Exactly computing map overlays using rational numbers

W. Randolph Franklin

ECSE Dept, 6026 JEC, Rensselaer Polytechnic Institute, 110 8th St, Troy NY 12180 USA

mail@wrfranklin.org, http://wrfranklin.org/

Keywords: overlay, roundoff error, rational numbers, topology, intersection

We present a solution to overlaying two maps without any roundoff errors and the resulting topological impossibilities. From two input maps (planar graphs) containing polygons separated by polyline edges, the overlay is a map, each of whose polygons is the intersection of a polygon from each input map. Roundoff errors caused by the finite precision of floating point numbers are a major problem since they cause topologically impossible results. This gets worse when the inputs are bigger, and when input edges almost coincide. Various heuristics help, only up to a point. We completely avoid roundoff error by representing coordinates as rational numbers, or vulgar fractions composed of a numerator divided by a denominator. Because the result of adding or multiplying two rationals has about the total number of digits of both inputs combined, the numerator and denominator are stored as arrays of groups of digits. Using rational numbers, lines intersect exactly; slivers are no longer a concern. Although rational numbers are well known to computer scientists, applying them here has challenges. First, the execution time can increase by a factor of a hundred; the space also grows considerably. Our solution is a very efficient map algorithm, taking expected linear time in the input size. We use a uniform grid to find all intersections of edges, and then classify the intersections and fit the pieces together. Second, rationals cannot be used for some operations, such as computing polygon areas. Third, this application is large enough to stress the off-the-shelf rational number packages, so that development is required here. Our current implementation, in progress, can now process small maps. Our algorithm and data structures are regular enough to be amenable to future parallelization, whether with OpenMP or CUDA.

This research was partially supported by NSF grant IIS-1117277.