

Prevention of "Error Overlaying Objects" : the POLYNODE MapBasic application

Jacques Paris

October 1998

EOO (Error Overlaying Objects) : generalities

"Error Overlaying Objects" have a tendency to appear during some operations involving Object Manipulating Functions in MI versions up and including 5.0. That functional deficiency should be corrected in a future release but until then, and for those that will continue using previous versions, the problem will remain.

EOO occur on data sets that seem to be topologically sound. That means that the polygons of these sets are not only exempt of spikes, pseudonodes, bowties ... that are strict topological considerations, but, also, that they do not contain marginal situations such as "Closed-C and "Eight" regions, be they multi- or single polygon regions.

Regions that do not satisfy the strict conditions are often revealed during simple display operations (such as thematic mapping) but MI can handle marginal situations not only for display but also for some

simple OMF such as "Combine", failing more systematically with an "Intersect".

A Closed-C region is formed by two polygons, one internal to the other, having a single contact point. In an Eight region, the polygons are external to each other and have one common node. The single polygon equivalent has exactly the same shape but it is drawn as a continuous line that does not cross over itself (it would be then a classical "internal hernia" or a bowtie).

Strict adherence to topological soundness can be achieved with the use of several tools. MAPCHECK.mbx is certainly very effective to detect "marginal" situations, even though his messages are rather cryptic and need to be interpreted "on the map" and his point map output is useless because very few of the occurrences detected as messages are transformed in points.

The notion of potential problem situation

But despite all these checks and corrections, EOO may still occur during OMF activity. The best explanation I came up with, that has rather been supported than challenged by many, is that EOO may occur if more than a given number of polygons abut at any given node.

That rule must be qualified with the position of the node either on the perimeter (node partially surrounded by polygons) or on the inside (node totally surrounded by polygons). And the notion of polygon must be used, not that of region, which implies "exploding" multi-polygon regions in single polygons.

The maximum "safe" number of polygons is thus 3 for an inside node, and 2 for a perimeter node.

The explanation is simple. Let us imagine an inside node with 4 abutting polygons. During an OMF operation, there might be some combining done before overlaying; this would happen in "MI black box" and would not be seen by the user. If the combine operation involves 2 adjacent polygons among the 4 implied, there would be no problem, but if the 2 are opposite to each other, the combine MAY create a Closed-C or an Eight polygon, depending on the remaining geography of these objects. And that is a break in the marginal

Using POLYNODE : General procedure

The only operation the user need to undertake when the application has been loaded, is to select the table to be processed. There are no constraints on that table because every region it contains is transferred to a work table, as such if it is a single polygon region, as a series of "exploded" polygons for a multi-polygon one. Nodes are

perimeter	normal	values -1 and -2
	problem	values <-2
inside	normal	values 2 and 3
	problem	values >3

TEMPOY is automatically displayed in a mapper with the original table open as ORI.

Using POLYNODE : Potential uses

POLYNODE is most effective if all strict and marginal conditions are respected. Results can be misleading when they are not.

topological rules MI follows. OEE may not result automatically, but it can, and if it does, that reason may be the source of it.

Prevention of these situations is a meticulous process and the work it represents will have to be justified by the expected results. Besides, the adjustments to be made are only valid for the original set of objects and if these objects are transformed in some ways (regroupings or divisions), new problematic situations might have been created, and the entire operation has to be done again on the new set.

then extracted, avoiding multiple occurrences, and values calculated representing the numbers of polygons at each nodes.

The final outcome is a table called "TEMPOY" of points (one for each node) displayed as 4 different symbols :

small open black circles
small red filled circles
black asterisks
thick red crosses

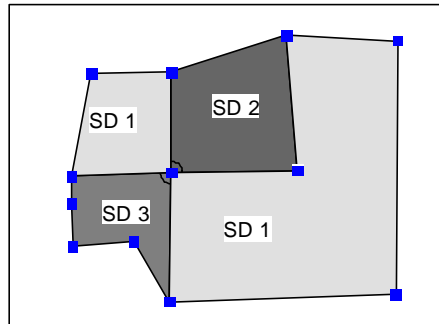
The user can save the table as such under a different name, make any selection he wants and save the selection, or close it without any save.

Problem nodes are easily detected by inspection of symbols or selections on "problem" values.

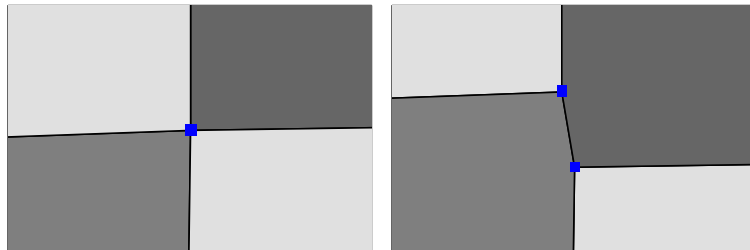
Selection of negative values provides the outline of the perimeter; presence of such points in the theoretical inside reveals the existence of slivers between polygons (missing node on one of the

adjacent sides of two polygons) if such gaps have not been detected by previous topology checking applications. Those gaps must be corrected if they are not intentional.

Correcting problem nodes



This example reveals that 4 polygons belonging to 3 regions meet at one node. It is a special case that should not happen because SD1 is an Eight region. We will see how to "repair" an Eight and, generally speaking, any "problem" node



The polygons have to be transformed in such a way there is no common node with more than 3 polygons attached to it. As a further constraint, for a Closed-C or Eight "repair", the two polygons of the same region must be isolated from each other. The following diagrams show the situation "before" (left) and "after" (after) for an Eight, but the principle is the same for any situation.

To do it, one must "Reshape" one polygon by adding a new node and adjust all the polygons to these two nodes, adding and moving nodes where needed.

Distortions created by the adjustments.

The "new" polygons are slightly different from the original ones. Are these distortions such that the integrity of the geographic objects is altered?

It is true that in many cases some boundaries should meet at exactly the same point. It is easy to imagine a grid made of square cells placed in a neat row/column arrangement. But it is also true that these conditions may exist rather rarely in reality and that most cases may be generated by a over-simplified digitizing procedure.

Judging the impact of the distortions can be done only by taking into account the original geographic precision of the location of nodes in the real space and the extent of the distortion relatively to that location precision. The location precision is independent of the MI user, it is built in the definition of the map objects themselves, but the actual displacement of the replicated node is under the control of the user and can be limited to the internal precision of the map stored in MapInfo.

For a general discussion on Internal Precision, see the INTPREC2.zip file on our web site (<http://www.total.net/~rparis>) or the chapter with that name in "My (MI) Book o'Tricks" (table of contents and order form also on the web site).

Minimum distance between two distinct points as stored in a MI table measures the internal precision for that table. As x and y are stored individually, there are two measures, for longitude, for latitude. In a lon/lat table, that distance correspond to $1/1\,000\,000$ of a degree which is about 11cms. This measure is good along the y (latitude) axis, but for the longitude it must be multiplied by the cosine of the latitude at the given location. For a projected table, the values for the x and y directions are calculated as $(\text{bound maxi} - \text{bound mini})/2\,000\,000$ in the map units for each direction.

Therefore, actual distortions in a lon/lat could be as small as $11 \cdot \cos(\text{latitude})$ cms, while ground location precision may be of the order of the meter, more frequently higher than that. In such conditions, distortions brought in by these corrections will not seriously alter the table. But it is always better to check each situation before deciding on altering the table.

Users must also be aware that working at or near the internal precision level can sometimes be tricky. MI graphic editing behavior is affected in different ways; some manifestations are explained in the references given above; adding nodes and polygon selection are in particular altered if the zoom is too small. That can be remedied by zooming out, doing what is required and zooming in again to "tighten" the changes.