CHAPTER 25

Application of a New Model of Vector Data Uncertainty

G.J. Hunter, J. Qiu, and M.F. Goodchild

INTRODUCTION

In previous papers by Hunter and Goodchild (1996) and Hunter et al. (1996), a model of vector data uncertainty was proposed and its conceptual design and likely manner of implementation were discussed. The model allows for probabilistic distortion of point, line, and polygon features through the creation of independent positional error fields in the x and y directions. These are overlaid with the vector data so as to apply coordinate shifts to all nodes and vertices in the data set to establish new, but equally likely, versions of the original data. By studying the variation in the family of outputs derived from the distorted input data, an assessment may be made of the uncertainty associated with the resultant information product. The model has now been developed and tested, and the purpose of this chapter is to report on its application in practice.

DEVELOPMENT OF THE UNCERTAINTY MODEL

The uncertainty model involves the creation of two independent, normally distributed, random error grids in the x and y directions. These grids are combined to provide the two components of a set of simulated positional error vectors regularly distributed throughout the region of the data set to be perturbed (Figure 25.1). The assumptions made are (a) that the error for each node or vertex has a circular normal distribution, and (b) that its x and y components are independent of each other. The grids are generated with a mean and standard deviation equal to the estimate for positional error in the data set to be perturbed (a prerequisite for use of the model). These error estimates, for example, might come from the residuals at control points reported during digitizer setup, or from an associated data quality statement.

By overlaying the two grids with the data to be perturbed, x and y positional shifts can be applied to the coordinates of each node and vertex in the data set to create a new, but equally probable, version of it. Thus, the probabilistic coordinates of a point are considered to be (x + error, y + error). With the distorted version of the data, the user then applies the same set of procedures as required previously to create the final product, and by repeating the procedure a number of times the variability residing in the end product may be assessed. Alternatively, several different data sets may be perturbed (each with its own error estimate) before being combined to assess final output uncertainty. While the model does require an a priori error estimate for creation of the two distortion grids, it is the resultant uncertainty arising from the use of perturbed data due to simulation which is under investigation—hence its label as an "uncertainty" model.

As discussed more fully in Hunter and Goodchild (1996), the first step in implementing the model is to determine an appropriate error grid spacing. If it is too large, the nodes and vertices of small features in the source data will receive similar-sized shifts in x and y during perturbation and the process will not be random. Conversely, if the grid is too small then processing time is increased as additional grid points are needlessly processed. Experience to date suggests that an appropriate spacing be selected from one of the following:

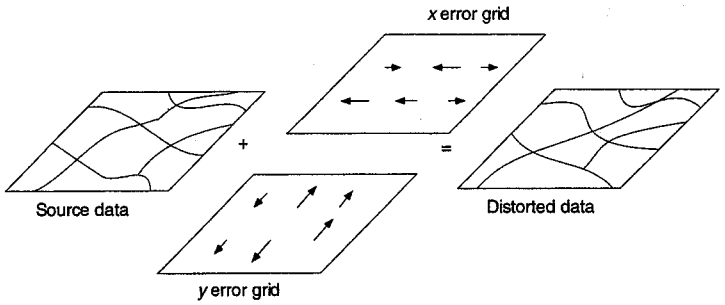

Figure 25.1. The model of vector data uncertainty uses normally distributed, random error grids in the x and y directions to produce a distorted version of the original data set.

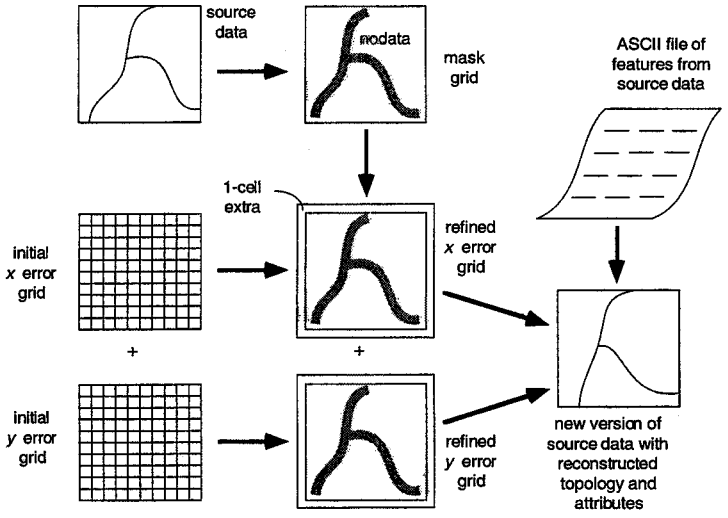

Figure 25.2. Development diagram for the uncertainty model.

- the standard deviation of the horizontal positional error for the source data; or
- a distance equal to an established standard—for example, 0.5 mm at source map scale where the data has been digitized; or
- a threshold value smaller than the user would care to consider, given the nature of the data to be processed.

Using Figure 25.2 as a guide, the second stage is to generate the x and y error grids. To ensure the grids completely cover the extent of the source data, their dimensions are predetermined by setting a window equal to the data set's dimensions, and the cell size equivalent to the chosen grid spacing. The grids are created automatically with these parameters and populated with randomly placed, normally distributed values having a mean (usually zero) and a standard deviation as previously defined. It should be noted that selecting the standard deviation as the grid spacing has no effect on the random population of the grids. These two grids are only temporary and will require further refinement before being used to perturb the source data.

To optimize processing time, the number of cells in the error grids needs to be reduced, since unless the data set is extremely dense there will be many unwanted cells processed during the operation. To achieve this the original vector data set is converted to grid format to form a temporary masking grid that only contains "live" cells—that is, those which the source data either lie within or pass through. Polygons are processed as line strings since only their boundaries are perturbed. Cell attributes that are maintained during rasterization are unimportant, given that the grid is only used for masking purposes and all other noncontributing cells are given a null or "nodata" value.

At this point there is a potential problem with using a masking grid that contains 1-cell-wide strings representing line or polygon features (Figure 25.3). As mentioned in Hunter and Goodchild (1996), there is some likelihood (although small) that the magnitude and direction of adjacent x and y error grid shifts may cause

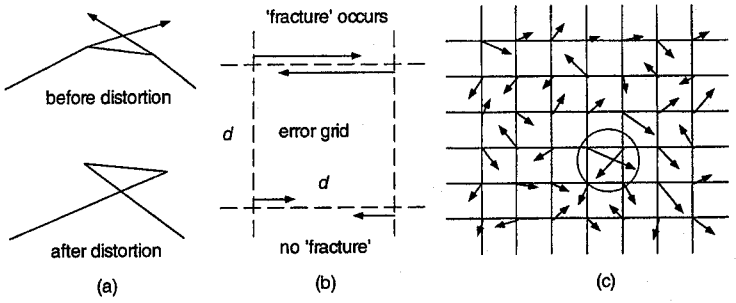

Figure 25.3. In (a), uncontrolled shifts between neighboring error grid points may cause unwanted "fractures" or transposition of features. In (b), "fractures" occur when the difference between neighboring shifts is larger than their separation (*d*). In (c), a "fracture" is circled, requiring filtering on the basis of neighboring shifts.

them to overlap, resulting in possible loss of topological integrity in the source data when applied to it (for example, unwanted loops caused by the transposition of adjacent vertices or nodes). The solution is to filter any "offending" pairs of shifts, which first requires that a spread function be applied to the mask grid for a distance of at least three standard deviations either side of the initial mask cell (up, down, left and right). A new masking grid is created during the process and any previously null cells affected by the operation are automatically returned to active status. The masking grid is then overlaid with the initial error grids to provide reduced versions of the x and y grids that contain only shift values surrounding features in the source data. Finally, the error grids are expanded by the width of a cell on all sides (with null values) to support the treatment of edge effects during processing.

To test for possible "fractures" between neighboring shifts (Figure 25.3a), a routine was developed to test the difference between consecutive cells (in horizontal or row sequence for the x grid, and vertical or column sequence for the y grid) to determine whether the absolute value of the difference between them was greater than their separation distance (Figure 25.3b). If so, then a "fracture" has the potential to occur at that location if there are data points nearby and a filter must be applied to average out the shift values on the basis of their neighbors. The procedure is iterative and proceeds until no "fractures" exist in either error grid.

In the final step of the model's development, values in the error grids must be transferred to the data set being perturbed. Naturally, it will be rare for nodes and vertices in the source data to coincide exactly with the error grid points, and a method was required for calculating x and y shifts based on the neighboring values in the grid. To achieve this, a bilinear interpolation procedure was used in which the x and y shifts assigned to each point are calculated on the basis of the respective shifts of the four surrounding grid points.

An ASCII feature file containing the identifier and coordinates of each data point was automatically derived, and the four surrounding error shifts were determined for each point then used to interpolate the shift values to be applied. A proximity threshold was also applied to ensure that data points close to a grid point would automatically receive that points' x and y shifts without computation. The distorted coordinates were then written to an output file and the file topology was rebuilt. Finally, the attributes belonging to each feature in the original data set were rewritten to their parent features in the distorted version of the data set.

The entire process runs as an Arc/Info AML script which calls a random number generator written in C. The AML program prompts the user for the name of the file to be perturbed, its data type (point, line, or polygon), the error grid size, the standard deviation of the horizontal positional error in the features, and the number of perturbations required. The code is freely available at the primary author's website given at the end of this chapter.

APPLICATION OF THE UNCERTAINTY MODEL

Polygon Area Estimation

The first application of the model is a simple one—estimation of the areal uncertainty of a set of polygons. In this case we took a group of six polygons that had been digitized from a source map at a scale of 1:50,000. We estimated that the digitizing was performed with a standard deviation of 25 m which was also the error grid spacing chosen—given that any polygon boundary segment length less than this value would have no significant impact upon subsequent application of the data. The set of polygons was perturbed 20 times and the results of overlaying the 20 realizations can be seen in Figure 25.4. Then, by appending the 20 polygon sets and statistically analyzing the ar-

eas for each of the six polygon identifiers, we were able to easily construct a table of mean polygon areas and their standard deviations (see Table 25.1).

Point-in-Polygon Overlay

In the next application we took a set of 30 point features and overlaid them with the set of six polygons used before (see Figure 25.5). The points were deliberately placed near polygon boundaries and junctions. In the first instance we held the polygon boundaries fixed (that is, we assumed they had high positional accuracy), and perturbed the point set 20 times (with an error grid spacing again of 25 m and a standard deviation of 25 m). As each perturbed point set was overlaid with the fixed polygon boundary file, we recorded the identifier of the polygon in which each point was deemed to lie and appended the point identifiers and their associated polygon numbers to an output file. When the 20 overlays were completed, a frequency count was taken and the results were summarized in Table 25.2.

We then perturbed both the points and polygons 20 times each and overlaid them a total of 400 (20×20) times—a process that was automated quite simply with a short AML script. While the two data sets once more employed an error grid spacing of 25 m and a standard deviation of 25 m, these parameters are easily varied by a user and need not be the same—which would enable perturbation of different data sets with different errors. The results of the 400 overlays are shown in Table 25.3.

Polygon to Grid Conversion

In the final application we took the same set of six polygons, perturbed them 20 times using the same error grid size and standard deviation as previously, and converted them to grid cells in order to estimate the variation associated with both the allocation of polygons to grid cells and total class areas. After each polygon to grid conversion, the number of cells belonging to the six polygons were counted (since polygon IDs were maintained during conversion), and the mean and standard deviation of the number of cells formed from each parent polygon were recorded. As expected, the mean number of cells remained within one or two of the number recorded when the unperturbed polygon set was converted. However, we believe the standard deviation of the number of cells is a useful statistic that could be put to further use as described later. The results are shown in Table 25.4.

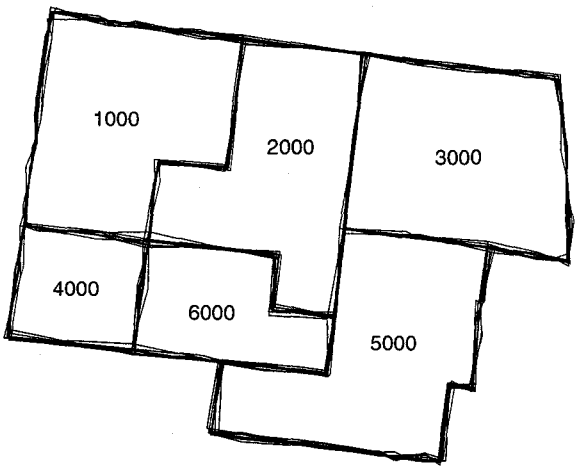

Figure 25.4. Showing the results of 20 perturbations of the polygon data set when overlayed.

Table 25.1. Showing the mean and standard deviation for the area of each polygon after 20 perturbations (using a standard deviation of 25 m for the horizontal positional error of the polygon boundaries).

Polygon ID	Mean Area (sq. m)	Standard Deviation (m)
1000	891858.3	5419.6
2000	890108.5	9920.3
3000	945221.7	3889.6
4000	358774.9	5407.7
5000	980114.9	6748.4
6000	459806.7	7175.6

DISCUSSION OF RESULTS

From these examples, there are several comments that can be made with respect to applying the vector data uncertainty model in practice.

Clearly, it has the potential to help educate users about the meaning of metadata items that are attached to a data set. For example, in conjunction with a statement of the standard deviation of positional error, a diagram such as Figure 25.4 could be included in a data quality report showing how the data may probably vary in position according to the meaning of that error descriptor.

Some useful statistics also arise from the model. For instance, the class membership frequencies shown

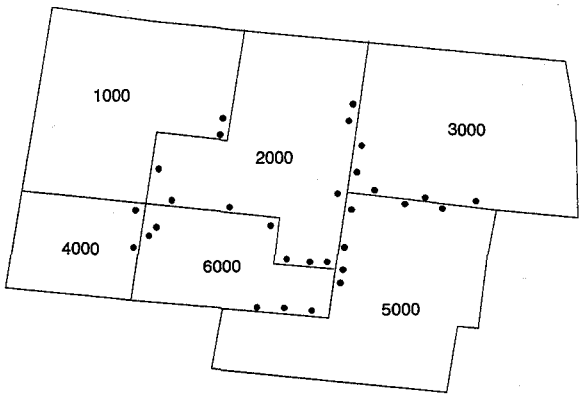

Figure 25.5. Showing the set of 30 points that were overlayed with the set of six polygons.

Table 25.2. Showing the observed frequencies for which each point lies in any of the six polygons after 20 perturbations (for clarity, points lying only in the one polygon are not listed). Asterisks indicate the polygon in which the point was assigned prior to perturbation of the data.

Point ID	Poly 1000	Poly 2000	Poly 3000	Poly 4000	Poly 5000	Poly 6000
2	16*	4	–	–	–	–
9	–	11*	–	–	–	9
13	–	3	17*	–	–	–
17	–	–	7	–	13*	–
18	–	–	12*	–	8	–
19	–	–	6	–	14*	–
23	–	–	–	–	5	15*
27	–	–	–	–	8	12*

Table 25.3. Showing the observed frequencies for which each point lies in any of the six polygons after 20 perturbations each of the point and polygon data sets, then overlayed 400 (20 x 20) times (for clarity, points lying only in the one polygon are not listed). Asterisks indicate the polygon in which the point was assigned prior to perturbation of the data.

Point ID	Poly 1000	Poly 2000	Poly 3000	Poly 4000	Poly 5000	Poly 6000
2	290*	110	–	–	–	–
9	–	260*	–	–	–	140
13	–	27	373*	–	–	–
17	–	–	156	–	244*	–
18	–	–	280*	–	120	–
19	–	–	129	–	271*	–
23	–	–	–	–	108	292*
27	–	–	–	–	140	260*

in Tables 25.2 and 25.3 represent quantities that until now have been quite difficult to define. Certainly, they have been able to be computed in certain cases—for example, when classifying remotely sensed imagery—but there has been no ready solution for vector overlay operations. Furthermore, the standard deviations computed for polygon and cell class areas can serve as input to formal error propagation computations. For example, when calculating population densities we could combine the standard deviation of the area with that of the population count to yield the density variance.

The model has the added ability to indicate portions of a data set that may be highly sensitive to perturbation—thereby warning users that the data set is potentially unsuitable for use in a particular region. For instance, in the polygon-to-grid conversion example above it is suspected that there is little variance in the number of polygons formed after each perturbation due to the fairly regular N-S, E-W boundaries of the polygons. On the other hand, polygons with direction trends at 45° to the cardinal axes might prove highly variable when perturbed and subsequently converted.

Importantly, the model has the capacity (in certain cases) to be able to turn simulated error in position into measurable attribute uncertainty—for example, the transformation of polygon boundary error into polygon area uncertainty. There is also the potential to assess the uncertainty of a final information output after a sequence of spatial operations, in which all data sets have had their positions perturbed to varying degrees according to their individual accuracies. At the same time, we believe that some problems are ill-posed and not well suited to this model—for instance, perturbing closely set contour lines where there is a likelihood of the perturbed contours crossing each other.

CONCLUSION

This chapter has described the development and application of an uncertainty model for vector data which operates by taking an input data set of point, line, or polygon features and then applying simulated positional error shifts in the x and y directions to calculate new coordinates for each node and vertex. In effect this produces

Table 25.4. Showing the mean and standard deviation of the number of cells formed from each of the six polygons after perturbation 20 times and subsequent conversion to grid format.

Polygon ID	Mean No. of Cells	Standard Dev. (cells)
1000	355.4	2.7
2000	356.1	3.5
3000	376.5	2.3
4000	145.3	2.2
5000	385.6	3.4
6000	182.8	2.6

a distorted, but equally probable, representation of the data set that can be used to create a family of alternative outputs, usually in map form. Assessment of the variation in the outputs can be used to provide an estimate of the uncertainty residing in them, based on the error in the source data and its propagation through the subsequent algorithms and processes employed. The model was tested in several applications, viz: (a) perturbing polygon boundaries to determine a mean and standard deviation for the area of each polygon; (b) perturbing point and polygon data sets prior to point-in-polygon overlay, which yielded class membership frequencies for each point; and (c) perturbing polygon boundaries prior to polygon-to-grid conversion, to generate a standard deviation for the number of cells in each polygon as a result of the conversion algorithm.

FURTHER INFORMATION

For further information, readers are directed to the principal author's home page where a tutorial containing AML and C source code exists to implement the uncertainty model and automatically perturb point, line and polygon files. The URL is:

http://www.geom.unimelb.edu.au/people/gjh.html

ACKNOWLEDGMENTS

The authors acknowledge funding support received under Australian Research Council (ARC) Large Grant No. A49601183—"Modeling Uncertainty in Spatial Databases."

REFERENCES

Hunter, G.J. and M.F. Goodchild. A New Model for Handling Vector Data Uncertainty in GIS, *J. Urban Reg. Inf. Syst. Assoc.*, 8(1), pp. 51–57, 1996.

Hunter, G.J., B. Höck, M. Robey, and M.F. Goodchild. Experimental Development of a Model of Vector Data Uncertainty, *Proceedings of the 2nd International Symposium on Spatial Accuracy Assessment in Natural Resources and Environmental Sciences*, Fort Collins, CO, 1996, pp. 214–224.