PEER-REVIEWED ARTICLE

Integration of Simulation Modeling and Error Propagation for the Buffer Operation in GIS

Howard Veregin

Abstract

Error propagation modeling in layer-based GIS is based on explicit mathematical models representing the mechanisms whereby errors in source layers are modified by GIS data transformation functions. For many classes of data transformation functions, however, little is currently known about error propagation mechanisms. Simulation modeling is an attractive alternative in such cases, as it can be applied whether or not a formal error model has been developed. However, due to its computationally intensive character, simulation modeling is not practical in the realm of applied GIS, and thus serves primarily as a source of informal, anecdotal information about the dangers inherent in GIS-based spatial analysis when input data are of imperfect quality. This paper explores a methodology for enhancing the utility of the information derived from simulation modeling for assessing the quality of GIS derived data. The methodology is based on the integration of simulation modeling and error propagation. Using the buffer operation as an example, simulation modeling is used to derive a formal mathematical expression describing how error is propagated through the operation from the source layer to the derived buffer. This expression is based on information about the source layer that is relatively easy to obtain, and can be used in an applied environment to estimate derived buffer accuracy without the need to perform sensitivity analysis.

Introduction

Error propagation modeling in a GIS focuses on the dynamic processes whereby errors in source data are modified by GIS data transformation functions and then passed to derived data. Due to the importance of issues of database accuracy and quality assurance for GIS-based spatial analysis, error propagation has long been a subject of concern in the GIS research community. However, it has only been relatively recently that researchers have been able to construct automated systems to perform error propagation in real-time for selected subsets of data transformation functions (e.g., Heuvelink *et al.*, 1989; Lanter and Veregin, 1990; Carver, 1991). This is a reflection of the complexity of error propagation modeling in a GIS context and the lack of theoretical knowledge about error propagation mechanisms for all but the slimmest collection of data transformation functions.

Error propagation modeling involves the application of formal mathematical models that describe the mechanisms whereby specific types of source errors are modified by particular data transformation functions. Error models have been derived from statistical theory applied to error analysis (e.g., Taylor, 1982; Burrough, 1986), probability theory (Newcomer and Szajgin, 1984; Veregin, 1989a), and approximation by Taylor series expansion (Heuvelink *et al.*, 1989; Wesseling and Heuvelink, 1991). For many classes of GIS data transformation functions, however, little is known about the theory of error propagation, and error propagation models corresponding to these functions remain to be developed. Despite a great deal of research into the issue of database accuracy (see Veregin, 1989b), error propagation mechanisms are not well-understood for all but a handful of GIS data transformation functions. Formal specification of error propagation functions in a GIS applications environment is also problematic given the difficulty of meeting the controlled conditions that are often assumed to exist in the theoretical realm.

Given such limitations, alternatives to formal error propagation modeling have received considerable attention in the literature. One such alternative is Monte Carlo simulation modeling. This method is based on random introduction of error into source data to create a set of "realizations" reflecting some assumptions about the nature and level of error present. These realizations are then passed through a particular sequence of data transformation functions multiple times to compute summary statistics for a particular set of assumptions about the nature and level of error. Simulation modeling has been used to examine the effects of errors in suitability analysis procedures (Lodwick, 1989), classification methods (Goodchild and Wang, 1989), land valuation based on land use and soil information (Fisher, 1991a), and viewshed calculations from digital elevation models (Fisher, 1991b; Fisher, 1992).

Unfortunately, while the results of simulation modeling are often of great practical importance, the technique itself has limited utility in an applied GIS environment. The technique is often too computationally intensive to be practical as a means of assessing derived data quality when human, computer, and financial resources are limited. Simulation modeling therefore serves primarily as a source of informal, anecdotal evidence about the errors that might result from the application of a particular data transformation function when source data are of imperfect quality. One unfortunate

Department of Geography, Kent State University, Kent, OH 44242-0001.

Photogrammetric Engineering & Remote Sensing, Vol. 60, No. 4, April 1994, pp. 427–435.

0099-1112/94/6004–427$03.00/0

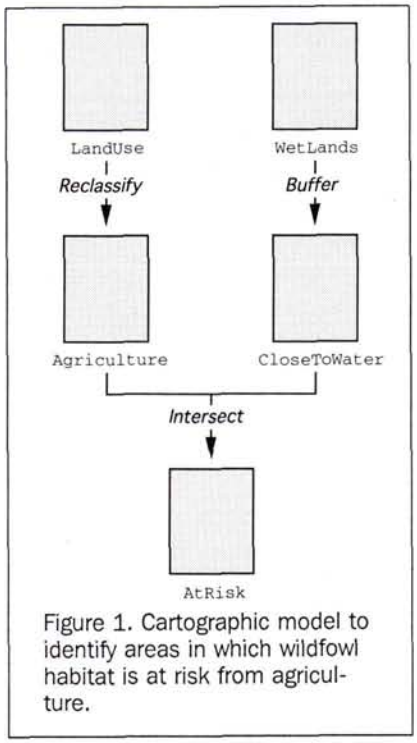

Figure 1. Cartographic model to identify areas in which wildfowl habitat is at risk from agriculture.

implication is that the knowledge gained from simulation modeling is probably not being applied in a way that maximizes its utility or significantly enhances the quality of the results of GIS-based spatial analysis.

This paper explores a methodology for enhancing the utility of the information derived from simulation modeling for assessing the quality of GIS derived data. The methodology is based on the integration of simulation modeling with a formal error propagation paradigm in the context of the buffer (or proximity) operation. The buffer operation involves the delineation of a geometric zone of specified width around a set of input features.

The operation is frequently applied in GIS-based spatial analysis involving site selection, suitability analysis, and environmental modeling (Star and Estes, 1990). In this study, simulation modeling is used to derive a formal mathematical expression describing how error is propagated through the buffer operation from the source data to the derived buffer. This expression can then be used to estimate buffer accuracy directly based on characteristics of the input data and parameters defining how the buffer operation is applied.

Error Propagation Modeling in Layer-Based GIS

Error propagation modeling in layer-based GIS is based on a formalism often referred to as the "geographic data matrix." According to this formalism, geographical data are defined in terms of three domains -- space, time, and theme -- such that any observation can be located in a three-dimensional coordinate system based on its spatial, temporal, and thematic coordinates (Berry, 1964). By convention, the spatial domain is decomposed into horizontal dimensions (usually denoted by "x and y") and a vertical dimension (usually denoted by "z"). Time (usually denoted by "t") and theme are conventionally referred to as the "aspatial" domains.

Geographical features are real-world entities encoded in a GIS database as "objects" (Moellering, 1992). These objects acquire meaning once they have been attributed with thematic information, or "attributes." An attribute value is a measurement of a specific attribute for a particular object in a database. Through the association of attributes with objects, a database achieves a representation of the multiplicity of relations among real-world entities. However, the attributes stored in a database necessarily represent only a small subset of the attributes associated with the corresponding real-world entities. Any GIS database is therefore an abstraction of the real world, an incomplete generalization designed for some specific purpose.

A GIS database may be organized following a variety of models. Among the more common models are

- Least Common Geographical Units (LCGUs) or Integrated Terrain Units (ITUs),
- objects, and
- layers (Lanter, 1992).

LCGU-based models collapse all geographical data into a single integrated record. Object-oriented models define individual features and their associated attributes as semantically meaningful objects with inheritable properties (Lanter, 1992). In layer-based approaches, data are organized in a set of co-registered thematic map separations known as "layers." Each layer is a collection of objects and the corresponding values for a selected attribute. Co-registered layers have the same spatial and temporal coordinates; only their thematic content varies. Alternatively, space and theme may be held constant while time varies, as in databases constructed for performing change detection.

The layer-based model is the basis for a formal method of representing data transformations in GIS, known as "cartographic modeling" (Tomlin and Berry, 1979; Tomlin, 1990). A cartographic model depicts a flow of data from source layers through derived layers for a specific sequence of GIS data transformation functions. The data transformation functions and source layers used to produce a given derived layer effectively define the meaning of that layer in the context of the database. A cartographic model is thus a representation of a transformation of selected components of a database in order to make explicit a set of relationships that are implicit in the data encoded in the source layers (Lanter, 1992).

Figure 1 shows a cartographic model for a simple GIS application designed to identify areas in which wildfowl habitats are at risk from agricultural activity. These areas are delineated as locations that meet the following two criteria:

- they are classified as agricultural land, and
- they are in close proximity to water (a surrogate for wildfowl habitat).

The cartographic model depicted in Figure 1 can also be represented as nested set of data transformation functions, i.e.,

AtRisk = *Intersect (Reclassify* (LandUse), *Buffer* (WetLands))

This functional form represents the propagation of data through a sequence of data transformation functions, such that the final derived layer (AtRisk) depends only on the two source layers (LandUse and WetLands). This facilitates error propagation modeling, as described below. (Note that the labels used to designate the GIS data transformation functions in this example are generic. The exact implementation of this example would depend on the specific characteristics of the system that was employed.)

Error propagation in layer-based GIS exactly mirrors the process of data propagation depicted in the functional repre-

sentation above. In error propagation, each layer is replaced with an accuracy index value corresponding to the layer, and each data transformation function is replaced with a corresponding error propagation function that models the mechanisms whereby the accuracy index is modified by the data transformation function. Any error propagation function is, therefore, specific to

- a particular accuracy index, and
- a particular data transformation function.

In many cases, it may also be important to consider factors such as the spatial distribution of error or the degree to which errors on different data layers coincide (Lanter and Veregin, 1992), because error propagation models do not typically emulate error propagation mechanisms perfectly.

Integration of Simulation Modeling and Error Propagation

The main limitation of the error propagation paradigm described above is the lack of theoretical knowledge of error propagation mechanisms for all but the slimmest subset of GIS data transformation functions. As noted above, systems that have been built to perform error propagation in real time are able to accommodate only a limited number of data transformation functions and rely on simplifying assumptions about the mechanisms of error propagation. This basic deficiency in knowledge reflects the complexity of error measurement and propagation in a GIS context. Given this limitation, alternatives to formal error propagation modeling have received considerable attention in the literature. Monte Carlo simulation modeling is one such alternative. Its main attraction lies in the fact that it can be universally applied to any data transformation function, whether or not a formal error model has been developed for that function (Openshaw, 1989).

Simulation modeling involves six basic steps, as detailed below.

Step 1. Make assumptions regarding error characteristics (e.g., error type, level, spatial distribution, etc.) for the source layer.

Step 2. Randomly introduce error into the source layer based on the error characteristics assumed to exist for that layer. A source layer perturbed in this way represents a "realization" of the error characteristics assumed to exist for that layer.

Step 3. Apply a GIS data transformation function, or a sequence of such functions, to the perturbed source layer to obtain a desired derived layer.

Step 4. Compare the derived layer to a "reference" layer derived from an unperturbed source.

Step 5. Repeat Steps 2, 3, and 4, M times. The value of M depends on the data, error characteristics, and spatial data transformation functions under consideration.

Step 6. Compute statistics summarizing the characteristics of the errors present in the set of M derived layers.

These statistics contain information about the net effects of applying the particular data transformation functions to input data with the assumed error characteristics.

Like error propagation modeling, simulation modeling is concerned with the implications of source errors for the quality of the data derived through the application of GIS data transformation functions. However, simulation modeling is typically too computationally intensive to permit the accuracy of derived data to be assessed in real time. This study demonstrates that the utility of simulation modeling can be enhanced through its integration with a formal error modeling paradigm. This integration is achieved through the derivation of empirical relationships from simulation model results. These relationships are then used to estimate output accuracy directly based on characteristics of the input data and the GIS data transformation function under consideration.

This study uses the buffer operation as an example to document the manner in which simulation modeling and error propagation can be integrated. A simulation-based approach is used to derive a mathematical expression describing how error is propagated through the buffer operation. This expression is then used to obtain an estimate of buffer accuracy for a simple GIS application based on characteristics of the input data. Although the primary purpose of this paper is to describe and document how empirical relationships derived from simulation modeling may be used to define error propagation functions, readers may also be interested in the empirical results themselves, and their implications for data quality and error propagation in the context of the buffer operation.

Propagation of Error in the Buffer Operation

The buffer operation is a basic GIS data transformation function in which a buffer of some specified width is delineated around a set of features in an input layer. The buffer consists of the set of locations for which the distance from the set of features is no greater than some specified distance threshold. The buffer operation may be applied to either raster or vector data. For vector data, the buffer is a geometric object encompassing the set of locations within the distance threshold. For raster data, the buffer comprises the set of cells within the distance threshold from a set of "feature cells" that define the features around which the buffer is to be generated.

Error propagation for the buffer operation is often assumed to refer to the consequences of errors in feature locations. This is consistent with the vector representation of a buffer as a geometric object. However, thematic error is relevant in the context of the buffer operation, particularly for that class of geographical data for which the spatial component (feature location) is a function of the thematic component (categorical attribute value). This class of data, often encountered in environmental modeling applications, is sometimes referred to as "area class data" (Chrisman, 1989; Goodchild and Dubuc, 1987). Such data are often derived in remote sensing using image classification methods that identify nominal cover types based on pixel reflectance values. Cover type "polygons" can be defined as groups of contiguous pixels with the same cover type assignment. If the cover type assignments change, the locations of the boundaries between polygons also change. This class of data is distinct from another major class of geographical data for which polygon boundary locations are defined *a priori* as the basis for sampling and data collection (e.g., census tracts).

This study focuses on the propagation of thematic error through the buffer operation for area class data. The study employs raster data in the simulation modeling procedures, due to the amenability of raster data to the measurement of

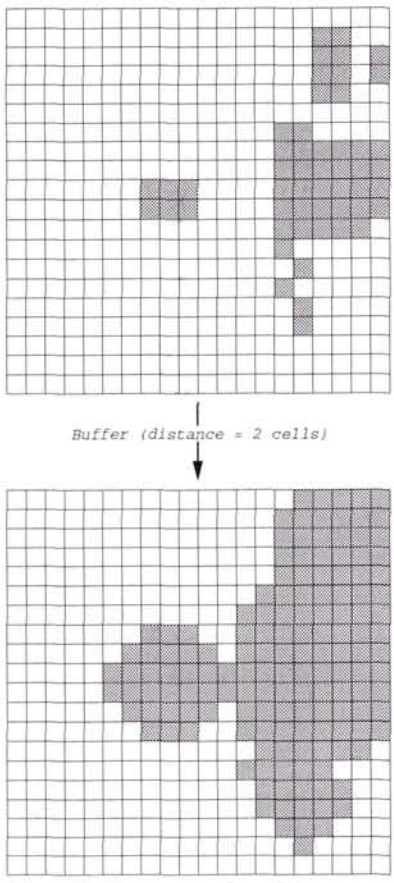

Figure 2. Representation of the buffer operation.

thematic error. A representation of the buffer operation for raster data is shown in Figure 2. The techniques described in the study are not generally applicable to vector data. Measures of error that are integral to the application of the simulation model (e.g., the particular error index used, the definitions given for feature geometry, and the spatial distribution of error) are poorly-defined in the case of vector data. The buffer operation is also qualitatively different for vector data, as it involves explicit geometrical identification of the features around which buffers are constructed. The study also assumes that buffer size is a constant value for any layer, rather than a variable quantity related to the nature of the features to be buffered (e.g., class of road or stream).

Thematic error, as defined here, refers to an incorrect assignment of an attribute value to a cell in a raster layer. Two kinds of thematic errors affect error propagation for the buffer operation:

- *Error of Omission.* This corresponds to a situation in which a cell that actually is an element of the set of features around which the buffer is to be generated has been assigned a value indicating that it is not a member of this set.
- *Error of Commission.* This corresponds to a situation in which a cell has been assigned a value indicating that it is a member of the set of features around which the buffer is to be generated, when in fact the cell is not a member of this set.

Propagation of error through the buffer operation is depicted in Figures 3 and 4. Each of the rectangles in these figures represents a layer, while the shaded areas on each layer represent features. The rows of layers depict buffers of increasing size (i.e., increasing distance threshold). The first column (labeled "actual") represents the true location of features, which in practice will be unknown. The second column (labeled "estimated") represents the location of features as depicted in the database. This second column contains errors, which are shown in the third column (labeled "errors"). The relative accuracy of the buffer for any given buffer size is easily interpreted in terms of the number of error cells.

Figure 3 depicts the propagation of errors of omission. For any given buffer size, a portion of the actual buffer is missing due to incomplete specification of the set of features around which the buffer is to be generated. The figure depicts an example in which the buffer constructed around the correctly represented feature cells tends to over-ride the inaccuracies in the buffer introduced by errors of omission. The magnitude of this effect, however, is dependent on the configuration of the features in a layer, and does not occur in all situations.

Figure 4 depicts the propagation of errors of commission. A commission error results in a buffer being constructed around a set of features that do not actually exist at that particular location. As buffer size increases, errors of commission tend to spread out over space, thereby increasing the amount of error introduced.

Figure 4 depicts an important effect associated with error propagation for the buffer operation. Following an initial decrease in the accuracy of the derived buffer layer, further increases in buffer size result in an accuracy increase. This can be seen in the decline in the number of error cells between the last two rows of Figure 4. The buffer has become so large that it saturates the entire layer, such that differ-

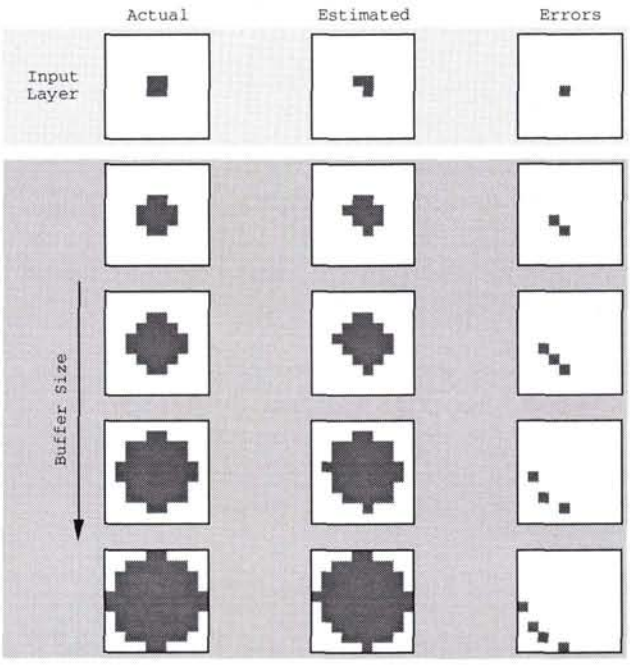

Figure 3. Propagation of omission errors.

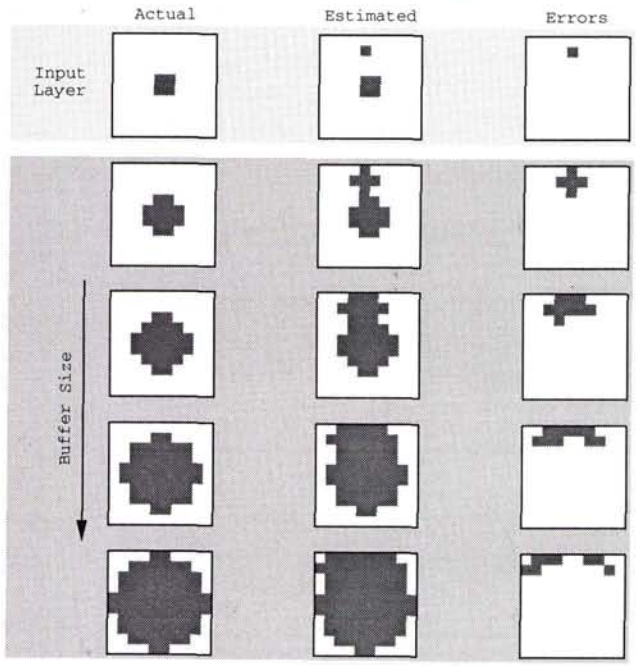

Figure 4. Propagation of commission errors.

ences between the actual and estimated buffers are less pronounced. This saturation effect depends on the likelihood that a given cell will be within the threshold distance of more than one feature cell. The tendency for this to occur is driven upwards as buffer size increases. (It is also dependent on the number and spatial distribution of feature cells, as described below.) If a cell becomes a member of the set of buffer cells by virtue of its proximity to more than one feature cell, then an error in any one of these feature cells will have no impact on the accuracy of the buffer.

The tendency for this saturation effect to occur implies that, especially when buffer sizes are large, the derived buffer layer may actually be more accurate than the source layer used to construct it. Indeed, in situations where the buffer encompasses nearly the entire derived layer, the accuracy of the layer may be close to 100 percent (although the information content of the layer would be relatively low in this case). This contradicts the prevailing "weakest link" argument that derived data in GIS can never be any more accurate than the sources used to derive them (e.g., Walsh *et al.*, 1987). Other GIS data transformation functions have also been shown to be associated with an accuracy increase in certain circumstances (Veregin, 1989a; Lanter and Veregin, 1992).

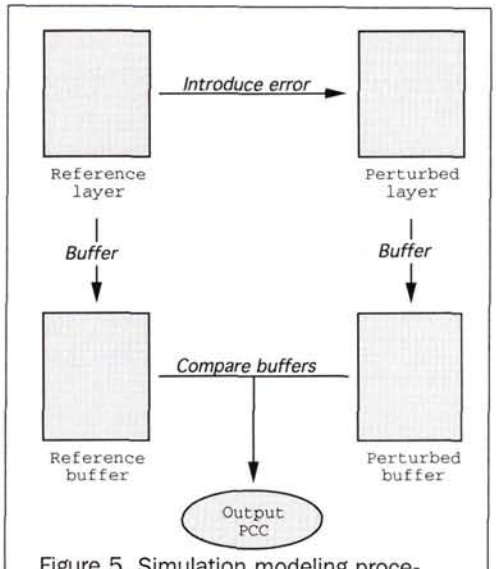

Figure 5. Simulation modeling procedure.

Simulation Modeling for the Buffer Operation

Simulation modeling of error propagation for the buffer operation follows the procedure outlined in Figure 5. This procedure is based on a comparison of a reference buffer that is assumed to be completely accurate with a perturbed buffer into which error has been introduced. The reference buffer is derived from a reference layer that contains what is assumed to be an accurate representation of the set of features around which the buffer is to be generated. Various realizations of the degree and spatial distribution of error are derived from the reference layer to produce a perturbed layer, which is then used to construct the perturbed buffer.

Comparison of the reference and perturbed buffers yields information on the degree of error present in the perturbed buffer layer. This information is summarized using the familiar PCC (proportion correctly classified) index (see Congalton (1991) for a more detailed discussion). The PCC index, often used in classification accuracy assessment in remote sensing, is a measure of the probability that a given cell on a layer has been assigned to the correct cover type. The "output PCC" index in Figure 5 refers to the proportion of cells that have been assigned the same cover type on both the reference buffer and perturbed buffer layers. In this case, there are only two cover types, which define whether or not a given cell is a member of the set of buffer cells.

It is expected that output PCC will be affected by characteristics describing the source data and defining how the buffer operation is applied. These characteristics are summarized below.

- *Input PCC* refers to the probability that the cells in the perturbed layer are correctly classified as either feature or non-feature cells. A positive relationship should exist between input PCC and output PCC, indicating that the higher the accuracy of the input data, the more accurate the derived buffer will be. The procedure employed here assumes that input PCC is the same for feature and non-feature cells.
- *Buffer size* refers to the width of the buffer around the feature cells as a function of the distance threshold value selected. Buffer size should have an impact on the accuracy of the derived buffer, but this impact depends on two competing forces. In one sense, a larger buffer implies less accuracy, because errors, whether commission or omission, grow in direct proportion to the width of the buffer. This effect is offset by the tendency for saturation to occur. As described above, saturation implies that a buffer size may be reached at which further buffer size increases will be associated with an increase in buffer accuracy.
- *Feature probability* refers to the proportion of cells in the perturbed layer that are defined as feature cells. In general, a

higher probability implies a higher buffer accuracy, due to the enhanced tendency for saturation to occur, even for relatively small buffer sizes. The procedure employed here ensures that the feature probability is equal for the reference and perturbed layers.

- *Feature geometry* refers to the degree to which feature cells tend to cluster in space. There should be a general tendency for less clustered distributions to be associated with more accurate buffers. For any given feature probability, a more dispersed set of feature cells will result in a greater number of buffer cells, which will in turn result in an enhanced propensity for saturation. The procedure employed here does not ensure that feature geometry will be identical for the reference and perturbed layers. Measurements of feature geometry used in the procedure refer to the perturbed layer, as the reference layer is typically unknown in applied contexts. Feature geometry is computed using a 1-1 joins count to measure autocorrelation (Cliff and Ord, 1973; Congalton, 1988). This statistic is based on the propensity of feature cells to be neighbors. Neighbors are defined in terms of rook's case (i.e., only cells that share a border of non-zero length are considered neighbors). Other measures of autocorrelation could, of course, also be used.
- *Error distribution* refers to the degree to which misclassified cells tend to cluster in space. A higher degree of clustering should be associated with a lower level of error in the derived buffer. If misclassified cells tend to cluster in space, then the number of misclassified buffer cells that will be produced for a given buffer size will be minimized. Like feature geometry, error distribution is defined in terms of a 1-1 joins count. In this case, the statistic is based on the propensity of misclassified cells to be neighbors.
- *Grid size* is defined as the square root of the number of cells in the input layer. For square grids, grid size will be equal to the number of rows or columns. Grid size should mitigate the effects of many of the factors identified above, e.g., the relative significance of buffer size and the propensity for saturation to occur. Grid size is identical for the reference and perturbed layers, as well as the reference and perturbed buffers.

Results

The program described above was run for 150 iterations for grid sizes ranging from 25 to 125 cells on a side (i.e., 625 to 15625 cells). A first-order polynomial regression was applied to the output data, using output PCC as the dependent variable. Explanatory variables were derived from the factors expected to affect buffer accuracy, as documented above. Final results, consisting of the set of all significant explanatory variables, are presented in Table 1.

Results indicate a close fit ($R^2 = 0.988$) between derived buffer accuracy and a set of explanatory variables that includes input PCC, buffer size, feature probability, feature geometry, and the spatial distribution of error. None of the interaction terms in the polynomial model is significant, and only one squared term is significant (feature probability). The linear model (obtained by excluding the squared feature probability term) yields an R^2 of 0.984.

Of the set of variables expected to affect output PCC, as documented above, only grid size appears to be insignificant. For the range of grid sizes examined here, this variable seems to exert no significant effect on output PCC either by itself or through mitigation of the effects of other variables. The same result was obtained for a separate polynomial regression model in which the inverse of grid size was employed.

Regression results suggest the following interpretation of the effects of the various explanatory variables:

TABLE 1. REGRESSION RESULTS.

Coefficient	Estimate	Std. Error	t-value	p-value
Input PCC	0.485	3.036×10^{-2}	15.97	<0.001
Buffer size	6.928×10^{-2}	6.732×10^{-3}	10.29	<0.001
Feature probability	2.164	0.202	10.72	<0.001
Feature geometry	-2.438×10^{-3}	4.721×10^{-4}	−5.16	<0.001
Error distribution	1.414×10^{-3}	4.409×10^{-4}	3.21	0.002
Feature probability^2	−2.605	0.395	−6.60	<0.001

$R^2 = 0.988$ F-value = 1919.13 df = 6,144 p-value < 0.001

- *Input PCC.* As expected, a positive relationship exists between input PCC and output PCC. Thus, derived buffers tend to be more accurate when the input layer is more accurate.
- *Buffer Size.* A positive relationship exists between buffer size and output PCC. Thus, derived buffers tend to be more accurate when buffer size is large. As noted previously, the effects of buffer size on output PCC depend on two competing forces, the first being the tendency for thematic errors to grow in direct proportion to the width of the buffer, and the second being the tendency for saturation to occur for large buffer sizes. The positive relationship observed in the regression results indicates that it is the second of these two forces that is more important.
- *Feature Probability.* The relationship between feature probability and output PCC is slightly more complicated than expected. The fact that the square of this probability is significant in the regression results suggests that a simple linear relationship, in which a higher probability implies a higher buffer accuracy, does not exist. Rather, the signs of the regression coefficient estimates for feature probability and squared feature probability indicate that, as the probability continues to increase, a point is reached at which output PCC will begin to decline. The exact cause of this effect is uncertain.
- *Feature Geometry.* As expected, an inverse relationship exists between feature geometry (the degree of clustering of feature cells) and output PCC. The less clustered the feature cells, the more accurate the derived buffer. This is due to the propensity for a dispersed set of feature cells to produce a greater number of buffer cells, which in turn is associated with an enhanced propensity for saturation.
- *Error Distribution.* As expected, a positive relationship exists between the error distribution (degree of clustering of misclassified cells) and output PCC. The less clustered the misclassified cells, the less accurate the derived buffer. Clustering of misclassified cells tends to minimize the number of misclassified buffer cells that will be produced for a given buffer size.

The regression results define a simple expression that can be used to estimate the accuracy of a derived buffer layer. The appropriate equation, derived from Table 1, is given below.

$$\begin{aligned}\text{Output PCC} = {} & 0.485 \times \text{Input PCC} \\ & + 6.928 \times 10^{-2} \times \text{Buffer Size} \\ & + 2.164 \times \text{Feature Probability} \\ & - 2.438 \times 10^{-3} \times \text{Feature Geometry} \\ & + 1.414 \times 10^{-3} \times \text{Error distribution} \\ & - 2.605 \times \text{Feature Probability}^2\end{aligned}$$

For illustrative purposes, this equation has been incorporated into an error propagation context based on the cartographic model depicted in Figure 1. Error propagation models for the *Reclassify* and *Intersect* operations are derived from Lanter and Veregin (1992). Parameter values used in the error propagation are defined in Table 2. (Values are necessarily somewhat arbitrary, as this example is hypothetical.) It is assumed that a buffer size of 2 cells is used. For the *Intersect* opera-

TABLE 2. PARAMETERS FOR PROPAGATION OF ERROR.

Layer	Parameter	Value
LandUse	Input PCC	0.85
	Number of Classes	5
WetLands	Input PCC	0.90
	Feature probability	0.05
	Feature geometry (standardized 1-1 joins statistic)	5.0
	Error distribution (standardized 1-1 joins statistic)	10.0

tion, it is assumed that the PCC of CloseToWater conditional on Agriculture is the same as the PCC of CloseToWater, i.e., that PCC[CloseToWater|Agriculture] = PCC[CloseToWater]. Propagation of the PCC index through the example GIS application is represented in Figure 6. In this example, the accuracy of the final data product AtRisk is considerably lower than either of the two sources—LandUse and WetLands—from which it is derived. The ability to use the derived equation in an operational context would require additional testing and validation of empirical relationships.

Conclusion

This study describes how simulation modeling can be integrated into a formal paradigm for error propagation modeling for layer-based GIS. The buffer operation is used as an example to illustrate the derivation of an empirical relationship that can be used to estimate the accuracy of a derived buffer layer. The relationship is based on variables that are relatively easy to obtain, including characteristics of the input data and parameters that determine how the buffer operation is applied. The integration of simulation modeling and error propagation enhances the utility of the information derived from simulation modeling in assessing the accuracy of the data derived through the application of GIS data transformation functions. Simulation modeling results can be used in formal modeling of error propagation mechanisms, and can thus be incorporated explicitly in a GIS applications environment, where they should have the greatest impact and utility.

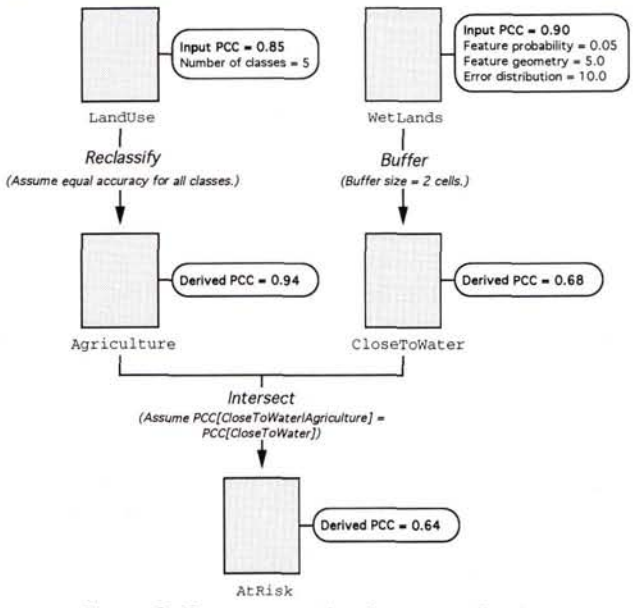

Figure 6. Error propagation for GIS application.

On a more prosaic level, this study also has implications for the propagation of thematic error through the buffer operation. Simulation model results indicate a close fit between derived buffer layer accuracy and a set of explanatory variables that include input PCC, buffer size, feature probability, feature geometry, and the distribution of error. More accurate buffers are associated with an accurate input layer, a large buffer size, a dispersed spatial distribution of feature cells, and a clustered spatial distribution of thematic errors. Feature probability -- the proportion of cells in the input layer that are defined as feature cells -- also exerts a significant impact, but the direction of this impact is somewhat ambiguous.

Acknowledgment

I wish to thank David Lanter, at the University of California, Santa Barbara, for his assistance in developing the conceptual framework for this paper, and for the infectious idealism he brings to his bear on his efforts to span critical gaps between GIS theory and practice.

References

Berry, B., 1964. Approaches to regional analysis: A synthesis. *Annals of the Association of American Geographers* 54:2–11.

Burrough, P. A., 1986. *Principles of Geographical Information Systems for Land Resource Assessment.* Clarendon, Oxford.

Carver, S., 1991. Adding error handling functionality to the GIS toolkit. *Proceedings EGIS '91*, pp. 187–196.

Chrisman, N. R., 1989. Error in categorical maps: Testing versus simulation. *Auto Carto 9 Proceedings*, pp. 521–529.

Cliff, A. D., and J. K. Ord, 1973. The choice of a test for spatial autocorrelation. *Display and Analysis of Spatial Data* (J. C. Davis and M. J. McCullagh, editors), Wiley, London, pp. 54–77.

Congalton, R. G., 1988. Using spatial autocorrelation analysis to explore the errors in maps generated from remotely sensed data. *Photogrammetric Engineering & Remote Sensing* 54:587–592.

———, 1991. A review of assessing the accuracy of classifications of remotely sensed data. *Remote Sensing of Environment* 37:35–46.

Fisher, P. F., 1991a. Modeling soil map-unit inclusions by Monte Carlo simulation. *International Journal of Geographical Information Systems* 5:193–208.

———, 1991b. First experiments in viewshed uncertainty: The accuracy of the viewshed area. *Photogrammetric Engineering & Remote Sensing* 57:1321–1327.

———, 1992. First experiments in viewshed uncertainty: Simulating fuzzy viewsheds. *Photogrammetric Engineering & Remote Sensing* 58:345–352.

Goodchild, M. F., and O. Dubuc, 1987. A model of error for choroplethic maps, with applications to geographic information systems. *Auto Carto 8 Proceedings*, pp. 165–174.

Goodchild, M. F., and M.-H. Wang, 1989. Modeling errors for remotely sensed data input to GIS. *Auto Carto 9 Proceedings*, pp. 530–537.

Heuvelink, G. B. M., P. A. Burrough, and A. Stein, 1989. Propagation of errors in spatial modelling with GIS. *International Journal of Geographical Information Systems* 3:303–322.

Lanter, D., 1992. *Intelligent Assistants for Filling Critical Gaps in GIS: A Research Agenda.* Technical Report 92-4, National Center

for Geographical Information and Analysis, Santa Barbara, California.

Lanter, D., and H. Veregin, 1990. A lineage meta-database program for propagating error in geographic information systems. *GIS/LIS '90 Proceedings*, pp. 144–153.

———, 1992. A research paradigm for propagating error in layer-based GIS. *Photogrammetric Engineering & Remote Sensing* 58:526–533.

Lodwick, W. A., 1989. Developing confidence limits on errors of suitability analyses in geographical information systems. *Accuracy of Spatial Databases* (M. F. Goodchild and S. Gopal, editors), Taylor and Francis, London, pp. 69–78.

Moellering, H., 1992. STDS. *ACSM Bulletin* 137:30–34.

Newcomer, J. A., and J. Szajgin, 1984. Accumulation of thematic map errors in digital overlay analysis. *The American Cartographer* 11:58–62.

Openshaw, S., 1989. Learning to live with errors in spatial databases. *Accuracy of Spatial Databases* (M. F. Goodchild and S. Gopal, editors), Taylor and Francis, London, pp. 263–276.

Star, J., and J. Estes, 1990. *Geographic Information Systems: An Introduction*. Prentice Hall, Englewood Cliffs, New Jersey.

Taylor, J. R., 1982. *An Introduction to Error Analysis*. University Science Books, Mill Valley, California.

Tomlin, C. D., 1990. *Geographic Information Systems and Cartographic Modeling*. Prentice-Hall, Englewood Cliffs, New Jersey.

Tomlin, C. D., and J. K. Berry, 1979. A mathematical structure for cartographic modeling in environmental analysis. *Proceedings of the American Congress on Surveying and Mapping*, pp. 269–283.

Veregin, H., 1989a. Error modeling for the map overlay operation. *Accuracy of Spatial Databases* (M. F. Goodchild and S. Gopal, editors). Taylor and Francis, London, pp. 3–18.

———, 1989b. *A Taxonomy of Error in Spatial Databases*. Technical Report 89-12, National Center for Geographical Information and Analysis, Santa Barbara, California.

Walsh, S. J., D. R. Lightfoot, and D. R. Butler, 1987. Recognition and assessment of error in geographic information systems. *Photogrammetric Engineering & Remote Sensing* 53:1423–1430.

Wesseling, C. G., and G. B. M. Heuvelink, 1991. Semi-automatic evaluation of error propagation in GIS operations. *Proceedings EGIS '91*, pp. 1228–1237.

(Received 7 July 1992; revised and accepted 17 February 1993)

Appendix

Description of Program and Associated Algorithms

The program for performing the simulations described in this study was written in C. The code segments documented below use C-based syntax, but their meaning should be clear even to those unfamiliar with the language. The following notes will assist interpretation:

- All statements end with a semi-colon.
- Statements within a loop or dependent on some condition are enclosed in braces.
- Functions are denoted by a pair of empty parentheses. Arguments that need to be passed to these functions are omitted.
- Array subscripts are enclosed in square brackets.
- Array subscripts begin at 0 (rather than 1 as in FORTRAN).
- Two-dimensional arrays corresponding to raster layers (grids) are treated as one-dimensional arrays for computational simplicity.
- Generally, all procedures assume that arrays have previously been initialized to zero.
- Statements enclosed within the symbols /* and */ are comments.

Step 1. Randomly select a grid size based on user-defined minimum and maximum values. For the purposes of the simulations, the grid size selected is assumed to represent the number of rows or the number of columns (i.e., a square grid is assumed). This is easily modifiable to account for rectangular grids.

Step 2. Randomly select a "feature probability" between 0 and 1. This value indicates the probability that a cell will be a member of the set of feature cells.

Step 3. Randomly select a PCC (proportion correctly classified) between 0 and 1. This value indicates the probability that a cell is correctly classified.

Step 4. Randomly select a target level of autocorrelation for feature geometry. The target level is expressed as a standard normal deviate, based on a mean and standard deviation computed as a function of grid size and the feature probability selected in Step 2.

Step 5. Randomly select a target level of autocorrelation for the error distribution. The target level is expressed as a standard normal deviate, based on a mean and standard deviation computed as a function of grid size and the PCC selected in Step 3.

Step 6. Randomly select a buffer size. Because large buffers can quickly saturate a layer, especially when grid size is small and autocorrelation in features is low, the range of buffer sizes is restricted to be less than the grid size.

Step 7. Randomly assign cells to the set of features to be buffered. This is achieved by assigning individual cells based on the feature probability selected in Step 2. The following statement is applied to each cell i in turn.

```
CellValue[i] = floor(random() + FeatureProbability);
```

For each cell, a random number function is used that returns values uniformly distributed between 0 and 1. The number returned by this function is then added to the feature probability selected in Step 2. A floor function is then used to obtain the largest integer not greater than this sum, and this value is assigned to the cell. A cell value of 1 indicates that the cell is a member of the set of feature cells, and a value of 0 indicates that is cell is not a member of this set.

Step 8. Iterate to obtain the target level of autocorrelation, selected in Step 4, for the feature cells. This procedure is based on the swapping algorithm described by Goodchild and Wang (1989). The algorithm simply swaps two cell values at random, and, if the swap results in a level of autocorrelation closer to the target, the swap is retained. The procedure is repeated while the difference between the computed and target autocorrelation levels (NewDifference) is greater than some user-defined threshold value (Threshold). Threshold must be greater than 0 or the program may continue looping in search of a perfect match, which may be unobtainable.

The following code segment documents the procedure more precisely.

```
/* Compute the 1-1 joins count and standard normal deviate. */
JoinsCount = ComputeJoins( );
StdJoinsCount = (JoinsCount - JoinsMean) / JoinsStdDev;
/* Beginning of loop. */
do{
  /* Select two cells using a random number function that returns values uniformly distributed between 0 and a number larger than the number of cells. Use the modulus operator (denoted by the symbol %) to derive a cell number between 0 and one less than the number of cells. */
  cell1 = LargeRandom() % NumCells;
  cell2 = LargeRandom() % NumCells;
  /* Initialize NewDifference for end of loop. */
```

```
NewDifference = 2.0 * Threshold;

/* Continue only if the swap will result in a change, i.e., only if the values
for the two selected cells are different. */

if (CellValue[cell1] != CellValue[cell2]){

  /* Store values associated with the cells to be swapped. */

  OldCellValue1 = CellValue[cell1];
  OldCellValue2 = CellValue[cell2];
  OldJoinsCount = JoinsCount;
  OldStdJoinsCount = StdJoinsCount;

  /* Subtract joins associated with cells to be swapped. */

  JoinsCount = SubtractJoins();

  /* Perform swapping. */

  CellValue[cell1] = OldCellValue2;
  CellValue[cell2] = OldCellValue1;

  /* Add new joins resulting from the swap. */

  JoinsCount = AddJoins();

  /* Recompute standard normal deviate. */

  StdJoinsCount = (JoinsCount - JoinsMean) / JoinsStdDev;

  /* Compute absolute values of differences between joins count and tar-
  get for swapped and pre-swapped cases. */

  NewDifference = abs(StdJoinsCount - JoinsTarget);
  OldDifference = abs(OldStdJoinsCount - JoinsTarget);

  /* If the swapped case is farther from the target than the pre-swapped
  case, reset the pre-swapped values. */

  if (NewDifference >= OldDifference){
    CellValue[cell1] = OldCellValue1;
    CellValue[cell2] = OldCellValue2;
    JoinsCount = OldJoinsCount;
    StdJoinsCount = OldStdJoinsCount;
  }
 }
} while (NewDifference > Threshold);
```

It is sometimes the case, especially when dealing with small grids or a high target autocorrelation, that the procedure will cycle through the loop infinitely, trying to achieve a target autocorrelation that it cannot meet. Thus, it may be necessary to include an alternative stopping criterion for the loop.

Step 9. Randomly assign errors. This is achieved by assigning a value of 1 (error) or 0 (correct) to individual cells based on the PCC selected in Step 3. The procedure exactly parallels that performed to randomly assign cells to the set of features to be buffered (Step 7).

Step 10. Iterate to obtain the target level of autocorrelation for the errors, as defined in Step 5. This procedure is based on the same swapping algorithm described above, except that an array called Errors is processed to create a layer depicting cells that are misclassified.

Step 11. Construct the perturbed layer by merging the reference layer (CellValue) and the layer of misclassifications (Errors). This is achieved using a bitwise XOR (exclusive OR) operator (denoted by the symbol ^) to flip the values in the CellValue array whenever an error occurs in the corresponding cell of the Errors array, i.e.,

```
PerturbedValue[i] = CellValue [i] ^ Errors[i];
```

The array called PerturbedValue represents the perturbed layer.

Step 12. Compute the level of autocorrelation for the feature cells in the perturbed layer.

Step 13. Buffer the reference and perturbed layers based on the buffer size selected in Step 6. This will create two additional arrays, CellBuffer and PerturbedBuffer, which represent the reference and perturbed buffer layers, respectively. A cell is included in the buffer if its distance from a feature cell is less than or equal to the buffer size selected in Step 6.

Step 14. Compute the output PCC in terms of the discrepancies between the reference and buffer layers. The following statement

```
if (CellBuffer [i] != PerturbedBuffer[i]) NumErrors++;
```

is evaluated for each cell i in turn. If the condition is true (i.e., there is a discrepancy) then the variable NumErrors is incremented by one. Once each cell has been evaluated, the output PCC is computed as follows:

```
OutputPCC = 1.0 - NumErrors / NumCells;
```

Step 15. Repeat the steps for another realization.

Howard Veregin

Howard Veregin holds the BA (Hons.) and MA degrees in Geography from the University of Manitoba. He received the PhD degree in Geography from the University of California, Santa Barbara, in 1991. He is currently Assistant Professor of Geography at Kent State University. His main research interests in GIS lie in spatial database accuracy, quality assurance, error propagation, and uncertainty in simulation modeling.