex. Note, depending on particular design trade-offs, other parameters may be used to compute the contours on the $\beta_2 \cdot \beta_3$ plot.

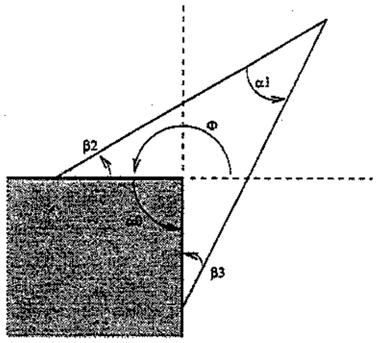

Fig. 9. Parameterized Generic Vertex and 3-Point Compensation Structure.

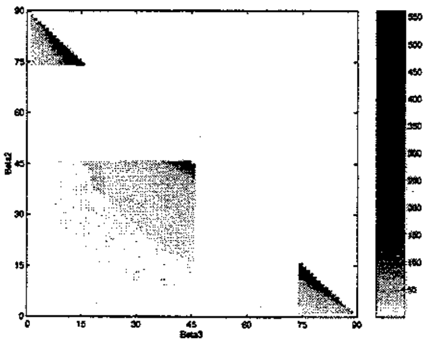

Fig. 10. $\beta_2 \cdot \beta_3$ diagram. Shaded areas are permissible compensation structures. Grayscale value is the normalized distance of the preimage vertex at α_1 from the corner vertex.

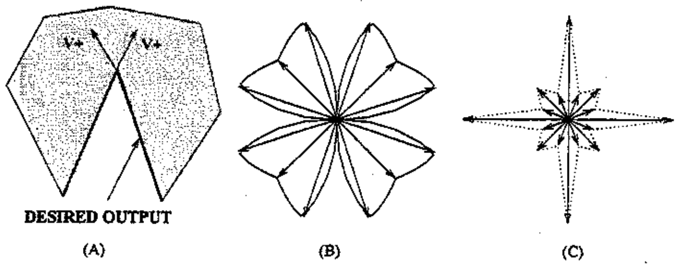

Fig. 11. (A) Compound Concave Vertex, (B) Etch Rate Diagram, (C) Slowness Diagram

Other criteria may now be used to select from the candidate structures. For example, the structure corresponding to coordinate $(25^o, 25^o)$, Figure 10, may be selected due to it's distance from graph boundaries and size of the associated compensation structure. Once the structure has been selected, the preimages for it's three *simple* verticies may be computed using the methods discussed in the previous sections. Note that the preimages for these vertices are not restricted to *direct* preimages. Resulting compensation structures are shown in Figure 14.

D. Compound Concave

There does not exist a preimage that will micro-machine a *compound-concave* vertex. Therefore, the best we can do is approximate the geometry.

Locally, any vertex preimage must have the same included angle as the two planes that form the vertex. Therefore, the plane(s) that appear at this vertex will appear in the etching of any preimage that includes this angle. Since these undesired plane(s) will appear in the output, the best we can do is to minimize their size. We know the specified etch time, and hence can compute the vertex velocity vectors for these new planes and find which neighboring planes will minimize their size.

Figure 11 shows a *compound-concave* vertex with the associated etch rate and slowness diagrams. To minimize the length of new planes in the approximated geometry, we select neighboring planes in the preimage that minimize the vertex velocity in the direction tangential to the plane itself. From Figure 12 we select a vertex with $\phi = 0$ and $\alpha_0 = 270$ (ϕ and α defined as in Figure 9). These inserted neighboring planes form a new topology for which a preimage can be computed using the methods described above. The dimension D_{min} is given by:

$$D_{min} = \epsilon + V_{R,min} \cdot \tau + V_{L,min} \cdot \tau \qquad (4)$$

where τ is the specified etch time, ϵ is the minimum line width in the mask, and $V_{R,min}$ and $V_{L,min}$ are the minimum tangential velocities of the right and left vertices respectively.

Here we have described just one of many possible approximations to the desired local geometry. Note, an approximate geometry must be chosen in which a valid preimage exists for each new vertex. This approximate geometry has now been decomposed into two *simple-concave* vertices

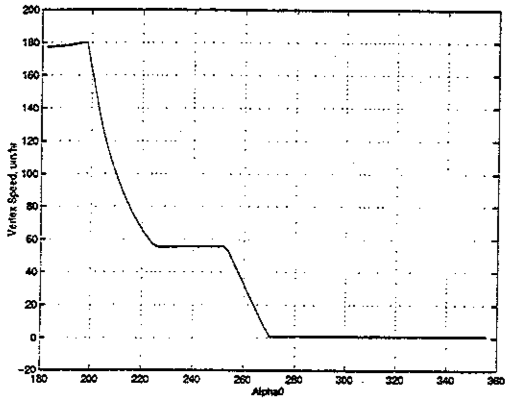

Fig. 12. Vertex Velocity for candidate insertion planes for $\phi = 0$.

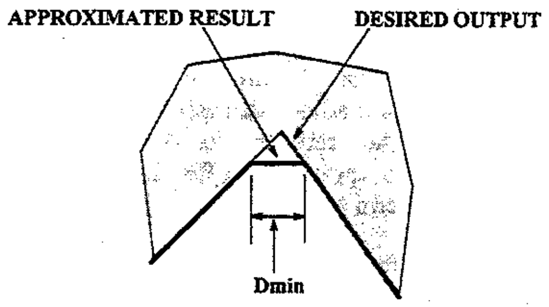

Fig. 13. Resulting Approximated Geometry.

and preimages can be computed using the methods of the previous section.

A desired part geometry and the accompanying mask is show in Figure 14. The mask was computed using the methods described in this section. Note that the different points on the $\beta_2\beta_3$ diagram were used for each of the two compensation structures.

IV. Conclusions & Future Work

We have laid a foundation for a Computer Aided Micro Machining system for wet etching. More work remains to expand the set of available preimages for each of the fours cases. Two important pieces remain to complete this system: computing mask alignment sensitivity and process calibration. One of the largest sources for errors in the photolithographic etching process is misalignment of the mask with the silicon crystal. The greater the anisotropy of the machining process being used, the greater the potential errors caused by misalignment. Future work will focus on developing a computable measure of the sensitivity of geometry to misalignment errors which will be available to the engineer during the mask layout process.

Secondly, the entire mask layout process relies on accurate characterization of the etching process being used. Most commonly this is represented in the form of an etch rate diagram. Since etching characteristics vary significantly with temperature, concentration, and even from facility to facility, it is important to develop a well engineered procedure that will accurately characterize the process being used.

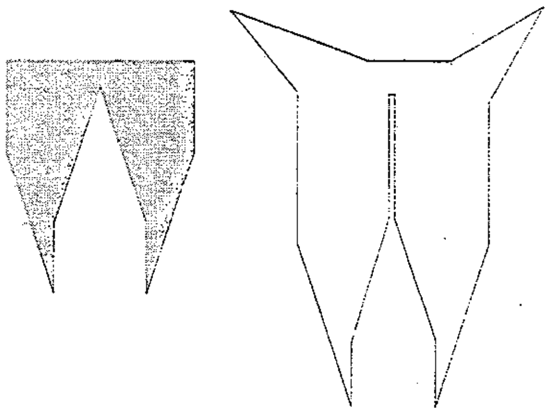

Fig. 14. Desired Geometry and Computed Mask for KOH.

References

[1] F. Pourahmadi and J. Twerdok, "Modeling micromachined sensors with finite elements," *Machine Design*, pp. 44–60, July 1990.

[2] H. U. Schwarzenbach, J. G. Korvink, M. Roos, G. Sartoris, and E. Anderheggen, "A micro electro mechanical CAD extension for SESES," *J. of Micromechanics and Microengineering*, vol. 3, pp. 118–122, 1993.

[3] S. D. Senturia, R. M. Harris, B. P. Johnson, S. Kim, M. A. Shulman, and J. K. White, "A computer-aided design system for microelectromechanical systems (MEMCAD)," *Journal of Microelectomechanical Systems*, vol. 1, pp. 3–13, Mar. 1992.

[4] J. Fruhauf, K. Trautman, J. Wittig, and D. Zeilke, "A simulation tool for orientation dependent etching," *J. of Micromechanics and Microengineering*, vol. 3, pp. 113–115, 1993.

[5] Ted J. Hubbard, *MEMS Design: The Geometry of Silicon Micromachining*, Ph.D. thesis, Caltech, 1994.

[6] F. C. Frank, "On the kinematic theory of crystal growth and dissolution processes," in *Growth and Perfection of Cyrstals; Proceedings of an International Conference on Crystal Growth.* 1958, pp. 411–419, John Wiley and Sons.

[7] Bill Foote, "Simulation of anisotropic crystal etching," M.S. thesis, U.C. Berkeley, 1990.

[8] C. H. Sequin, "Computer simulation of anisotropic crystal etching," in *Transducers '91*, San Francisco, CA, USA, 1991, pp. 801–806.

[9] O. Than and S. Buttgenbach, "Simulation of anisotropic chemical etching of crytalline silicon using a cellular automata model," *Sensors and Actuators*, vol. A45, pp. 85–89, 1994.

[10] A. Koide, K. Sato, and S. Tanaka, "Simulation of two dimensional etch profile of silicon during orientation-dependent anisotropic etching," in *Transducers '91*, Institute of Electrical Engineers, 1991, pp. 216–220.

[11] H. Seidel, L. Csepregi, A. Heuberger, and H. Baumgartel, "Anisotropic etching of crystaline silicon in alkaline solutions," *J. Electrochem. Soc.*, vol. 137, pp. 3613–3631, 1990.