3D MEMS Design via Matlab Interactive Plots

Nanping R. Lo and Kristofer S.J. Pister*

*Department of EECS, 497 Cory Hall, UC Berkeley, Berkeley, CA 94720
Electrical Engineering Department, UCLA, 405 Hilgard Avenue, Los Angeles, CA 90095
E-mail: nanping@ee.ucla.edu

Several commercially available MEMS CAD packages perform excellent rendering of microstructures. However, since 3D display is not the primary goal of these packages, other than providing views from different perspectives, the renderings from these MEMS CAD packages are static and do not permit interaction with parts within the design. The previous works by the authors attempted to address the interactivity issue with custom software and Java/VRML solution [1]. It was not widely used because it was viewed primarily as a visualization tool and there were installation issues. In this paper, the authors extend the solution for interactive 3D MEMS design by using Matlab as the underlying analysis and displaying engine. With it, the designers can fold structures away from substrate directly from their 2D, flat CIF design. Raytracing, for MEOM design, and collision detection, for micro robotic design, have already been incorporated in this toolkit.

Figure 1 outlines the process flow for using Matlab's 3D plot for interactive MEMS design. A preprocessor processes the standard CIF files by adding the process dependent topology information. The topology is obtained by performing boolean operations on layers within the design. Once the layers in the design have acquired the height information, all spatially jointed boxes are sorted into corresponding structures. If the interconnects between structures, i.e. hinges, are drawn in a specific fashion, the preprocessor can detect the interconnection between structures. Thus, although all structures are spatially disjointed, linkage information (connected via hinges) between structures, as well as rotational data can be generated. Furthermore, if the specified pseudo layer is used in the design, the preprocessor will treat it as reflecting surfaces for raytracing. The topologized design together with all the information gathered by the preprocessor is written to an intermediate text data file.

The boxes in the intermediate data file are read into Matlab by specifying the vertices of Matlab graph object called "patch". It is Matlab's underlying plotting functions that render all the patches as 3D plot. Manipulation and movement of structures, such as rotation and translation, are done by applying the spatial delta to the vertices of all the patches involved. After each movement, Matlab's plotting engine updates the plot accordingly. By entering commands in the Matlab console window, or submitting an m-script file, a series of movements can be done such that the design appears to undergo animation. A set of commands exist to do the following tasks:

- identify, manipulate and view structures
- get and set linkage and rotational information (if not already extracted by the preprocessor)
- specify and commence raytracing parameters
- commence collision detection

Figures 2 thru 4 show a series of plots and a SEM of a XYZ micro-optical stage [2], from its 2D design, to the assembled device with raytracing to illustrate its functionality. Within Figure 5 is a SEM and plots of a micro mirror [3]. Figure 6 shows a hollow triangular beam linkage in 2D design and the linkage and rotational information automatically extracted for assembly. With a rich set of tools available in Matlab, additional MEMS analysis tools can be added on top of this toolkit. Thus perhaps one day enable the rapid design and analysis necessary to bring the conceptual bug-on-a-chip design, as seen in Figure 7, to life.

The authors invite the readers to visit the Mat3d website, http://www.ee.ucla.edu/~nanping/mat3d, where more information and plots of Mat3d, its on-line documentation, as well as the package itself can be obtained.

Application Area: CAD/CAE/CAM

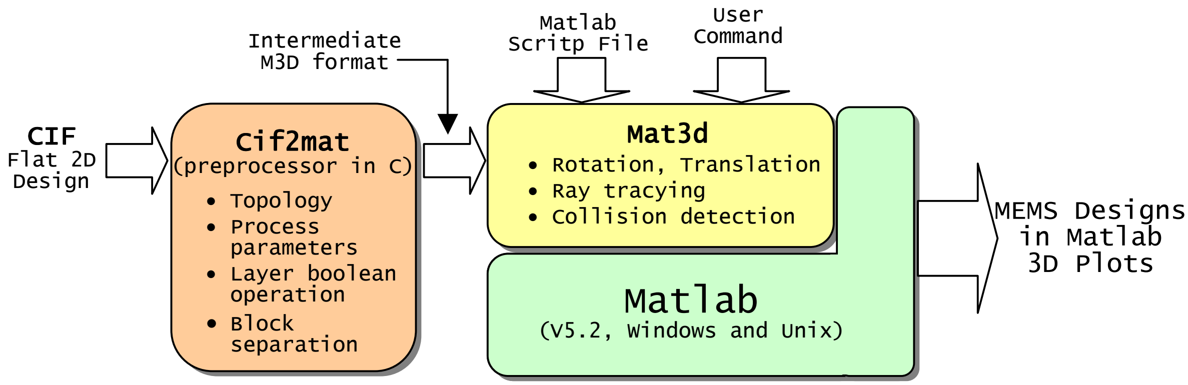

Figure 1. Process flow of plotting 3D MEMS designs with Matlab.

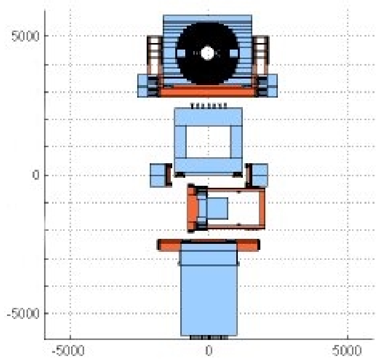

Figure 2. Plot of the 2D flat design of the XYZ-stage.

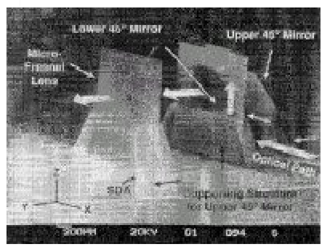

Figure 3. SEM of the XYZ-stage.

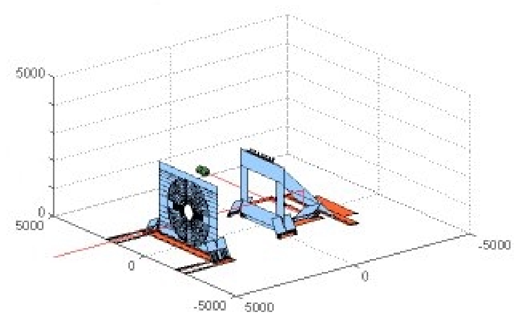

Figure 4. Plot of the assembled XYZ-stage with raytracing demonstrating its functionality.

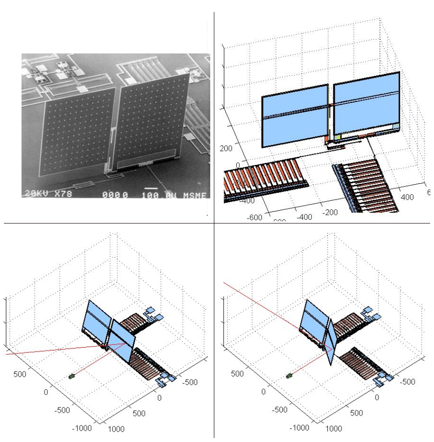

Figure 5. SEM and 3D plots of an assembled micro mirror (top). Raytracing to illustrate operation (bottom)

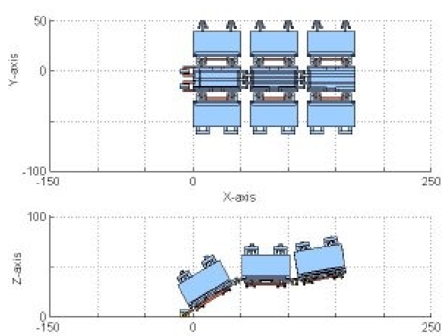

Figure 6. Plots of 2D design (top) in xy-view, and the assembled linkages (bottom) in xz-view.

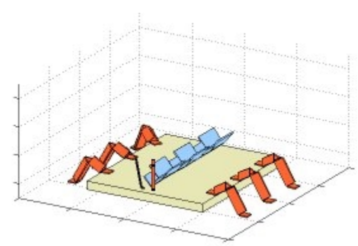

Figure 7. A plot of a conceptual design of a micro-bug on a chip.

[1] N. Lo and K.S.J. Pister, "*3DµV - A MEMS 3-D Visualization Package*", SPIE, 1995.
[2] L. Y. Lin, J. L. Shen, S. S. Lee, and M. C. Wu "Surface-Micromachined Micro-XYZ Stages for Free-Space Micro-Optical Bench", IEEE Photonics Technology Letter, Vol. 9, No. 3, March, 1997.
[3] Matt Last, K.S.J. Pister, "2-DOF Actuated Micromirror Design for Large DC Deflection", MOEMS '99.