A – Exact vs. Approximate Predicates

Many computational geometry algorithms make decisions based on geometric tests (i.e., predicates) such as the 3D orientation predicate to determine whether a point is above, below or on a given plane. Such tests may produce incorrect answers because of round-off errors that arise due to finite-precision arithmetic. Representing and computing all geometric information by exact arithmetic can avoid such instability in the geometric predicates. However, the use of exact arithmetic methods results in a high memory and running time overhead, as compared to finite precision arithmetic, e.g. the widely used IEEE 754 floating-point standard.

An alternative strategy is to employ exact arithmetic only when it is needed. Shewchuk presented four robust geometric predicates for 2D and 3D orientation and incircle tests by using this strategy [1]. Another similar approach is the lazy evaluation scheme [2] that is used in CGAL library [3]. However, the numerical errors are introduced not only by such predicates but also by the procedures used to compute the coordinates of a new object from the coordinates of existing objects – called constructions. Round-off errors can only be completely eliminated when all the intermediate and final coordinates are represented using arbitrary-precision arithmetic. However, use of arbitrary precision arithmetic can be expensive and result in high memory overhead.

B – Point-based vs. Plane-based Representation

There are two kinds of representations used for vertex coordinates in geometric modeling: point-based and plane-based. Most approaches employ point-based representation, where the coordinates of vertices in R^3 are represented by vectors with three scalars. In plane-based representations, the coordinates of every vertex are represented using three intersecting planes. Point-based representations tend to have more problems with robustness due to numerical inaccuracies, but have a lower memory overhead and involve fewer arithmetic operations (ref. [4]).

The recent work of Bernstein and Fussell [5] is based on the earlier work of Sugihara and Iri [6] that uses plane-based representations and avoids constructions when computing Boolean operations on polyhedra. Four numerical predicates similar to the ones proposed by Banerjee and Rossignac [7] were used in [5]. The use of such predicates results in higher number of arithmetic operations as opposed to point-based representations. In practice, plane-based representations are not sufficient to generate exact results by themselves. The plane-based representations must be used together with exact predicates to generate correct results [5] – i.e., extended precision is required for the intermediate computation. When the required number of bits exceeds the hardware support (e.g., 64 bits), the resulting computations could become expensive. For example, when the computation of every predicate must perform exact arithmetic using higher precision, the time for the orientation predicates can be even more than 1,000 times longer than using IEEE double precision arithmetic (as shown in Table 1) if the optimization technique proposed in [1] is not used. The statistics shown in Table 1 are based on an implementation that includes 1) a single stage filter to detect the reliability of numerical predicates using approximate arithmetic and 2) applying arbitrary-precision arithmetic in the second stage to evaluate the exact results of numerical predicates.

References

[1] J.R. Shewchuk, “Adaptive precision floating-point arithmetic and fast robust geometric predicates,” Discrete & Computational Geometry, vol. 18, pp. 305–363, 1997.

[2] S. Pion and A. Fabri, “A generic lazy evaluation scheme for exact geometric computations,” in 2nd Library-Centric Software Design, 2006, pp. 75–84.

[3] CGAL, CGAL - Computational Geometry Algorithms Library, http://www.cgal.org/, 2010.

[4] C.M. Hoffmann, “Robustness in geometric computations,” ASME Journal of Computing and Information Science in Engineering, vol. 1, pp. 143–156, 2001.

[5] G. Bernstein and D. Fussell, “Fast, exact, linear Booleans,” Computer Graphics Forum, vol. 28, no. 5, pp. 1269–1278, 2009.

[6] K. Sugihara and M. Iri, “A solid modelling system free from topological inconsistency,” Journal of Information Processing, vol. 12, no. 4, pp. 380–393, 1989.

[7] R. Banerjee and J. Rossignac, “Topologically exact evaluation of polyhedra defined in CSG with loose primitives,” Computer Graphics Forum, vol. 15, no. 4, pp. 205–217, 1996.