

Spatial Objects

With the expanding use of spatial information comes a growing need to interchange this information. Parallel runs the development of spatial databases serving multiple applications. Both have highlighted the need for consistency of representation and behaviour of spatial objects across computing platforms.<P>

Until recently even such a supposedly well known technique as representing area features as polygons was fraught with pitfalls; there have been countless forms of 'standard' representations in use. The work of ISO TC211 attempted stringently to address this problem by developing a well-defined set of specifications and nomenclature for the representation of spatial objects. This has been supported by the Open Geospatial Consortium (OGC), which has adopted the ISO standards and extended them. Not so well addressed has been robust representation and behaviour of spatial objects. ISO 19107:2003, the standard on Geographic Information, Spatial Schema, is currently under review, and an excellent opportunity now arises for ISO TC211 to address these issues.

Equality

One of the most basic relationships between two spatial object representations is equality: answering the question, is A equal to B? Database technology assumes the presence of a well-defined and repeatable test for equality, and many indexing algorithms require the existence of such a predicate. It is thus surprising that equality is a complex and largely unsolved issue. Three forms of equality exist, all valid and necessary but not all readily computed. The first requires identical representation of spatial objects. For example, two simple polygons are equal if they have the same number of vertices at exactly the same points on each, the starting point is the same in both, outer and inner rings are similarly organised, and the orientation of the corresponding rings is equal. The second form relaxes these requirements, allowing that the representations may differ as long as both define exactly the same set of points. The third form recognises that point locations may be differently represented and allows deviation in location of all points within a certain tolerance. This is the form defined by the ISO and OGC specifications, but the details of tolerance and its method of application are not made explicit in the specifications.

Validity

The question of validity causes most difficulties in exchanging data or providing a shared repository. Any large and complex set of spatial object representations exported from one system probably contains one or more geometric constructions that will be seen by another as invalid and rejected. This is because no agreed definition for validity exists. ISO and OGC specifications define concepts such as `isSimple`, but it is up to the implementers to decide if validity requires simplicity. For example, one database implementation (reasonably) requires that the boundary of a polygon be simple, but that a linestring need not be so. In a perfect world, we would have well defined spatial Abstract Data Types (ADT). This implies (1) a standard allowing representation and interchange of data such that equality and validity is preserved, and (2) a consistent set of functions, such as union and intersection, and predicates, such as `connectedTo`, `includedIn`, and `intersects` that give predictable results on all platforms.

Possible Solutions

One solution would be to mandate the exact storage form to be used; for example, in analogy with the IEEE floating-point arithmetic specifications, 8-byte floating point could be required for all point coordinate values and use of specified tolerance values. This would also require that the algorithms used to evaluate all functions be specified, especially those for validity and equality. But this approach still falls short, since the calculation of some functions can cause rounding that affects later functions, so that the sequence of applying functions needs also to be specified. Thus, to ensure a repeatable result, this approach requires not only specification of storage form but also of tolerance, algorithms and order of execution of operations. An alternative is to avoid any rounding and disallow any tolerance in the evaluation of all functions. This can be achieved by the use of integer-based homogenous coordinates. Rather than a pair (in 2D) of floating-point numbers to represent a point (x, y), this approach uses a triple of integers (X,Y,Q) where the point is interpreted as being at a location (X/Q, Y/Q). This representation allows all the usual functions to be calculated without any rounding errors, and is completely rigorous and repeatable.

Regular Polytope

A third approach is the Regular Polytope, which uses a dual of the homogenous coordinates approach, defining all objects (in 3D) in terms of a set of 'half spaces' (see textbox). In contrast to the homogenous-coordinates approach, these are represented in the computer systems as finite precision integers. This requires an approximation when data is loaded into the form. However, after this step all the usual functions can be evaluated without rounding. The evaluation does require higher than usual precision of calculation, but the precision requirement is limited. The Regular Polytope Representation is a boundary-free representation: any point must be within or outside any given regular polytope. The direct positions of vertices have no role in definition of the representation. This could result in new ways of representing spatial objects with improved potential for performing validity and equality tests, avoiding problems in data exchange of spatial data.

Further Reading

u Thompson, R.J., 2007, Towards a Rigorous Logic for Spatial Data Representation, in Publications in Geodesy 65, Nederlandse Commissie voor Geodesie ISBN 978 90 6132 303 7. ? u van Oosterom, P., Quak, W., Tijssen, T., 2004, About Invalid, Valid and Clean Polygons, in Developments In Spatial Data Handling. P. F. Fisher. New York, Springer-Verlag:1-16. ? u Mehlhorn, K., Näher, S., 1999, LEDA: A Platform for Combinatorial and Geometric Computing, Cambridge University Press. ? u Randell, D. A., Cui, Z., Cohn, A. G., 1992, A Spatial Logic Based on Regions and Connection, 3rd International Conference on Principles of Knowledge Representation and Reasoning, Cambridge MA, USA, Morgan Kaufmann. ? u Kazar, B.M., Kothuri, R., van Oosterom, P., Ravara, S., 2008, On Valid and

Half-space

A half-space is defined as the set of points (x,y,z) on one side of a plane surface, with rules defining whether points exactly on the plane are included or not. It allows definition of a convex object as the intersection of a number of half-spaces. Any convex polyhedron with planar faces can be defined in this way, and is known as a 'convex polytope', but defining more general objects requires definition of a 'regular polytope' as the union of a number of convex polytopes. This definition allows the definition of geometric objects that rigorously exhibit the behaviour of regions in the Region Connection Calculus and, in particular, have a well-defined 'equal' test of each of the three kinds. The main reason for validity checking in a database management system is to ensure that users of the data are not presented with degenerate or special cases that cause programming difficulties. For example, if adjacent points along a linestring are allowed to have the same direct position, a program attempting to calculate the bearing of the line between them may fail. This kind of checking is unnecessary in the case of the regular polytope. By contrast, in the more conventional approach of defining a spatial object in terms of its boundary (by points, lines, and surfaces), the question of validity in 3D is very complex.