韓國地形空間情報學會論文集 第9卷 第2號 2001年 12月 pp. 17-28

버퍼링 과정에서의 오차전파 측정을 위한 선험 프로젝트 수행

A Pilot Project to Measure Propagated Error in Buffering Process

유기윤*
Yu, Kiyun

要 旨

버퍼링은 근접 관련 공간분석에서 흔히 사용되는 기능의 하나이다. 버퍼링은 불가피하게 새로운 다각형을 생성하게 되는데 이는 버퍼전 객체의 주변에 볼을 굴려서 얻어지는 외곽선으로부터 생성된다. 버퍼링 과정에서 버퍼전 객체 상에 존재하는 오차는 새로이 생성된 다각형인 버퍼 후 객체로 전파된다. 본 논문에서는 버퍼링 연산의 시행시 이와 같은 오차의 전파양상을 선험 프로젝트에 기반하여 분석하여 보았다. 이를 위해 두개의 자료모델인 다중선과 스플라인 곡선모델이 이용되었다. 두 모델을 이용하여 버퍼전 객체상의 오차를 분류하고, 수학적으로 정의하고, 측정하여 보았다. 측정을 위한 선험 프로젝트의 대상은 미국 위스콘신주에 있는 호수의 경계로 하였다.

ABSTRACT

Buffering is one of the popular spatial analytical functions widely used in many proximity analyses. The buffering inevitably entails a new polygon of specified edge that is simulated by rolling a ball around the buffering object. While buffering, the error on the buffering object propagates to the new buffered object. In this paper the error propagation behavior during the buffering operation is analyzed based on a pilot project for two different data models: polyline and spline curve. Thus, the error on the buffered objects are classified, mathematically defined, and measured. For measurements, the pilot project is designed and performed using a test site that is a lake boundary at Wisconsin, USA.

1. Introduction

Buffering is one of the popular spatial analytical functions widely used in many proximity analyses. During buffering, the error on a buffering object usually propagates through the operation in additive or multiplicative fashion. The following section addresses such propagation mechanisms for the situation where two different spatial primitives are used. These two spatial primitives - line segments and spline segments construct polyline and spline curve respectively. The propagation mechanism is classified into three cases based on the error type on the buffering objects, and described mathematically respectively. Then, a pilot project is performed to measure the propagated error.

2. Error Propagation in Buffering

Buffering is a process to delineate a new

* 서울대학교 지구환경시스템공학부 전임강사 Email) Kiyun@plaza.snu.ac.kr

geometric object that is decided by a specified distance scheme from an object. Buffering can be done on each spatial object such as point, line, and polygon delineating a new polygon. The buffering inevitably entails the new polygon of higher smooth edge that is simulated by rolling a ball around the buffering object (Laurini and Thompson, 1992). While buffering, the error on the *buffering object propagates to the new buffered object*.

2.1 Delineation of Buffered Object

The buffering process consists of two steps. The first step is to stretch out towards the outside direction from the buffering object within a specified distance. The direction is orthogonal to the direction of the spatial object comprising the buffering object. The next step is simply drawing a line that touches the outer limit of this stretch. For the line segment case, coordinates of a number of points on the outer limit are required to delineate the buffered object. For the spline segment case, the curve equations must be derived using these points. In most cases, the buffered object requires more spline curve equations than the buffering object.

2.2 Conceptual Model of Error Propagation

Position Error Propagation on a Buffered Line Object

The propagated position error on the buffered line object depends on the position error on the buffering line object. Provided the position error of the buffering object of a line segment is expressed by equation (1),

$$e_i^{ln} = e_m + e_g \tag{1}$$

where, e_i^{ln} indicates position error on the line segment of the buffering object.

indicates source map error

indicates generalization error.

Now, the buffered object consists of multiple line segments including upper and lower line segments and side line segments. Then, the position error on these line segments of the buffered object in this case is,

$$e_i^{ln} = e_m + e_g \tag{2}$$

where, e_i^{ln} indicates position error on the line segment of the buffered object.

Equation (2) is a reasonable estimation when a series of line segments draw a fairly straight line. However, if the line segments draw curves, the error on the buffered object may not strictly follow such an equation. Rather, the error would be smaller due to the reduced sinuosity of buffered line segments.

Likewise, if the buffering object is in a spline segment, the position error of the corresponding spline segments of the buffered object is as follow.

$$e_i^{sp} = CR_{MVD}^{-1} e_g + e_m \tag{3}$$

$$e_i^{sp} = CR_{AD}^{-1} e_g + e_m \tag{4}$$

where, e_i^{sp} indicates position error on the spline segment of the buffered object

CR_{MVD}^{-1} indicates inverse CR of MVD

CR_{AD}^{-1} indicates inverse CR of AD.

Here, the *CR* indicates closeness ratio. The *CR* measures relative performance of spline segments by dividing the MVD (or AD) of line segments by MVD (or AD) of spline segments. The MVD measures maximum vector displacement of a spline segment (or a line segment) from the corresponding linear entity on which digital abstraction is proceeded. Whereas the AD measures area displacement between a spline segment (or a line segment) and the corresponding linear entity on which digital abstraction is proceeded. For detailed description of these measures, refer to the research results by McMaster (1983) and Kiyun (1999)

Again, these estimations are reasonable when the spline segments draw fairly straight lines. If they draw curves, the error would be larger along the outer buffered side of the curve due to line sinuosity characteristics. Equations (2), (3), and (4) show the propagated position error on the buffered object. The amount of propagated error on the buffered object basically depends on the amount of error on the buffering object. Provided the error by equation (2) is uniform within a line segment and between line segments, such error results in a certain width of error band that is uniform along the polyline. The same is true for the equation (3) and (4).

Area Error Propagation

Case a) Buffering is Done on Both Inside and Outside of a Buffering Object

Compared to this, the area error propagation is somewhat different. When the buffering of distance m is done on both the inside and outside of buffering object of line segments with n points (for closed polygon) or n+1 points (for open polygon), the area error of the buffered object is calculated as follows, provided the same buffer distance is applied on the linear entity of same span. Assume that the area of buffered object from the line segments is A_{bd}^{ln} and the area of the buffered object from the linear entity of same span is A_{bd}^{LE}. then,

$$A_{bd}^{ln} = 2m\sum_{i=1}^{n}\Delta_i^{ln} + \pi m^2 \quad (5)$$

$$A_{bd}^{LE} = 2m\sum_{i=1}^{n}\Delta_i^{LE} + \pi m^2 \quad (6)$$

where, A_{bd}^{ln} is area of buffered object from the line segments

A_{bd}^{LE} is area of buffered object from the linear entity

Δ_i^{ln} is length of ith line segment

Δ_i^{LE} is length of linear entity within ith line segment span

is buffer distance.

In the above equations, the πm^2 term is excluded if the line segments make a closed polygon. In actual calculations, the values A_{bd}^{ln} and A_{bd}^{LE} would be smaller than the above estimations because there is some overlapping of the buffered area due to the line sinuosity of the buffering object. If the line sinuosity of the buffering object increases, the reduction would be larger and vice versa. This will be explained later.

Then, the area error of the buffered object from the line segments can be calculated by difference of areas of these two buffered objects,

$$E_{bd}^{ln} = A_{bd}^{LE} - A_{bd}^{ln} = 2m\sum_{i=1}^{n}\left(\Delta_i^{LE} - \Delta_i^{ln}\right) \quad (7)$$

where E_{bd}^{ln} is area error of the buffered object from the line segments.

Equation (7) explains the propagated area error during buffering when the line segments are used. From this equation, the propagated area error is decided by the difference of two line lengths (linear entity length and polyline length) multiplied by the double of the buffer distance.

Likewise, if the spline segment is used instead of the line segment, by the same mechanism, the propagated area error would be,

$$E_{bd}^{sp} = 2m\sum_{i=1}^{n}\left(\Delta_i^{LE} - \Delta_i^{sp}\right) \quad (8)$$

where, E_{bd}^{sp} is area error of buffered object from the spline segments

Δ_i^{sp} is length of ith spline segment.

Case b) Buffering is Done on Outside of a Buffering Object

Now, let' s think about the case when the buffering is done on the outside of object of n line segments with n points. Such line segments make a closed polygon of area A_{bg}^{ln} that can be calculated as,

$$A_{bg}^{ln} = \left(\sum^{n-1} p_i p_{i+1} + p_n p_1\right) / 2 \quad (9)$$

where, $p_1, p_2, \cdots p_n$ indicate point vectors.

The corresponding area error of the buffering object of line segments is,

$$E_{bg}^{ln} = A_{bg}^{LE} - A_{bg}^{ln} = \sum_{i=1}^{n} e_i^{ln} \quad (10)$$

where, E_{bg}^{ln} is the area error of the buffering object

A_{bg}^{LE} is the area of the buffering object of linear entity

A_{bg}^{ln} is the area of the buffering object of line segments

e_i^{ln} is the area error at ith line segment which is positive if the line segment is on outside of the linear entity and negative if on inside of the linear entity.

Using the coefficient C that varies depending on the geometric characteristic of the buffering object, the area error of the buffered object is calculated as follows,

$$E_{bd}^{ln} = C^{ln} E_{bg}^{ln} \quad (11)$$

where, E_{bd}^{ln} is the area error of the buffered object of line segments

C^{ln} is the coefficient

E_{bg}^{ln} is the area error of the buffering object of line segments.

This coefficien C^{ln} reflects the geometric characteristics of the buffering object and it may not be easy to formulate its behavior mathematically. One behavior that can be expected is that it would increase as the buffer distance increases. Thus,

$$C^{ln} \propto m \quad (12)$$

where m is the buffer distance.

This is the propagated area error when the line segment is used. Likewise, the propagated area error when the spline segment is used is,

$$E_{bd}^{sp} = C^{sp} E_{bg}^{sp} \quad (13)$$

where, E_{bg}^{sp} indicates area error of buffering object of spline segments,

$$E_{bg}^{sp} = \sum_{i=1}^{n} e_i^{sp} = \sum_{i=1}^{n}\left(C R_{AD}^{-1} e_g + e_m\right) \quad (14)$$

where, C^{sp} is the coefficient for the spline curve

E_{bd}^{sp} is the area error of the buffered object of spline segments.

3. A Pilot Project to Measure Propagated Error in the Proposed Model

In this section, the propagated position and area error during the buffering operation is measured on a pilot project. The main purpose of such measurement is to compare the values of measured propagated error with the values from the proposed mathematical model so that deepen the intuitive understanding of the mathematical model.

3.1 Project Site Selections and Testing Data Generation

For project sites, a hydrologic feature from a USGS 7.5 quadrangle map was selected (USGS quad map, Boulder Junction SW, Wisconsin, N46000-W8937.5/7.5). The selected hydrologic feature was a lake boundary of Trout Lake, Vilas County, Wisconsin, USA (Figure 1).

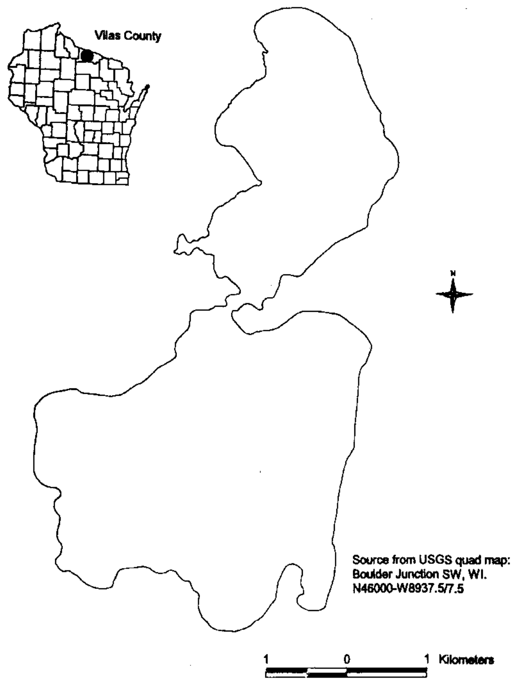

Figure 1. A Selected Testing Site for a Pilot Project (Wisconsin, USA)

Since the selected lake boundary was in paper format, it was scanned with 500dpi resolution and was head-up digitized very precisely in ArcInfo (v.7.0.4) in the Unix environment. Thus, the paper format was converted to the digital format. The digital format Trout Lake was transformed to register to the ground coordinate system. The ground coordinate system used was Universal Transverse Mercator (UTM) in meters.

Upon completion of the scanning and converting to digital format of the lake boundary, manual digitization was done. While digitizing, two indices were employed for digitization rule. These indices were number of intersection (NIT) and number of inflection (NIF). The NIT indicates how many times each line segment crosses over the linear entity. The NIF indicates how many times the linear entity within each line segment span inflects (Figure 2). For this project, digitization rule with some exceptions due to minor fluctuations is applied because this rule lets the spline curves simulate the linear entity best.

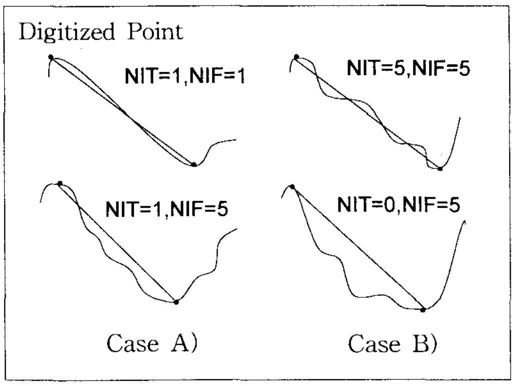

Figure 2 Relationship Between NIT, NIF, and Digitized Points

Note 1. The straight line indicates polyline whereas wiggled line indicates linear entity

2. The Case A) shows two linear entities digitized by two points with same NIT(1) and different

NIF(1-upper, 5-lower); the Case B) shows the same NIF(5) and different NIT(5-upper, 0-lower).

A polyline with 130 line segments was produced. Then, the New Bezier spline interpolation algorithm (Kiyun, 1999) was used to generate a corresponding spline curve. Thus, one spline curve was generated that consisted of 130 spline segments. The generated spline segments were primarily in the cubic polynomial equation format. If the new topological data model was implemented in terms of coding the appropriate program, the curve equations could be used directly for any of the following spatial analysis operations. However, at this stage, it is still in the conceptual realm. Instead of using a program with curve equations, a series of coordinate sets on the curve locus were generated and imported into ArcInfo to generate a line coverage. Then, the coverage from the spline curve was used for the following analysis.

3.2 Error Representation of a Polyline and a Spline Curve

The error on the abstracted objects, such as polyline or spline curves, can be represented by putting a width of error band (Chrisman, 1989) along the object. Normally, the width of error band is determined by checking the coordinates of selected points on the object with corresponding coordinates on the independent source of higher accuracy (FGDC, 1994).

In this project, to determine the width of error band, the entire error on the object is grouped into two categories; source map error and generalization error. With the source map error, the error level specified on the *National Map Accuracy Standard* (NMAS) is used because the USGS quadrangle map was compiled satisfying the standard. According to the standard, no more than 10% of the errors on a map exceed 1/50 inch for scale 1:20,000 or smaller. Thus, in this project case, more than 90% of errors on the quadrangle map are within 1/50 inch because the map scale is 1:24,000. On the ground, 1/50 inch error on the map corresponds to 12.2 meter.

To determine the error during generalization, the lake boundary and the polyline (or the spline curve) were examined. Each line segment span was fragmented into 50 smaller segments and the deviation between these two lines at each smaller segment span was calculated. Then, the mean of the calculated 50 deviations for each line segment was calculated. The same rule was applied when the spline segments replaced the line segment. The calculation results for the line segments showed that the deviation distribution is likely normal (Figure 3). Similar results are acquired from the spline segments using the New Bezier algorithm.

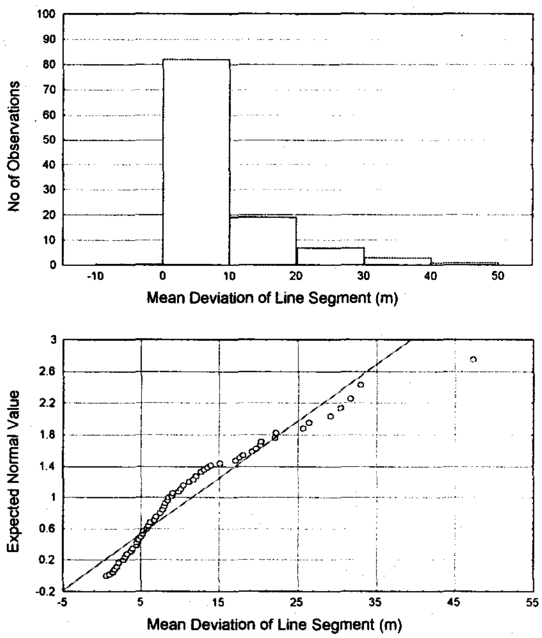

Figure 3 Distribution of Mean Deviation of Line Segment

From this figure, the reason that deviation is only on the positive side is because the sign is not considered and all deviations were measured as positive values. If we assign either positive or negative sign to the measured deviation based on whether it is on left side or right side along the line segment, the negative side of the above distribution would most likely be the same with the positive side. Based on this assumption, the overall mean deviation of the line segments and the spline segments from the New Bezier algorithm were calculated with their standard deviations.

Table 1 summarizes the results. In this table, the upper confidence limit is calculated at significance level 90%. The reason 90% was used was to conform the error level of the tested polyline or spline curves to the error specification of the NMAS.

Thus, at 90% probability, the lake boundary within line segment span falls into these confidence limits. Table 2 shows performance comparison results of the polyline and the spline curve using mean of raw deviation and CR.

Table 1 Mean Deviation Examination Results for Deciding Error Band Width: Deviation of Polyline/ Spline Curve from the Lake Boundary

	St.Dev.(m)	*Upper Confidence Limit (=0.1) (m)*
Polyline	17.771050	29.322233
Spline Curve from New Bezier	7.454549	12.300006

Note Confidence interval is calculated using Z=1.65 for =0.1 (significance level 90%).

Table 2 Performance Comparison Results of Polyline and Spline Curves: Using Mean of Raw Deviation and CR

	Mean of Raw Deviation		*Mean of CR*	
	MVD(m)	*AD(m2)*	*MVD*	*AD*
Polyline	14.983	2566.528		
Spline Curve from New Bezier	9.9354	1392.744	1.893	2.274

Note 1. *MVD* indicates the maximum vector displacement of the digitized polyline from the linear entity; *AD* indicates the area displacement of the digitized polyline from the linear entity. Values in the table are mean values for 130 line segments. The same is true for the New Bezier spline curve.
2. CR indicates closeness ratio that is calculated by dividing *MVD* (or *AD*) of polyline by *MVD* (or *AD*) of New Bezier spline curve. Values in the table are mean values for 130 line segments.

Using the above estimation of the source map error and generalization error and the equation (1), the width of the error band for the polyline can be calculated as follows,

$$e_i^{ln} = e_m + e_g = 12.2+29.3 = 41.5 \text{ (m)}.$$

Likewise, for the spline curve of the New Bezier, the width of the error band is,

e_i^{sp} = 12.2+12.3 = 24.5 (m).

If one compares this calculation with the theoretical value from equation (3),

$e_i^{sp} = e_m + CR_{MVD}^{-1} e_g$ = 12.2+1.893-1?9.3 = 27.7 (m).

where, CR_{MVD}^{-1} = 1.893^{-1} (from Table 2). This computation approximates the empirical values.

Figure 4 shows these two error bands applied on the polyline and the spline curve of the New Bezier respectively.

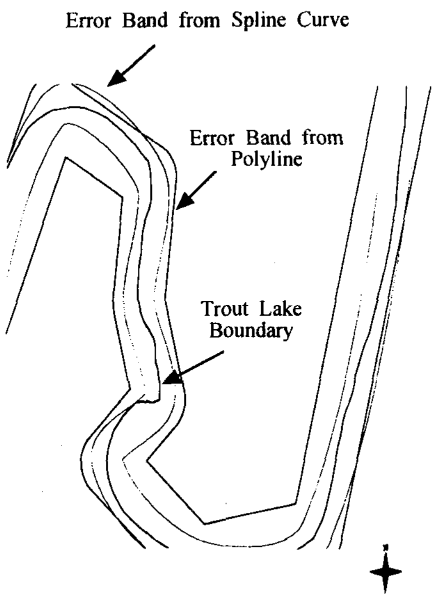

Figure 4. Comparison of Two Different Error Bands: One of the Polyline (41.5m) and the Other of the Spline Curve from New Bezier (24.5m)

3.3 Error Propagation in Buffering

As previously explained, there are two different error propagation models in buffering; one is position error propagation and the other is area error propagation. In this section, first the position error propagation is examined and then, the area error propagation is examined.

Position Error Propagation During Buffering

The propagated position error during buffering process is evaluated by checking the error band width of buffered objects. To examine the error band of a new buffered polyline and spline curve (or buffered objects), buffering was done on the buffering polyline and spline curve (or buffering objects) and the error band width of the buffered objects were calculated. Then, the calculated error band width of the buffered objects was compared with the error band width of the buffering objects. For buffering distance, the values were arbitrarily decided as 100m, 200m, and 300m. The results show that as the buffer distance increases, the standard deviation and corresponding upper confidence limit of the buffered object slightly reduces (Table 3). These results are the same for both the polyline and spline curve from the New Bezier.

Table 3 Deviation Examination Results for Deciding Error Band Width : Deviation of Buffered Polyline/ Spline Curve from the Buffered Lake Boundary

	Buffer Distance (m)	*St. Dev. (m)*	*Upper Confidence Limit (=0.1) (m)*
Polyline	*100*	17.65701	29.13407
	200	17.52698	28.91952
	300	17.01082	28.06785
Spline Curve of New Bezier	*100*	6.43768	10.62217
	200	5.96257	9.83824
	300	5.63036	9.29009

Note Confidence interval is calculated using Z=1.65 for =0.1 (significance level 90%).

The reason for these results comes from the line sinuosity changes during buffering. During buffering, complex lines change to simpler lines, reducing the line sinuosity of the buffered object. In this context, the deviation between the buffered polyline (or spline curve) and the buffered linear entity is reduced. Consequently, the standard deviation is reduced as well. Figure 5 and Figure 6 show the position error propagation behaviors during buffering of three buffer distances for polyline and spline curve cases.

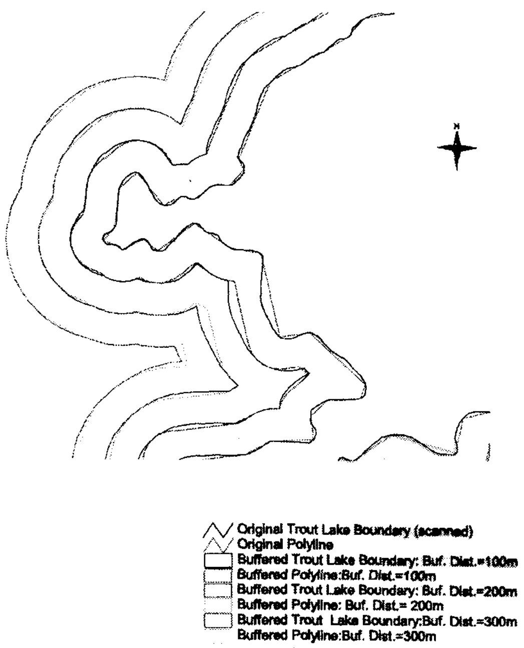

Figure 5 Position Error Propagation During Buffering: Polyline Case

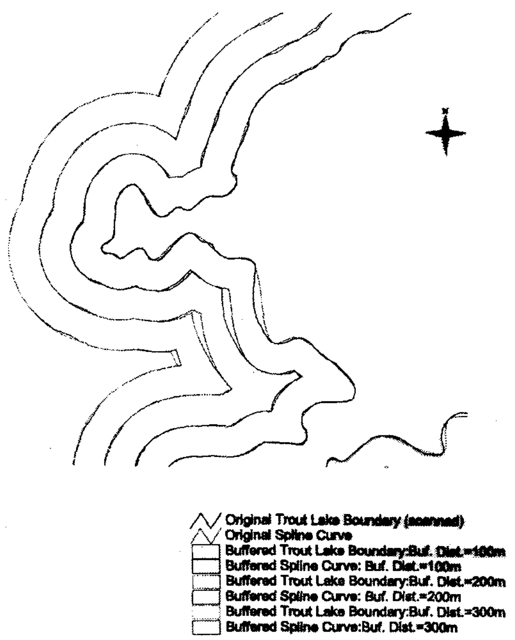

Figure 6. Position Error Propagation During Buffering: Spline Curve Case

Area Error Propagation During Buffering

The propagated error in area calculation during buffering can be scaled down to two different cases; 1) buffering is done on both inside and outside and 2) buffering is done only on outside.

Case 1) Buffering is done on both inside and outside

When buffering is done on both inside and outside of polyline (or spline curve), the area of buffered polyline (or spline curve) may be different from the area of buffered lake boundary. In this case, the error in buffered area calculation can be determined by subtracting the buffered area of the polyline (or spline curve) from the buffered area of the lake boundary.

For testing this, 100 m buffer distance was used to buffer the polyline (or spline curve) and the lake boundary. Then, the differences of their buffered areas were calculated. The results show that the difference of buffered polyline and buffered lake boundary is bigger than the difference of buffered spline curve and buffered lake boundary (Table 4). To compare these values with theoretical values from equation (7) and (8), area error was calculated using these equations. The calculated results show that buffered area of the lake boundary, polyline, and spline curve are 5,498,288.800-(a), 5,361,521.000-(b), 5,429,425.200-(c) respectively, and the difference between (a) and (b) is 136,767.000, and the difference between (a) and (c) is 68,863.600. In these calculations, the trend of area difference of the buffered polyline and buffered spline curve from the buffered lake boundary follows the trend of area difference of the real testing values. As previously explained, the real values from test are smaller than the theoretical values due to the effect of overlap of the buffered area from the line sinuosity.

Table 4 Area Error Propagation During Buffering: Buffering is Done on Both Inside and Outside

	Buffered Area (m2)	*Difference (m^2):(a)-(b)*	*Difference (m^2):(a)-(c)*
Lake Boundary(a)	5369364.500		
Polyline(b)	5241089.000	128275.500	
Spline Curve of New Bezier(c)	5323056.000		46308.500

Case 2) Buffering is done only on outside

When the buffering is done only on the outside of polyline (or spline curve), the area of the buffered polyline (or spline curve) includes the area within the original polyline (or spline curve) and the increased area by the buffering. Such area may be different from the area from buffering the lake boundary. Thus, the area difference results in error.

For the testing, a 100m buffer distance was applied along the outside of the polyline (or spline curve) and the lake boundary. Then, the area differences were calculated and compared. Table 5 summarizes the testing results. From this table, the area error from the buffered polyline was bigger than the area error from the buffered spline curve.

Table 5 Area Error Propagation During Buffering: Buffering is Done Only on Outside

	Lake Boundary (a)	*Polyline (b)*	*Spline Curve of New Bezier (c)*
Buffered Area (m^2)	18618590.000	18442202.000	18605050.000
Difference (m^2):(a)-(b)		176390.000	
Difference (m^2):(a)-(c)			13540.000

To compare these calculations with the theoretical values from the equation (11) and (13), the area coefficient should be calculated first. However, the calculation of this coefficient is not feasible due to the difficulty in mathematically formulating the geometric characteristics of the buffering polyline (or spline curve). One aspect we can expect about the coefficient is that it may increase as the buffer distance increases. As the equation (12) shows,

$$C \propto m$$

where, C is the coefficient.

To check behavior of C, real values were calculated for three different buffer distances, 100m, 200m, and 300m for the polyline and spline curve cases. The calculation of the C was done using the equation (11) and (13). From these equations, the C can be calculated as follows,

$$C^{ln} = \frac{E^{ln}_{bd}}{E^{ln}_{bg}} \quad \text{for polyline case}$$

$$C^{sp} = \frac{E^{sp}_{bd}}{E^{sp}_{bg}} \quad \text{for spline curve case}$$

where, C^{ln} and C^{sp} indicate the coefficient for polyline and spline curve case respectively. Table 6 summarizes the calculation results. From this table, it is clear that the C coefficient increases as the buffer distance increases.

Table 6 Calculation of C Coefficient for Different Buffer Distance

	Buffer Distance (m)	*$E_{bd}(m^2)$*	*$E_{bg}(m^2)$*	*C*
Polyline	100	176390.000	112609.000	1.566393
	200	203966.000	112609.000	1.811276
	300	206714.000	112609.000	1.835679
Spline Curve of New Bezier	100	13540.000	7855.000	1.723743
	200	21778.000	7855.000	2.773775
	300	23214.000	7855.000	2.955315

From these two tables (Table 5 and Table 6), it appears that the area error of the buffering object is related to the area error of the buffered object. One more important point is that even though the error band during the buffering operation is reduced slightly (Table 4), the area error has increased significantly after buffering.

4. Summary and Conclusions

This paper is dedicated to identify the propagated error during buffering operation based on a pilot project. For this, two different data models, polyline and spline curve, are selected, which are constructed from multiple line segments and multiple spline segments respectively. On these two data models, the possible error during buffering was mathematically explained and tested. For testing, a pilot project was performed using the Trout Lake boundary, Vilas County, WI, USA. The experiment results showed that the propagated position error and area error on the buffered objects are significantly rely on the error of the buffering objects.

5. References

Caspary, W. and Scheuring, R. (1992). Error-Bands as Measures of Geometrical Accuracy. Proceedings of 3rd European Conference on Geographical Information System '92. Vol.1, Munich Germany.

Chen Zhong-Zhong and Finn John T. (1994). The Estimation of Digitizing Error and Its Propagation with Possible Application to Habitat Mapping. Proceedings: International Symposium on Spatial Accuracy of Natural Resource Databases. Williamsburg, VA.

Chrisman, N. R. (1989). Error in Categorical Maps: Testing versus Simulation. AutoCarto 9, pp.521-529.

Chrisman, N. R. and Lester, M. (1991). A Diagnostic Test for Error in Categorical Maps. AutoCarto 10.

Chrisman, N. R. and Yandell, B. S. (1988). Effects of Point Error on Area Calculations: A Statistical Model. Surveying and Mapping, Vol. 48, No. 4.

Kiyun Yu (1998). Exploring a New Line Model for More Accurate Spatial Representation in Geographic Information Systems. Ph.D. Thesis, University of Wisconsin-Madison, WI.

Kiyun Yu(1999). Evaluation of a New Line Model for More Accurate Spatial Representation, GEOMATICA, Vol. 53, No. 1.

Lawrence, V. S., Dewitt, B. A., and Shrestha, R. L. (1996). Estimating Positional Accuracy of Data Layers within a GIS through Error Propagation. Photogrammetric Engineering & Remote Sensing, Vol.LXII, No.4.

McMaster R. B. (1983). Mathematical Measures for the Evaluation of Simplified Lines on Maps, Ph.D. Thesis, University of Kansas.

McMaster R. B. (1986). A Statistical Analysis of Mathematical Measures for Linear Simplification, The Americal Cartographer, Vol. 13, No. 1, pp. 103-116.