Proceedings of DETC'98
1998 ASME Design Engineering Technical Conferences
September 13-16, 1998, Atlanta, Georgia, USA

DETC98/MECH-5840

EVOLUTIONARY TECHNIQUES IN MEMS SYNTHESIS

Hui Li
Engineering Design Research Laboratory
California Institute of Technology
Pasadena, California 91125
frank@design.caltech.edu

Erik K. Antonsson, Ph.D., P.E.*
Engineering Design Research Laboratory
California Institute of Technology
Mail Code: 104-44
1200 East California Blvd.
Pasadena, California 91125
erik@design.caltech.edu

ABSTRACT

Initial results have been obtained for automatic synthesis of MEMS mask-layouts using a genetic algorithm. An initial random population of geometrically valid mask-layouts (non-self intersecting 2D polygons) is produced. The fabrication of each layout is simulated using a 3-D simulation of etching. The 3-D results of the fabrication simulation are compared to a desired 3-D shape specified by users. A genetic algorithm is applied to this initial population to iteratively search for a global optimum mask-layout whose fabricated 3-D shape is sufficiently close to the desired shape. During each iteration of the genetic algorithm, the mask-layouts with fabricated 3-D shapes close to the desired shape are more likely to survive. Random shape variations with ensured geometrical validity are applied to the surviving mask-layouts to produce the mask-layouts for the next iteration. The procedure is then repeated until one or more simulated shape is sufficiently close to the desired shape to stop the iteration. Initial results demonstrate the feasibility of this approach to mask-layout synthesis.

INTRODUCTION

We have developed an evolutionary approach to synthesis of mask-layouts for MEMS. Figure 1 illustrates the overall approach. The process begins with a desired 3-D shape, represented by a series of planar contours, with each contour parallel to the surface of the wafer. An initial random population of mask-layouts is produced. These mask-layouts are checked to ensure that they are geometrically valid (non-self intersecting, *etc*.). The fabrication of each geometrically valid layout is simulated using a computationally efficient geometrically accurate 3-D simulation of etching called Segs (Hubbard and Antonsson, 1996; Li, Hubbard, and Antonsson, 1998), built on earlier geometric (Hubbard and Antonsson, 1994) and cellular automata (Hubbard and Antonsson, 1997) methods. The 3-D results of the fabrication simulation are compared to the desired 3-D shape (by comparing individual contours using the turning functions of two polygons (Arkin, Chew, Huttenlocher, Kedem, and Mitchell, 1991)). Resulting 3-D shapes that are determined to be sufficiently close to the desired shape are kept in the candidate population. A genetic algorithm is applied to each member of the remaining population of mask-layouts to introduce random shape variations. The procedure is then repeated until one or more simulated shape is sufficiently close to the desired shape to stop the iteration.

Because reversing a fabrication process simulation (so that a 2-D mask-layout might be produced) appears not to be possible, and because each fabrication process would require (if possible) a reverse simulator to be developed, an approach using existing simulations of fabrication processes in an iterative refinement loop has been adopted, shown in Figure 1. One of the primary benefits of this approach is that *any* fabrication process can be utilized with the synthesis approach described here, as long as an efficient digital simulation of the process exists. Thus synthesis can be performed on devices to be fabricated from deposition, patterning and removal of surface layers by wet or dry

*corresponding author

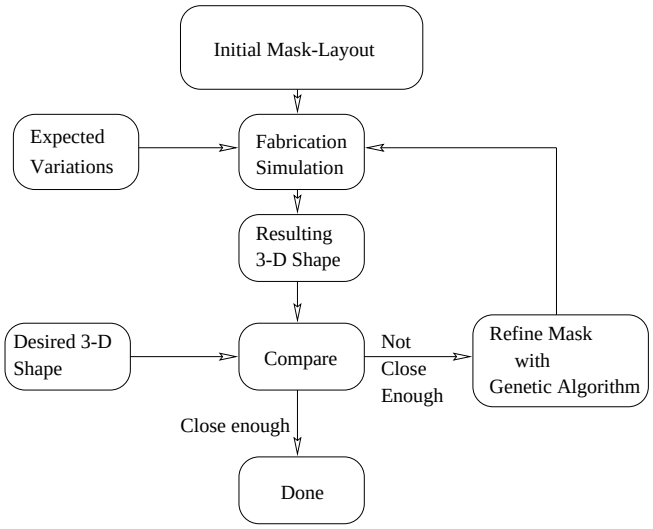

Figure 1. A SCHEMATIC REPRESENTATION OF AN EVOLUTIONARY MEMS SYNTHESIS TECHNIQUE.

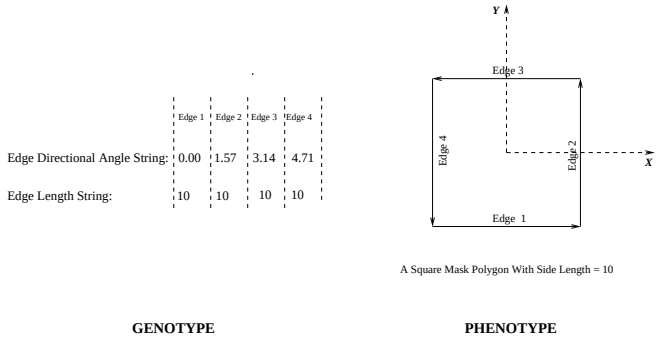

Figure 2. A SCHEMATIC ILLUSTRATION OF THE GA CODING SCHEME.

etching, or by bulk-wet etching, or any other processes for which an accurate digital simulator exists.

At present, we are using a polygonal representation for the mask-layout (derived from CIF). We re-encode this polygon into a form that can be modified with evolutionary methods of genetic algorithms utilizing the turning functions of polygons (Arkin et al., 1991) (each side of the polygon is represented as a length followed by an angle to turn to the next side). This has provided us with a representation that can be directly modified by stochastic optimization methods, such as genetic algorithms.

THE GENETIC ALGORITHM

Genetic algorithms (GAs) (one of the evolutionary techniques) adopt the idea of "survival-of-the-fittest" Darwinian principle to search the global optimum through selecting the observed best-suited individual. Typically, GAs encode each individual into a string of characters or digits along with some genetic operators. The original solution space is thus converted into an encoded space which becomes easier to search the global optimum. By analogy with genetics, the individuals in the original search space are referred as phenotypes and those in the encoded space are referred as genotypes. One feature of GAs compared to some other evolutionary techniques is that GAs maintain a population of individuals called a generation. A GAs' generation not only provides search robustness but also forms schemata which empower the search ability as illustrated through Holland's implicit parallelism (Holland, 1975). It has been found that GAs are particularly effective and powerful in exploring poorly understood spaces for optimization. A typical implementation of GAs starts from initial random seeded generation and then iteratively evolves the generations until some stopping criteria are met. During each iteration, the individuals will go through fitness evaluation and the genetic operations based on the fitness performance. The typical genetic operations are selection scheme, crossover and mutation (Goldberg, 1989).

We have developed a genetic algorithm package (GA) used for MEMS mask-layout synthesis design. It provides a complete set of hierarchic objects which can be used to build top-level genotypes and phenotypes with a binary or real coding scheme. One or more genetic operation functions such as rank-based selection function, *n*-ary tournament selection scheme and stochastic remainder without replacing selection function are applied to these objects (Goldberg, 1989). Both generational and steady-state, with or without elitist selection, can be easily implemented. The mask-layout synthesis method described here is built upon an iterative genetic algorithm. Currently, a simple sequential genetic algorithm has been used to iteratively synthesize mask-layouts with evolved 3D shapes closest to the target 3D shape.

Coding Scheme

The solution space in our problem is all simple 2D polygons which serve as mask candidates. The mask polygons are phenotypes in the GA (Goldberg, 1989). There are many ways to describe a 2D polygon. Here we choose edge length and edge directional angle (turning angle) to describe each edge of a polygon. A mask polygon is encoded into two real strings. One string contains edge directional angles and the other contains edge lengths. The size of each string is equal to the number of polygon sides. Two elements from each string with the same element position describe an edge of the polygon. These two real strings are genotypes in our GA (Goldberg, 1989). A schematic illustration of the coding scheme is shown in Figure 2.

This coding scheme directly captures the shape of mask the polygons, and can be directly used to measure the shape differ-

ence between any two mask polygons. When considering the fitness of candidate shapes, a shape difference (such as the turning functions (Arkin et al., 1991), Frechet or Hausdorff distances between polygons (Alt and Godau, 1995)) is a meaningful metric distance between the two candidate samples (mask polygons) in the search space. Such metric distances can be used to define the neighborhood of a point in search space as well as calculate the correlation length of the search space, which can further be used to tune different components of the GA (Manderick, de Weger, and Spiessens, 1991).

Crossover and Mutation

The genetic operators are applied to genotypes in the GA. As stated above, the genotypes in our problem are the two real strings of edge directional angles and edge lengths. Two genotypes are randomly paired. Simple crossover is separately applied to the pair of edge directional angle strings and the pair of edge length strings. For each string pair, a random site is chosen among the string bits and string elements right to the random site are exchanged. The two new children are generated and become the members of the next generation. Mutation is applied on a string element basis. If mutation is applied to a string element, the element value will be set as a real random number uniformly generated between a specified lower and upper bounds. Typically, crossover serves as the major genetic operator and the crossover rate is set 0.6. The usage of mutation is highly restricted by setting the mutation rate 0.005 and each time only one element (at most) of a string is allowed to be mutated.

Selection Scheme

In our genetic algorithm, the selection scheme is separated into the selection algorithm and the sampling algorithm (Grefenstette and Baker, 1989). The selection algorithm assigns a sampling rate to each individual in the generation. The sampling algorithm samples individuals from the current generation based on the assigned sampling rate of each individual. Different selection algorithms and sampling algorithms are provided. A selection scheme is the coupling of the chosen selection algorithm and chosen sampling algorithm. In our application, rank-based selection algorithm is chosen to assign the sampling rates based on individual's rank (Baker, 1985) through the following formula:

$$\text{Sampling rate}(i) = 1.0 + \text{bias} - 2 * \text{bias} * i/\text{size}$$

i	=	individual i
size	=	population size
bias	=	extra fitness awarded to the top-ranked individual compared to the average-ranked individual whose fitness is 1.0

In this way, the single bias variable can be used to control the selection pressure. Currently, the bias value is gradually increased as the genetic iteration proceeds to control the rate of convergence.

Stochastic universal sampling is chosen as the sampling algorithm to eliminate the sampling bias and reduce the sampling spread (Baker, 1987).

Fitness Evaluation

In each generation, the fitness value of each individual is evaluated in the following steps:

1. The 3D etching simulation (Segs (Hubbard and Antonsson, 1996)) is run on each candidate mask polygon. Each etching result is an evolved 3D shape represented by a stack of polygonal layers.
2. The pure shape mismatch value (explained below) and size mismatch value between each of the etching evolved 3D shapes and the specified target shape are calculated. The target shape is also represented as a stack of polygonal layers. The calculated pure shape mismatch value and size mismatch value are stored as the objective function values of each individual in the generation.
3. After obtaining the objective function values of all the individuals in the generation, the fitness values are calculated according to the following:

$$\text{Fitness}(i) = \frac{1}{\left(\frac{\text{pure shape mismatch}(i)}{\text{Average pure shape mismatch}} + \frac{\text{size mismatch}(i)}{\text{Average size mismatch}}\right)}$$

i	=	individual i
Average pure shape mismatch	=	average pure shape mismatch value of the generation
Average size mismatch	=	average size mismatch value of the generation

By taking the reciprocal of the combined normalized mismatch value, the good individuals with small mismatch values are spread by a larger degree according to fitness values. This can be helpful to increase the selection pressure during the final stage if a selection scheme sensitive to the magnitude of relative fitness values is chosen.

In the above fitness evaluation cycle, the mismatch between the evolved 3D shape and the specified target shape of each individual needs to be calculated. Both shapes are represented by a stack of polygonal layers. The etching simulation will ensure that the polygon layers of both shapes vertically match. Therefore, the mismatch of the two shapes becomes the weighted sum

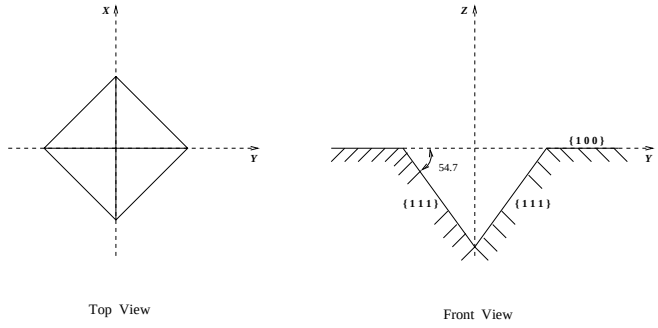

Figure 3. PYRAMIDAL TARGET SHAPE WITH $\langle 111 \rangle$ SIDEWALLS.

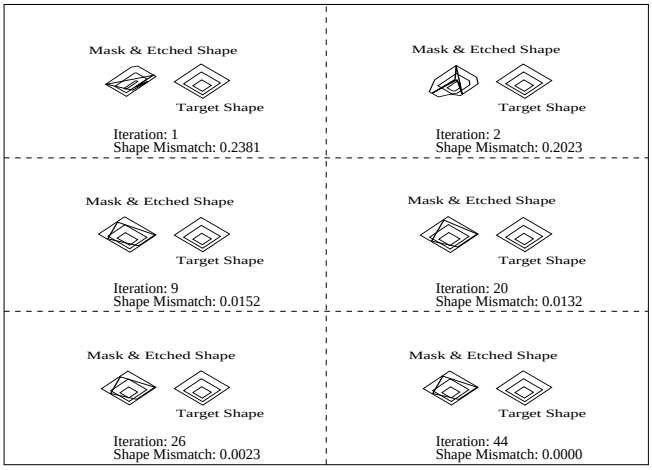

Figure 4. GENETIC ALGORITHM SYNTHESIS RESULTS FOR PYRAMIDAL TARGET SHAPE.

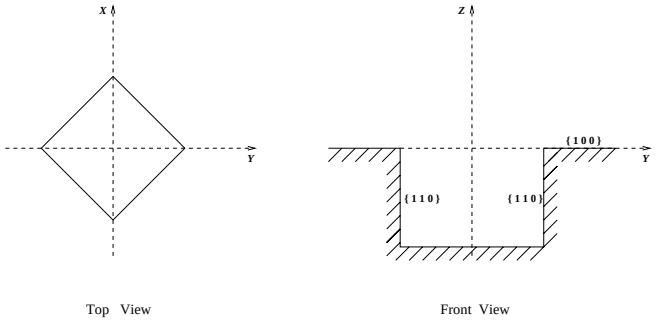

Figure 5. RECTANGULAR TARGET SHAPE WITH $\langle 110 \rangle$ SIDEWALLS.

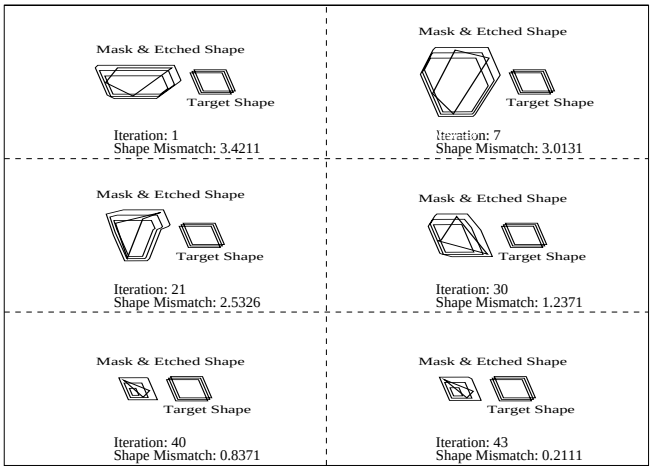

Figure 6. GENETIC ALGORITHM SYNTHESIS RESULTS FOR RECTANGULAR TARGET SHAPE.

of the mismatch between the two polygon layers in each vertical level. The weights are introduced to concentrate the major effort of the genetic algorithm on certain polygon layers of the target shape which may carry the major function of that shape. The mismatch between two polygon layers can further be decomposed into the pure shape mismatch and the size mismatch of two polygons. The size mismatch is simply the length difference of the two polygons. The pure shape mismatch are obtained by the method introduced in (Arkin et al., 1991).

The method calculates the minimum difference between the accumulated turning angles of each edge weighted by the scaled length of that edge. Arkin's method, however, is both translation and rotation invariant. Here in our problem, the rotation invariance needs to be removed because the etching mechanism is sensitive to the absolute mask edge orientation. So a small adjustment was made to obtain translation invariance only.

EXAMPLES

Our initial interest was to test the feasibility of using a genetic algorithm to iteratively synthesize the mask-layout for some simple desired MEMS devices. The current goal is to find a mask which produces a bulk-etched 3D shape which matches a specified target shape as closely as possible.

Our first test was a target shape with a 3D pit with four slanted sidewalls whose normals oriented along $\langle 111 \rangle$ crystal directions (shown in Figure 3) in a $\langle 100 \rangle$ wafer. For anisotropic wet-etching of a pit shape on $\langle 100 \rangle$ wafer (*e.g.*, using EDP type S at 100°C (Seidel, Csepregi, Heuberger, and Baumgartel, 1990)), it is known that $\{111\}$ crystal planes are the dominant planes because of their slowest etching rates. So by choosing long enough etching times, we can expect a large set of 2D masks to be evolved to our target shape. This means the solution space is large enough for the genetic algorithm to easily find global optimum. We ran several tests with population sizes varies from 20

to 100. The crossover rate was set as 0.6 and mutation rate is 0.033. All of them found global optimum within 50 iterations.

Example results, showing the closest simulated 3-D shape at six different times during the synthesis, is shown in an oblique view in Figure 4. The dark polygon, on the left-hand pair of shapes at each iteration, is the candidate mask-layout shape. The nested polygons, on the left at each iteration, are contours at three depths of the simulated 3-D shape. The stacked square polygons, on the right at each iteration, are contours at three depths of the desired shape. By iteration 44, the resulting shape matches the target shape.

The next test used a target shape that is the same as the one above except the four sidewalls are vertical (shown in Figure 5). The sidewalls are $\{110\}$ crystal planes. Note in this case, there is a conflict between the lateral polygon shape and the vertical wall formation. If the edge normals of lateral polygons are perfectly aligned with $\langle 111 \rangle$ directions, the sidewalls are dominated by $\{111\}$ planes and the $\{110\}$ sidewalls can never be reached. We ran the genetic algorithm to see how it evolves an optimum mask that compromises the lateral polygon shape with the vertical sidewall formation.

Figure 6 shows a result from the genetic algorithm. The tradeoff is clearly shown through iteration 40 and 43. In iteration 40, the shapes of all etched lateral polygon layers are very close to those of target shape but the sidewall is slanted off heavily. In iteration 43, the bottom lateral polygon size is increased and the top lateral polygon size is decreased with more nearly vertical sidewalls while the top lateral polygon corners are chopped off. This example points out that some desired shapes can only be approximated, with a single fabrication process.

CONCLUSIONS

Thus far we have completed the iterative mask-layout synthesis loop by combining the genetic algorithm and forward etching simulation. We have also completed the initial testing stage, which gave us reasonable and promising results. During the initial testing stage, we found that the genetic algorithm does not work well in cases where the target geometric shapes are complicated. This is likely caused by the weak power of the crossover operator (simple crossover) that we are using at current stage. A more intuitive encoding combined with heuristic crossover operator as well as efficient valid mask polygon random generator is being developed now to increase the search capability of the genetic algorithm. We also plan to develop an approach to more optimally balance the exploration and exploitation effort of the genetic algorithm to raise the efficiency of the searching.

Current work is extending this genetic algorithm approach to multiple processes, thus significantly expanding the range of shapes that can be synthesized exactly, or with a close approximation. As an additional extension of these methods, we plan to introduce expected variations into the fabrication simulation (*e.g.*, mask misalignment, process variations, *etc.*). By doing so the stochastic optimization will produce mask-layouts and process sequences that are least sensitive to these variations. In this way, robust designs will be synthesized.

REFERENCES

Alt, H., and Godau, M., 1995, "Computing the Frechet Distance Between Two Polygonal Curves," *International Journal of Computational Geometry and Applications*, Vol. 5, (No. 1 and 2), pp. 75–91.

Arkin, E. M., Chew, L. P., Huttenlocher, D. P., Kedem, K., and Mitchell, J. S. B., 1991, "An Efficiently Computable Metric for Comparing Polygonal Shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 13, pp. 209–215.

Baker, J. E., 1985, "Adaptive Selection Methods for Genetic Algorithm," In Grefenstette, J. ed., *Proceedings of the International Conference on Genetic Algorithms and Their Applications*, pp. 101–111.

Baker, J. E., 1987, "Reducing Bias and Inefficiency in the Selection Algorithm," In *The Second International Conference on Genetic Algorithms.*

Goldberg, D. E., 1989, *Genetic Algorithms in Search, Optimization, and Machine Learning* Addison-Wesley.

Grefenstette, J. J., and Baker, J. E., 1989, "How Genetic Algorithms Work: A Critical Look at Implicit Parallelism," In *The Third International Conference on Genetic Algorithms*.

Holland, J. H., 1975, *Adaptation in Natural and Artificial Systems* The University of Michigan Press.

Hubbard, T. J., and Antonsson, E. K., 1994, "Emergent Faces in Crystal Etching," *Journal of Microelectomechanical Systems*, Vol. 3, (No. 1), pp. 19–28.

Hubbard, T. J., and Antonsson, E. K., 1996, "Design of MEMS via Efficient Simulation of Fabrication," In *Design for Manufacturing Conference*. ASME.

Hubbard, T. J., and Antonsson, E. K., 1997, "Cellular Automata in MEMS Design," *Sensors and Materials*, Vol. 9, (No. 7), pp. 437–448.

Li, G., Hubbard, T., and Antonsson, E. K., 1998, "SEGS: On-line Etch Simulator," In *MSM'98, Modeling and Simulation of Microsystems, Semiconductors, Sensors and Actuators* Santa Clara, CA. IEEE.

Manderick, B., de Weger, M., and Spiessens, P., 1991, "The Genetic Algorithm and the Structure of the Fitness Landscape," In *The Fourth International Conference on Genetic Algorithms*.

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