Computational Evaluation of Three-Dimensional Topography Process Simulation Components

R.H. Wang, M.S. Karasick*, A.R. Neureuther
University of California, Berkeley
IBM Corporation*

This paper describes the use of centralized software services to implement topography process models, and evaluates their computational impact on three-dimensional topography process simulation. Of interest is the effectivness in computation time with which typical topography process simulation functions can be performed via application-specific interfaces to a solid modeller. The work described in this paper is carried out on the IBM Geometry Engine [1], a cell-complex solid modeller in C++.

Topography process simulation involves algorithms which assemble physical information from wafer topography and deform the wafer surface. Capabilities such as point-material classification, vertex- and edge-neighborhood access, and visibility solid angle calculation need to be efficiently supported by the underlying geometric representation. Numerical algorithms which deform the surface locally must work in conjunction with geometric operations which remove self-intersections in the deformed surface. This paper considers prototype examples of these operations and groups them as *basic*, *surface field support*, and *surface deformation support.*

The *basic* tests are designed to identify the properties of a solid modeller which fundamentally affect its effectiveness to support topography process simulation. One such test is the *staircase* test, shown in Figure 1, which benchmarks boundary interrogation capabilities. Figure 1 shows that requesting connectivity information can be orders of magnitude more expensive than requesting component information. Additional tests will also be described which verify the robustness of the set operations, characterize the effects of granularity on set operation performance, and identify use of auxilary data structures which reduce the number of objects considered during intersection tests.

The *surface field support* tests are used to characterize the efficiency of geometric queries used in surface deformation and surface diffusion calculations. Algorithms for (1) point-material classification, (2) vertex neighborhood access, and (3) visible solid angle calculation have been implemented. For a 1um x 1um silicon trench discretized into 500 0.1um x 0.1um square patches, on the average, algorithm (1) requires 0.50 CPU seconds per point and algorithm (2) requires 0.04 CPU seconds per surface vertex per neighbor. *Solid angle* tests, which intersects the solid model with cones formed using surface vertices and rim of the hemispherical source, are used to mimic visible solid angle calculations. For the trench structure, the *solid angle* test requires 9.89 and 24.21 seconds per surface vertex, for rim discretization of 4 and 40 points respectively.

Surface deformation support involves efficient use of boolean set operations in simulating surface deformation. Figure 2 illustrates a new algorithm which reduces the number of swept volumes to be merged at each time step by decomposing the wafer surface into monotone surface patches. The algorithm initializes a solid angle of monotone directions (hemisphere in 3D, half-circle in 2D) based on the first surface element (triangles in 3D, edges in 2D), and reduces the solid angle as new elements are accumulated into a monotone surface patch. Conformal trench deposition, as depicted in Figure 3, was simulated in two-dimensions using edge-sweep and monotone decomposition. Figure 4 shows that even in two-dimensional simulation, monotone decomposition significantly reduces the number of swept volumes, which results in significant reduction of set operation time. For three-dimensional simulation, the use of monotone decomposition is even more critical, since the ratio of the number of facet-sweep volumes to the number of monotone polyhedra is much larger.

In summary, cell-complex solid modellers offer convenience and robustness and are useful for topologically complex cases, especially when constructs such as monotonicity are used to reduce the number of set operations.

[1] M. Karasick, D. Lieber, ACM Symp. on CAD and Found. of Geom. Modelling, June 1991, pp. 15-25.

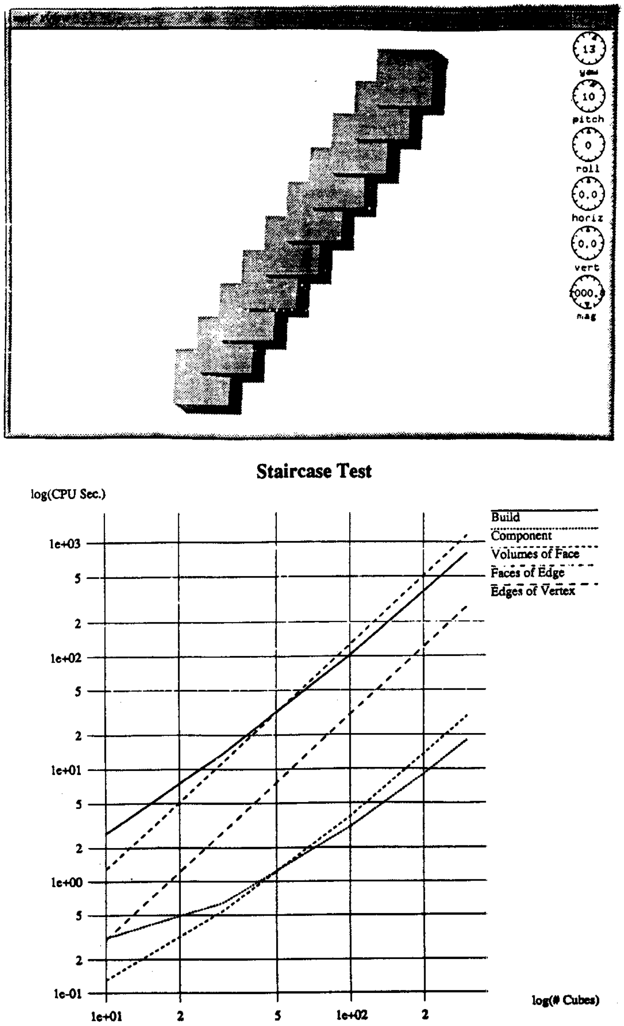

Figure 1. Staircase test

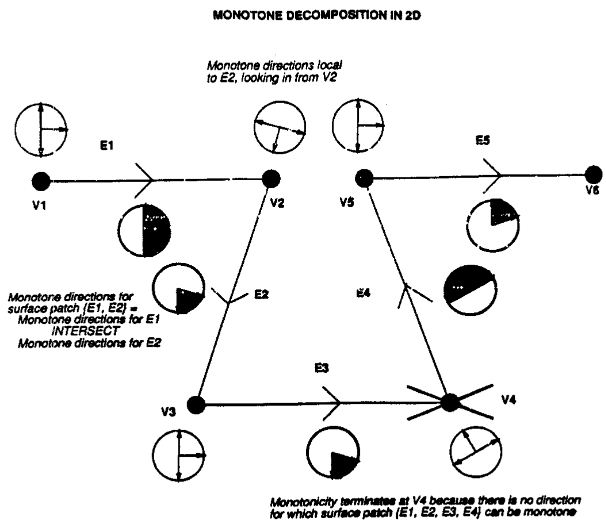

Figure 2. Monotone decomposition in two dimensions

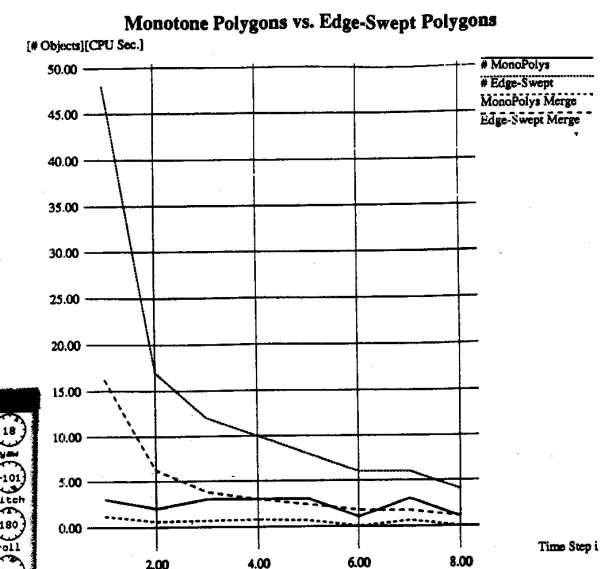

Figure 4. Monotone decomposition vs. edge-sweep

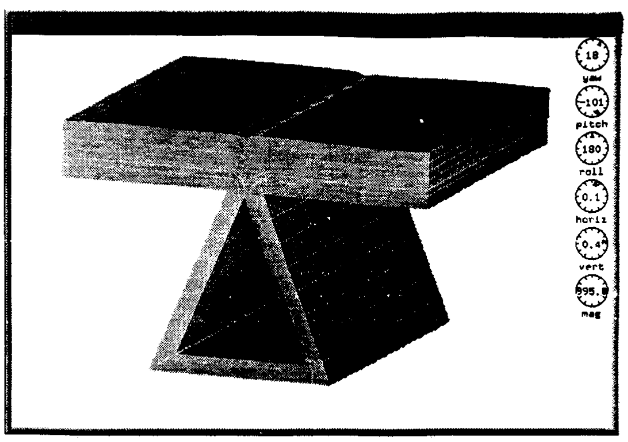

Figure 3. Conformal trench deposition