IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 25, 2020, accepted February 2, 2021, date of publication February 8, 2021, date of current version February 18, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3057719

Generalized Buffering Algorithm

GUOQING ZHOU[1,3], (Senior Member, IEEE), RONGTING ZHANG[2], AND SHENGXIN HUANG[4]

[1]Guangxi Key Laboratory for Spatial Information and Geomatics, Guilin University of Technology, Guilin 541004, China
[2]College of Geomatics Science and Technology, Nanjing Tech University, Nanjing 211816, China
[3]Center for Remote Sensing, Tianjin University, Tianjin 300072, China
[4]Guangxi Water and Power Design Institute Company Ltd., Nanning 530000, China

Corresponding author: Rongting Zhang (zrt@njtech.edu.cn)

This work was supported in part by the National Natural Science of China under Grant 41961065 and Grant 41431179, in part by the Guangxi Science and Technology Base and Talent Project under Grant Guike AD19254002, in part by the Guangxi Innovative Development Grand Program under Grant GuikeAA18118038 and Grant GuikeAA18242048, in part by the Guangxi Natural Science Foundation for Innovation Research Team under Grant 2019GXNSFGA245001, in part by the Guilin Research and Development Plan Program under Grant 20190210-2, in part by the National Key Research and Development Program of China under Grant 2016YFB0502501, and in part by the BaGuiScholars program of Guangxi.

ABSTRACT The existing buffers algorithms cannot effectively to meet the demands of high accuracy of buffer analysis in practice although many efforts have been made in the past 60 years. A generalized buffering algorithm (GBA) is presented, which considers the geometric distance and the attribute characteristics of all instances within buffer zone. The proposed algorithm includes three major steps: (1) select and initialize target instance; (2) determine buffer boundary points through mining homogeneous pattern; (3) ''smoothly'' connect buffer boundary points to generate the generalized buffer zone. The details for the generations of the generalized point buffer (GPIB) zone, the generalized line buffer (GLB) zone, and the generalized polygon buffer (GPLB) zone are discussed. Two dataset are used to validate the performances of the proposed GBA. Six parameters are applied as indexes to evaluate the proposed algorithm. The experimental results discovered that (1) the GBA is close to the tradition buffering algorithm (TBA) when the angle increment ($\Delta\varphi$) in GPIB, line increment (ΔL) in GLB, and arc length increment (ΔS) in GPLB approach to zero, respectively; (2) the proposed GBA can accurately reflect the real situation of the buffering zone, and improve the deficiency and accuracy of TBA in real application.

INDEX TERMS Buffering zone, data mining, geographic information science, homogeneous pattern, spatial analysis.

I. INTRODUCTION

The research for buffering algorithms can be *fully* retroactive to 1960s [1], which Europe first introduced statistics into geography. The traditional buffering algorithm includes point elements, line elements, and polygon element buffering. Since 1960s, many of novel algorithms have been developed. For example, in the early 1975, [2] first proposed buffer creation algorithm using computational geometry, and then [3] and [4] extended this method. [5] proposed an improved algorithm for generation of line buffering zone, called ''geometric model for both-side parallel lines buffer generation''. This method was presented on the basis of a comparison analysis of the two methods of line buffer generations, angle bisector method and circular arc method, and can effectively solve the problems such as determination of convex and concave of a chord arc. [6] proposed a vector buffer algorithm for point, line and polygon elements using buffer curve and edge-constrained triangulation network. This method can efficiently reduce the computational complication in the process of cutting and re-organizing the buffer boundary line or/and curve. [7] proposed a method called Voronoi k-order neighbors to recreate the buffer area. [8] improved the buffer generation algorithm the double parallel lines and arcs using the rotation transformation and the recursive. This method can largely simplify the process of line buffer generation, meanwhile efficiently corrects the acute corners of the boundary of the area buffer. This method was considered as the better solution for the intersection of line buffer generation. In order to reduce the time-consuming during generating the point buffering in the self-intersection section, [9] proposed the application of the Douglas-Peuker algorithm to

The associate editor coordinating the review of this manuscript and approving it for publication was Senthil Kumar.

extract the feature points of the curve before the buffer was generated. The experimental result demonstrated that this algorithm can largely decreased computational time. [10] proposed a novel method in which the buffer distance is variable during generating line buffers. However, the variable buffer distances are indeed an equidistant, as a result, this method have met challenges for the complicated and irregular regions with various geographic attributes. [11] proposed a random algorithm for buffer zone generation. This algorithm can be carried out using computer parallel processing, consequently, largely improves the efficiency of buffer zone generation. [12] conducted a comprehensive overview for various buffer algorithms, analyzed and compared the advantages and disadvantages of these early methods, and concluded the most propitiate solution to the problems such as distortion and correction of the acute corners during the process of buffer generation. [13] proposed the convex arc algorithm for solution of the problems of self-intersection and acute corner correction. This method can effectively solve the problem of area buffer generation with multiple embedded inner rings. [14] proposes a buffering approach for matching areal objects (e.g. buildings) on the basis of relaxation labeling techniques. This method has successfully applied in pattern recognition and computer vision. [15] proposed the function form of the buffer expression varying with a certain variable, which effectively solved the problem of determining the complex boundary of the buffer on both sides of the convex and concave inflection points under the variable buffer distance. [16] proposed a vector grid hybrid algorithm to solve the problem of buffer generation for line targets. First convert the vector data to a raster format, secondly use the Douglas-Peuker algorithm to resample the line, then generate a buffer based on the expansion principle, and finally deal with the problem of buffer self-intersection. [17] used GPU (Graphic Processor Unit) to calculate the distance between vector data, which can quickly realize the cache analysis of vector data, and also solve the problem of buffer boundary self-intersection. This method has successfully been applied in three-dimensional (3D) digital earth. [18] proposed "a parallel buffer algorithm to improve the performances of buffer analyses on processing large datasets," which was based on area merging and massage passing interface. But, in the algorithm, the relationship of adjacent vector features was not considered. [19] proposed a new method, which first classify attribute data, and then establish multiple temporal buffers, finally smooth the boundary line using Douglas-Peuker algorithm and Bézier curve. [20] proposed a parallel algorithm on the basis of equal arc segmentation method, which first segments a whole arc into many segments, and then take the segments as a unit, and finally, deal with each of the units, respectively. [21] proposed buffer generation method on the basis of a Geodesic of tiles. The buffer distances on both sides are different and are measured using geodetic facet. [22] present a buffering method using the probability and entropy theory. This method can effectively reduce the error of spatial analysis and avoid the error of spatial decision. [23] used a diagram of log distance and polar angle bins to compute the correlation. [24] proposed a buffer generation method on the basis of a combination of spatial index and data transformation in order to speed up data processing.

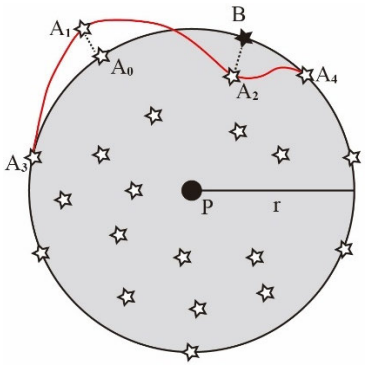

FIGURE 1. Traditional point buffer zone (surrounded by the arc of $A_0BA_4A_3A_0$) and Generalized point buffer zone (surrounded by the arc of $A_1A_2A_4A_3A_1$).

Although many efforts have been made over 60 years, the traditional buffering methods have encountered many challenging in practice due to increasing requirement in buffering zone accuracy. This is because the existing buffering method are based on a fixed buffer distance without considering the differences of attributes within buffer zone, *i.e.,* the existing methods take all of attributes within buffer zone are the same, i.e., homogenous. As a result, the resulted buffer zone deviates from the "true" buffer zone and lose accuracy. For example, as shown in Figure1, given point P and white star targets (such as A_0, A_1, etc.) are homogeneous patterns. The boundary of traditional buffer zone of point P is the arc of $A_0BA_4A_3A_0$, of which the buffer radius is a fixed value r. However, the attributes of black star target B in traditional point buffer zone is heterogeneous with the other targets. Additionally, the white star A_1 outside the buffer zone also is a homogeneous with the others. Therefore, the buffer zone of point P should NOT be a regular shape, while be the irregular shape, i.e., the zone surrounded by the arc of $A_1A_2A_4A_3A_1$, which is more in line with the actual situation. For this reason, a radical algorithm, called "Generalized buffering algorithm (GBA)", is proposed for buffer zone generation.

II. GENERALIZED BUFFERING ALGORITHM

A. A GROUP OF DEFINITIONS

In order to describe the generalized buffering algorithm, a group of definitions are first given below.

Definition 1: Spatial objects *are with different features in space, noted* $O = \{O_1, O_2, \ldots, O_n\}$, which presents the set of different categories of objects. For example, Figure 2 has five types of spatial objects: A, B, C, D, and E. One instance is the object which is in the specific location. The set of instances can be expressed by $\mho = \mho_1 \cup \mho_2 \cup \cdots \cup \mho_n$, where $\mho_i$ $(1 \leq i \leq n))$ is the instance set of object O_i. In order to distinguish different instances of object, every instance has a unique ID, so every instance's information contain {instance ID, object

TABLE 1. The instance information for five types of spatial objects in Figure 2.

Object type	Instances of Spatial Object
A	A_1, A_2, A_3, A_4
B	B_1, B_2, B_3, B_4, B_5
C	C_1, C_2, C_3
D	D_1, D_2, D_3, D_4
E	E_1, E_2, E_3, E_4, E_5, E_6

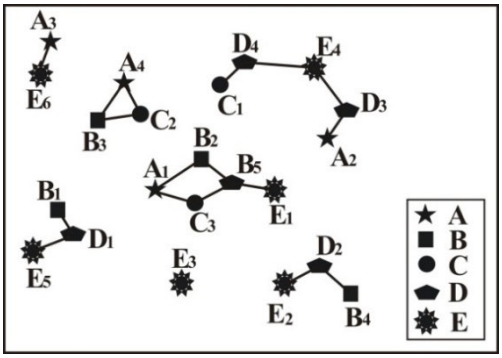

FIGURE 2. Spatial objects with their homogenous clustering patterns, where Ψ_i represents *i*-th instances with feature type Ψ; *i* presents the unique ID inside each spatial object type. Connection lines between instances represent the *R*-relationship.

type, spatial location, attributes}. The instances information of objects in Figure 2 is described in Table 1.

Definition 2: ***R-proximity relationship*** *describes the proximity relationship among instances. Proximity relation can be expressed by spatial topological relationship such as intersection and join, distance relation (Euclidean distance), mixed relation (shortest path between two points in the map), and so on.* For example, in Figure 2, if and only if the Euclidean distance of A_3 and E_6 is less than or equal to the distance threshold, there exists *R*-proximity relationship between them, i.e., $\{if\,[\text{distance}(A_3, E_6)] \leq d\} \Leftrightarrow R(A_3, E_6)$.

Definition 3: If R-proximity neighborhood instance set $\mathbf{\Gamma}$ *contains all types of homogenous pattern* $\mathbb{C}$ *and no proper subset does so, so* $\mathbf{\Gamma}$ *is a* row ***instance*** *denoted by row_instance* ($\mathbb{C}$). For example in Figure3, $\{A_4, B_3, C_2\}$ is a row instance of homogeneous pattern $\{A, B, C\}$, but $\{A_1, B_2, B_5, C_3\}$ is not a row instance of homogeneous pattern $\{A, B, C\}$, since its proper subset $\{A_1, B_2, C_3\}$ or $\{A_1, B_5, C_3\}$ can contain all object types of homogenous pattern $\{A, B, C\}$.

Definition 4: ***Table instance*** *expressed by* table_instance ($\mathbb{C}$) *is the set of all row instances of homogeneous pattern* $\mathbb{C}$. For example, table_instance($\mathbb{C}$) $= \{\{B_1, D_1\}, \{B_4, D_2\}\}$ is the table instance of $\mathbb{C} = \{B, D\}$. The length of a homogeneous pattern, c is called the order of homogeneous pattern. It is called size(c) $= |c|$. For example, size($\{A, B, C\}$) $= |3|$.

Definition 5: ***Participation rate (PR)*** *is a specific value which obtained by the number of* O_i'*s instances which don't repeatedly present in the table instance of co-location pattern* $\mathbb{C}$ *dividing the total number of instance of* O_i. *It can be expressed by*

$$PR(\mathbb{C}, O_i) = \frac{\left|\pi_{o_i}(table\_instance(\mathbb{C}))\right|}{|table\_instance(O_i)|} \tag{1}$$

where π_{o_i} *is the projection of relationship.*

Definition 6: ***Participation index (PI)*** *is the minimum value of participation rate of all objects of co-location pattern* $\mathbb{C}$. *It can be denoted by*

$$PI(\mathbb{C}) = min_i^k\{PR(\mathbb{C}, o_i)\} \tag{2}$$

Definition 7: If $PI(\mathbb{C})$ is greater than or equal to the threshold $\exists$, homogeneous pattern $\mathbb{C}$ is a **prevalent homogeneous pattern**. In Figure 3, for homogeneous pattern $\mathbb{C} = \{B, D\}$, table_instance($\mathbb{C}$) $= \{\{B_1, D_1\}, \{B_4, D_2\}\}$ is the table instance of $\mathbb{C}$. So, $PR(\mathbb{C}, B) = 2/5 = 0.4$, $PR(\mathbb{C}, D) = 2/4 = 0.5$. Thus, $PI(\mathbb{C}) = \min\{PR(\mathbb{C}, B), PR(\mathbb{C}, D)\} = 0.4$. If the threshold is $\exists = 0.3$, so $PI(\mathbb{C}) = 0.4 > \exists = 0.3$. According to above analysis, $\mathbb{C} = \{B, D\}$ is a prevalent homogeneous pattern.

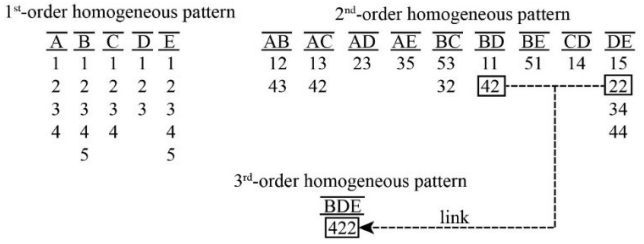

FIGURE 3. The generation process of table instances and 3rd-order homogeneous pattern.

Taking a 3rd-order homogeneous pattern as an example to explain the generation processes of prevalent homogeneous pattern as follows: Firstly, the distance of any two instances (for example B_4 and D_2 in Figure 2) are calculated, and the distance, Δ should be compared with distance threshold D_θ. If Δ is less than or equal to D_θ, there exists ***R***-proximity relationship between them. Thus $\{B_4, D_2\}$ is one of the 2nd-order candidate row instances of $\{B, D\}$. When all 2nd order candidate row instances are generated, according to the participation index (PI) threshold $\exists$, if $PI(B, D)$ is greater than or equal to $\exists$, pattern $\{B, D\}$is prevalent homogeneous pattern. Presumptively, (k-1)th order homogeneous pattern $\mathbb{C}_1$ connects (k-2)th order homogeneous pattern $\mathbb{C}_2$, then kth order co-location patterns can be generated. For example in Figure 3, $\{B_4, D_2\}$ connects $\{D_2, E_2\}$, and then $\{B_4, D_2, E_2\}$ can be obtained. When table instance of $\{B, D, E\}$ is generated, if $PI\{B, D, E\}$ is greater than or equal to the threshold, then $\{B, D, E\}$ is a prevalent homogeneous pattern. Repeating the same operation, kth-order homogeneous patterns can be generated.

B. GENERALIZED BUFFERING ALGORITHM

From a viewpoint of spatial entities, the generation of buffer zone is to build a polygon around the point, line and polygon instances. On the other hand, from a viewpoint of mathematics, the generation of buffer zone is to determine the

neighborhood of one certain spatial object, and the range of neighborhood upon the radius of neighborhood (called *buffer radius*, or *buffer distance* in traditional buffering algorithm). At this point, the traditional buffering algorithm can be described as follows:

Let $A = \{A_1, A_2, \ldots, A_d\}$ be a set of objects, and then the buffering of one spatial object is expressed by

$$Buffer_j = \{p : d(p, A_j) \le r\} \tag{3}$$

where r, which is a constant, is the buffer radius of buffer zone, and $d(p, A_j)$ is the Euclidean distance between p and A_j. $Buffer_j$ is the instances set that the distance is less than or equal to r.

For the set of spatial objects $A = \{A_1, A_2, \ldots, A_d\}$, the buffer zone can be defined by

$$Buffer = \bigcup_{j=1}^{d} Buffer_j \tag{4}$$

As mentioned in Section I, the existing buffering algorithms have shortcomings, such as the equidistance of buffer distance, and ignoring the homogenous and/or heterogeneous of neighbor instance's attributes. Consequently, the resulted buffer zone exposures low accuracy and useless zone (information), and loses of the useful zone (information).

To overcome these shortcomings of the existing buffer algorithm, an innovate algorithm called, *generalized buffer algorithm (GBA)* is proposed. The basic idea of the GBA is to utilize homogeneous rules to induce the generation of buffer zone, with which buffer radius is variable upon the attributes of instances. Consequently, the size of buffer zone is not a regular shape. The GBA mathematical model can be expressed by

$$CL\_B_j = \{p : d(p, A_j) \le r_{CL}\} \tag{5}$$

where r_{CL} is the generalized buffer radius, which is constrained by homogeneous rules; CL_B_j is the instance set, in which the distances of all instances are less than or equal to r_{CL}; $d(p, A_j)$ is the distance between p and A_j. The generalized buffer radius r_{CL} is the function of the attributes of the instances, and can be calculated by the homogeneous rules (*CLR*), i.e.,

$$r_{CL} = d\,[(x, y),\ CLR] \tag{6}$$

where $d[(x, y), CLR]$ is a distance function of the coordinates (x, y) of boundary point which is obtained using homogeneous rules.

The flowchart of the GBA is depicted in Figure 4. Let each instance $A_i = (x_1, x_2, \ldots, x_N)^T$ in data set $A = (A_1, A_2, \ldots, A_d)^T$ be a vector, representing instance-ID, spatial object type, location $\prod$, and attributes, etc., which d is the number of object types, T is transpose, and location $\in$ spatial framework. The spatial object types contain point, line, and polygon entities. The basic steps for the generalized buffer generation are described as follow.

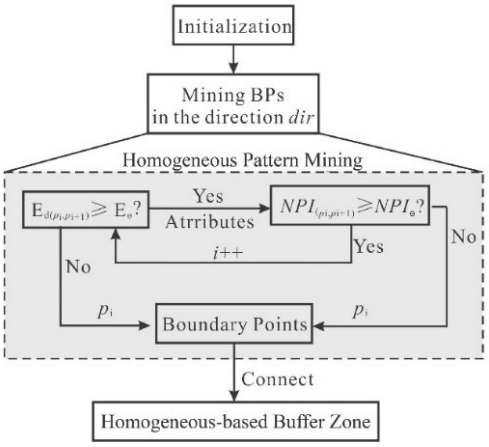

FIGURE 4. The flowchart of generalized buffering algorithm (GBA).

Step 1: *Initialization*: To build the generalized buffer zone of point, line, and polygon, the polar coordinate system is established, where axis is horizontal and angle φ_i is polar angle, varying from 0 to 2π. For point object, the pole is the itself. While, for line or polygon, the pole is the sampling points at the line or the arc of polygon.

Step 2: *Determining buffer boundary points.* This step consists of the following steps.

(a) In the direction of φ, firstly, the Euclidean distance between instances p_i and p_{i+1} is calculated by

$$E_{d(p_i, p_{i+1})} = \sqrt{(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2} \tag{7}$$

where $E_d(p_i,\ p_{i+1})$ is the Euclidean distance between instances p_i and p_{i+1}; x and y are the coordinates of instances. The $E_d(p_i,\ p_{i+1})$ should be judged whether it is less than or equal to the threshold of distance E_θ by Eq. 8. If Yes, turn to (b). Otherwise, instance p_i is regarded as the buffer boundary point in direction of φ, and then turn to (c).

$$R\,(p_i, p_{i+1}) = \begin{cases} 1 & if\ E_{d(p_i, p_{i+1})} \le E_\theta \\ 0 & if\ E_{d(p_i, p_{i+1})} > E_\theta \end{cases} \tag{8}$$

where $R(p_i,\ p_{i+1})$ represents the R-proximity relationship between instances p_i and p_{i+1}. $R(p_i,\ p_{i+1}) = 1$ means that instances p_i and p_{i+1} are the candidate of homogeneous instances.

(b) When instances p_i and p_{i+1} are determined as the candidate homogenous instances, their attributes should be used to determine whether they are homogenous instances. A new participation index (*NPI*) is defined by

$$NPI\ (p_i, p_{i+1}) = \frac{similar\_nA(p_i, p_{i+1})}{total\_nA} \tag{9}$$

where $NPI(p_i,\ p_{i+1})$ is the participation index of instances p_i and p_{i+1}; $similar\_nA(p_i,\ p_{i+1})$ is the number of attributes in which instances p_i and p_{i+1} meet the same thresholds of attributes, $total\_nA$ represents the number of all attributes which attend the computation of *NPI*. For example, let $AT = \{AT_1, \ldots, AT_5\}$ be a set of attributes. For the attributes AT_1, AT_2, AT_4, and AT_5, the values of attributes of instances

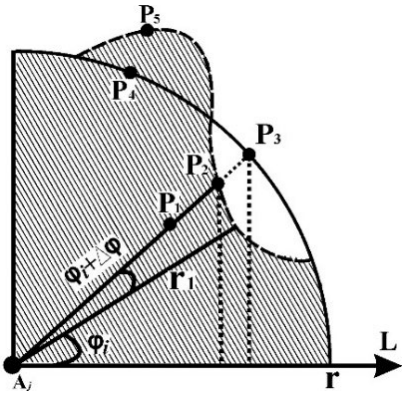

FIGURE 5. Generalized point buffer (GPIB) algorithm.

p_i and p_{i+1} meet the same thresholds of attributes, i.e., $similar\_nA(p_i, p_{i+1}) = 4$, and $total\_nA = 5$. Consequently, *NPI* is 4/5. *NPI* is judged whether it is greater than or equal to the threshold of participation index NPI_θ. If Yes, turn to (a). And then the value of i automatically increases 1. If No, turn to (c). And instance p_i is regarded as the buffer boundary point in direction of φ.

(c) Collect buffer boundary points, and put them into a matrix in order. The direction line φ increase $\Delta\varphi$ which is the infinitesimal of φ. And then turn to (a), and initialize the value of i. If $\varphi = 2\pi$ and no homogenous instance is found, turn to (d).

(d) Exit.

Step 3: *Connecting buffer boundary points to create the generalized buffer zone:* When all buffer boundary points are obtained, they are connected using interpolation method to form generalized buffer zone.

C. DISCUSSION OF THREE TYPES OF GBAS

The buffer generation usually consists of point, line, and polygon buffering algorithms. Thereby, three types of the generalized buffer generation algorithms for point, line, and polygon elements are described below, respectively.

1) GENERALIZED POINT BUFFERING ALGORITHM

The basic idea of the generalized point buffer generation is to determine the buffer radius of the neighbor instances using homogeneous rules, with which the zone with the same attributes are clustered into the buffering zone (i.e., homogenous attributes). This implies that the buffer radius is no longer a constant, i.e., a variable. The details are described as follows.

Step 1: Let a polar coordinate system, where axis is horizonal and angle φ_i is polar angle, varying from 0 to 2π. With the given definitions described in Section II.A and the basic principle of the generalized buffer generation described in Section II.B above. For a target point object A_j (see Figure 5), A_j is taken as the point to be buffering and meanwhile the pole of polar coordinates system, and r_φ represents the polar radius. The coordinates of boundary point ($BP(x, y)$) *of forming* the point buffer zone of target point object A_j can be defined by

$$P\_B_{A_j} = = \left\{p_i : E_d(p_i, A_j) \le r_\varphi\right\}, \quad \varphi \in [0, 2\pi] \tag{10}$$

$$r_\varphi = H\left\{BP(x, y),\ CLR_\varphi\right\}, \quad \varphi \in [0, 2\pi] \tag{11}$$

$$CLR_\varphi = G\left\{R(p_i, p_{i+1}), NPI(p_i, p_{i+1})\right\}, \quad \varphi \in [0, 2\pi] \tag{12}$$

where P_B_{Aj} represents the point instance set that the distance between P_i and A_j is less or equal to buffer radius r_φ in the corresponding direction; CLR represents homogeneous rules; $H\{BP(x, y), CLR_\varphi\}$ is the function about the coordinates of boundary point ($BP(x, y)$) and CLR. R(p_i, p_{i+1}) represents there exists R-proximity relationship between p_i and p_{i+1}. For a group of target point objects $A = \{A_1, A_2, \ldots, A_n\}$, the generalized point buffer is mathematically expressed by

$$P\_B_A = \bigcup_{i=1}^{n} P\_B_{A_i} \tag{13}$$

When the polar angle φ varies from 0 to 2π based on infinitesimal $\Delta\varphi$, in every corresponding orientation, homogeneous instances are mined until there are no new instances added into the homogeneous set. And the buffer radius r_φ, which is the distance between the boundary point (the farthest homogeneous instance away from target point object A_j) and target point object A_j, will be determined.

According to the theory of double integral, assuming that the boundary line is a continuous, differentiable, and integrable curve, the area of point buffer zone can be expressed by

$$S\_B_{A_j} = \int_0^{2\pi} d\varphi \int_0^{r_\varphi} r dr \tag{14}$$

where S_B_{Aj} is the area of point buffer zone, and $\Delta\varphi$ is the infinitesimal of polar angle φ, and r_φ is the buffer radius which changes with the change of boundary point's coordinates and subjects to homogeneous rules.

Step 2: **Mining buffer boundary points using homogeneous rules,** which consists of the steps below.

(a) **Determination of homogeneous candidates.** In order to determine homogeneous candidates, the R-proximity relationship of instances must be determined firstly. According to Eq. 7, if the Euclidean distance between instances P_i and P_{i+1}, which are at the same direction line $\varphi_i + \Delta\varphi$, is less than or equal to distance threshold E_θ, there exists R-proximity relationship between these two instances. Therefor those instances which satisfy R-proximity relationship constraint condition are the 2^{nd}-order homogeneous candidates.

(b) **Determination of prevalent homogeneous patterns.** When all 2^{nd}-order homogeneous candidates are found, Eq. 9 is utilized to determine prevalent homogeneous patterns. According to Eq. 9, if *NPI* of 2^{nd}-order homogeneous candidate $\mathbb{C}_i$ is greater than or equal to the threshold NPI_θ, $\mathbb{C}_i$ is a 2^{nd}-order prevalent homogeneous pattern. After above processing, only 2^{nd}-order prevalent homogeneous are determined. Based on the generation method of k^{th}-order

homogeneous candidate mentioned in the section II.A, k[th]-order prevalent homogeneous patterns will be obtained.

(c) **Determination of Buffer Boundary points.** As shown in the Figure 8, an example is made to explain the generation processes of the boundary point of point-based buffer when the polar angle is $\varphi_i + \Delta\varphi$. According to the above two steps, those instances (such as points P_1 and P_2) which satisfy homogeneous rules in this orientation are saved in a matrix. And the farthest homogeneous instance (for example point P_2) away from the target object instance A_j is regarded as the boundary point in this orientation, and is put into the matrix of boundary point. Thus, the buffer distance $r_{\varphi i}$ in arbitrary polar angle φ_i can be mathematically expressed by

$$r_{\varphi_i} = \sqrt{X^2_{b\_point} + Y^2_{b\_point}} \tag{15}$$

When φ_i varies from 0 to 2π, all boundary points will be collected and put into a matrix in order.

Step 3: **Generation of generalized point buffer zone.** When all buffer boundary points are obtained, they can be connected using interpolation method to form a generalized point buffer zone.

Compared to the implementation processes of traditional buffer analysis algorithm, in the proposed algorithm, at beginning, φ is increased in anticlockwise order based on infinitesimal $\Delta\varphi$. Meanwhile instances are judged whether they are homogeneous in corresponding direction. And $n+1$ boundary points can be got, among which the first point and the last point are the same point which makes the zone close. Finally, cubic spline interpolation method is used to connect these boundary points, and the generalized point buffer zone is the area which is within the boundary. As shown in Figure8, we can see that instance point P_3 which is the boundary point of buffer zone in traditional buffer zone is replaced by point P_2 which is the new boundary point of generalized point buffer zone, because instance point P_3 isn't the homogeneous instance and instance point P_2 is the last homogeneous instance in this orientation.

2) GENERALIZED LINE BUFFERING

The basic idea of generalized line buffer (GLB) algorithm is: a target line object is expressed by a point set $L_i = \{A_0, A_1, \ldots, A_m\}$, and is considered as an axis. The generation of the generalized line buffer (GLB) is mathematically expressed by

$$L_i\_B = \left\{ p : \bigcup_{j=0}^{m} [d(p, A_j) \le r_{A\,j}] \right\} \tag{16}$$

$$r_{A\,j} = D\,[BP_{A\,j}(x_b, y_b),\ CLR_{A\,j}] \tag{17}$$

where A_j is the j-th sample point along the target line object; r_{Aj} is the buffer distance corresponding to A_j, whose size various upon the attributes of neighbor instances; Li_B is the instance set that the distance between p and A_j is less or equal to r_{Aj} in the normal direction of the j-th sampled point along the target line object; $D[BP_{Aj}(x_b, y_b), CLR_{Aj}]$ is the function of the coordinates of boundary points, which is upon the attributes of the neighbor instances decided by the homogeneous rule. *CLRAj* represents the induced homogeneous rules (*CLR*) in the corresponding normal direction of target line object, which regards the j-th sample point as the foot point. For multiple target line objects $L = \{L_0, L_1, \ldots, L_n\}$, the generalized line buffer can be defined as

$$L\_B = \bigcup_{i=1}^{n} L_i\_B \tag{18}$$

In order to calculate the area of the generalized line buffer zone, across two endpoints of line, vertical lines of line are made, such as $V_1(x)$ and $V_2(x)$, respectively. Let d_s and d_r represent the infinitesimal element of line object and buffer radius, respectively. And let S_{A0} and S_{Am} be the areas of zone which are formed by the buffer zone of endpoints and vertical lines, respectively. The area of generalized line buffer zone can be calculated by

$$LS = \int_0^{L_i} ds \int_0^{r_{Aj}} dr + S_{A0} + S_{Am} \tag{19}$$

where r_{Aj} can be got from Eq. 17 which is constrained by homogeneous rules; S_{A0} and S_{Am} can be acquire based on Eq. 14.

When all boundary points are found, they will be connected by using cubic spline interpolation method to form an enclosed zone (see Figure9). Compared to the traditional line buffer algorithm, the proposed method considers that the buffer distance is no longer a constant, is a variable which is decided by attributes of neighbor instances and decided by homogeneous rules. As a result, the produced line buffering zone is much close to the actual situation.

With the algorithm proposed above, the details of the generation of GLB for a target line object can be summarized as the following steps.

Step 1: **Initialization of line:** First, the target line object is virtually sampled and noted as a series of points. Let ***ss*** be a sampling distance, a series of sampling points are obtained along the target line, and is expressed by $L_i = \{A_0, A_1, \ldots, A_7\}$ (see Figure 9). As shown in Figure 9, A_0, A_4, and A_7 are the starting point, inflection point, and endpoint, respectively.

Step 2: **Determination of the rectilinearity of adjacent three points.** Assume that coordinates of the adjacent three points be $A_{i-1}(X_{i-1}, Y_{i-1})$, $A_i(X_i, Y_i)$, and $A_{i+1}(X_{i+1}, Y_{i+1})$, their vectors are expressed by $\overrightarrow{A_{i-1}A_i}$, $\overrightarrow{A_iA_{i+1}}$. The vector product of the two vector is used to determine their rectilinearity, i.e.,

$$\overrightarrow{A_{i\text{-}1}A_i} = (X_i - X_{i-1},\ Y_i - Y_{i-1}) = \vec{g} = (m_1, n_1) \tag{20}$$

$$\overrightarrow{A_iA_{i+1}} = (X_{i+1} - X_i,\ Y_{i+1} - Y_i) = \vec{f} = (m_2, n_2) \tag{21}$$

The normal vector is calculated by

$$\begin{aligned} \vec{N} &= \overrightarrow{A_{i-1}A_i} \times \overrightarrow{A_iA_{i+1}} = \vec{g} \times \vec{f} \\ &= (m_1 n_2 - m_2 n_1)\,\vec{j} = \beta\,\vec{j} \end{aligned} \tag{22}$$

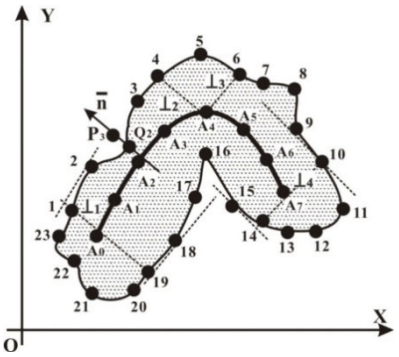

FIGURE 6. Generation of generalized line buffer zone.

where $m_1 = (X_i - X_{i-1})$, $n_1 = (Y_i - Y_{i-1})$, $m_2 = (X_{i+1}\text{-}X_i)$, $n_2 = (Y_{i+1}\text{-}Y_i)$; $\vec{j}$ is unit vector which is perpendicular to the plane constructed by vectors $\overrightarrow{\boldsymbol{A}_{i-1}\boldsymbol{A}_i}$ and $\overrightarrow{\boldsymbol{A}_i\boldsymbol{A}_{i+1}}$. If β is equal to zero, then $\overrightarrow{\boldsymbol{A}_{i-1}\boldsymbol{A}_i}$ and $\overrightarrow{\boldsymbol{A}_i\boldsymbol{A}_{i+1}}$ are collinearity, for example, points $A1$, $A2$, and $A3$ in Figure 9 almost lay a straight line, the β value is close zero; if β is greater than or less than zero, for instance, points A_3, A_4, and A_5, the middle point A_4 is usually an inflection point. Furthermore, the concavity and convexity of inflection point should be judged. When β determined by the products of the two vector is greater than zero, if the former vector turns to the latter vector with the minimum angle at counterclockwise direction, the inflection point is a convexity point; on the contrary, the inflection point is a concave point.

Step 3: **Generation of line buffering:**After linearity, inflection, and concavity-convexity of inflection point at the sampled points along a target line are completed, the following work is generation of GLB using homogeneous rule. First spatial and non-spatial attributes are selected to determine rough candidate the zone using homogeneous rule. The mathematical model is expressed by

$$NS\_A_{i,j} = \begin{cases} NS\_A_{i,j} & NS\_A_{i,j} \textit{ satisfies threshold} \\ NAN & NS\_A_{i,j} \textit{ does not satisfy threshold} \end{cases} \tag{23}$$

where $NS\_A_{i,j}$ is the value of a certain non-spatial attribute, i and j are coordinates. With implementing this step, those instances without meeting the given threshold will be excluded. Afterward, the generation of GLB are carried out, for which three types of sampled points are considered as follows, respectively.

(a) *When the points are neither at inflection points nor at endpoints*. Taking point A_2 in Figure 9 as an example to explain the line buffering generation. Assume that $\vec{\boldsymbol{n}}$ is a normal lines at line L_A, the instances points A_2, Q_2 and P_3 lay on the normal line $\vec{\boldsymbol{n}}$. The Euclidean distance between them is

$$D_{A_2,Q_2} = \sqrt{(X_2 - X_{Q_2})^2 + (Y_2 - Y_{Q_2})^2} \tag{24}$$

If $D_{\mathrm{A2},_{Q2}}$ is less than or equal to a given threshold D_θ, A_2 and Q_2 exists R-proximity relationship. Thereby, they are considered as candidate of homogeneous set. When all instances of candidate homogeneous pattern $\{A, Q\}$ are determined, according to Eq. 1 and Eq. 2, participation index (***PI***) is applied to decide whether the candidate homogeneous pattern is prevalent homogeneous pattern or not. If NOT, it is not considered as homogenous. When *k-th* order homogeneous patterns are mined, $k+1$-order homogeneous pattern may be found. This homogeneous mining process will not be stopped until no qualified homogeneous pattern is found. The point belongs to boundary point, i.e. point Q_2 in Figure 9.

(b) *When the points are at inflection.* Taking point A_4 in Figure 9 as an example to explain the line buffering generation. Making two perpendicular lines, noted as $\perp_2$ and $\perp_3$, along the adjacent two line segments (see line segments A_3A_4 and A_4A_5), and then making the parallel line of directed line segments (A_3A_4 and A_4A_5) crossing the buffer zone's boundary point (for example points 3) corresponding to the adjacent points (i.e., point A_3) of point A_4 are made. Then at the side of convex, an arc is constructed, which makes point A_4 as the center and regards the distance between point A_4 and the parallel line as the radius. The arc intersects with $\perp_2$ and $\perp_3$ to form a closed zone which is the initialization zone for mining homogeneous pattern. For inflection point A_4, R-proximity neighbor relations between instances will be determined by Eq. 14 in the closed zone, and homogeneous patterns are mined in this initialization zone. At last, buffer zone's boundary points, such as points 4, 5, 6, will be got. However, at the side of concave, R-proximity neighbor relations and homogeneous patterns are mined on the angular bisector of $\angle A_3A_4A_5$. And then the boundary points corresponding to inflection points are determined.

When the points are at endpoints. Taking point A_7 in Figure 9 as an example to explain the generation of GLB. Crossing the endpoint, vertical line of broken line is made, and parallel lines of broken line are produced across the buffer zone's boundary points (such as points 9 and 15) corresponding to point A_6 which adjacent to endpoint A_7. And then an arc is made, which makes the midpoint of the vertical line segment $\perp_4$ which intersects with the two parallel line as the center, and the distance between the midpoint and parallel line as the radius. The arc intersects with the two parallel lines respectively. The vertical line segment $\perp_4$ and the arc form a closed zone which is the initialization zone for mining homogeneous pattern. For endpoint, the process of mining homogeneous patterns is same to (a). At last, the boundary points of generalized line buffer zone (such as points 9, 10, 11, 12, 13, and 14) are collected and put into the set.

3) GENERALIZED POLYGON BUFFERING

Polygon-based buffer algorithm includes outwards or inwards buffer radius to generate a buffer zone. If the boundary of a polygon is considered as an enclosed curve, the polygon-based buffer algorithm is somewhat similar to line-based buffer algorithm (see Figure10). The mathematical models can be referenced to Eq. (16) - Eq. (18). The implementations can be carried out by:

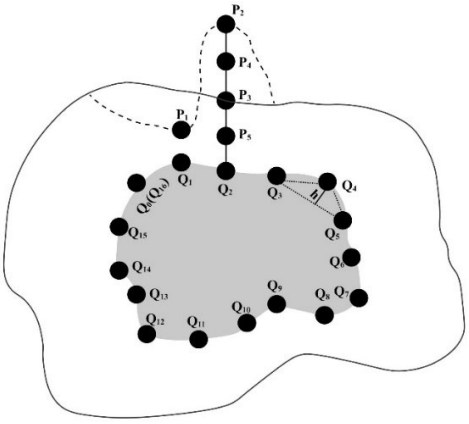

FIGURE 7. Generalized polygon buffer (GPLB).

Step 1: **Extraction of the polygon boundary, and initialization of the boundary line.**Because the buffer of polygon makes the boundary line as the axis, and a given distance as the buffer distance to dilate outside or shrink inside, so the initialization process is similar to broken line's. Points, which are on boundary line, should be collected by applying step size. Although the first point and the last point is overlap (such as points Q_0 and Q_{16} in Figure 10), it is not necessary to separately consider the buffer boundary of terminal points, as long as according to the generalized buffer construction method of middle point of consecutive three points.

Step 2: **Determination of the rectilinearity of adjacent three points.**The polygon boundary is a curve, but in this paper, directed line segments are connected to replace the curve (such as $\overrightarrow{a}$, and $\overrightarrow{b}$ in Figure10). When the amount of line segments is more enough, the boundary of closed zone, which is constructed by directed line segments, is smoother. The rectilinearity of adjacent three points (such as points Q_3, Q_4 and Q_5) can be determined by

$$h = \frac{(Dx_4 + Jy_4 + K)}{\sqrt{D^2 + J^2}} \tag{25}$$

where h is the distance between middle point Q_4 and straight line across the first point Q_3 and last point Q_5 of adjacent three points; D, J, and K are coefficients of equation of the straight line; $\boldsymbol{x_i}$ and $\boldsymbol{y_i}$ are coordinates of middle point (for example point Q_4). If h_i is less than or equal to the threshold, then the adjacent three points can be regarded as collineation.

Step 3: **Creation of generalized polygon buffer.**Similar to the operation for line in section II.C, the points on the boundary line of polygon need to be judged whether they are inflection points. If h_i is greater than the threshold, then the $\boldsymbol{i}$-th point is inflection point. When the types of points are determined, homogeneous rule should be utilized to guide the generalized polygon buffer zone. Firstly, **non-spatial attributes** are selected to determine rough candidate homogeneous instances using Eq. 17, and then in order to get candidate homogeneous patterns, R-proximity relations among these rough candidate homogeneous instances should be determined in normal line direction by Euclidean distance, and the points which are on the boundary of polygon but not the inflection points are made as foot of perpendicular. For instance, in Figure 13b, Q_j is a point on the boundary line of polygon, and there are four points which are satisfy Euclidean distance condition, i.e., there are R-proximity relationship among them in normal line direction which make Q_j as the foot of perpendicular. Then 2-order candidate homogeneous patterns can be got, such as $\{Q_j, P_i\}$ and$\{P_i, P_{i+1}\}$. For the sake of determining whether these candidate homogeneous patterns are prevalent homogeneous patterns, participation index ($NPI\{\mathbb{C}\}$) of patterns should be calculated by Eq. 1 and Eq. 2. If $NPI\{\mathbb{C}\}$ is greater than or equal to the threshold, then pattern $\mathbb{C}$ is prevalent homogeneous pattern. In addition, k^{th} order homogeneous patterns can be generated based on k-1^{th} order and k-2^{th} order homogeneous patterns. At last, the point (for example point P_2) which meets homogeneous rules and is farthest away from Q_j in normal line direction is regarded as the boundary point of generalized polygon buffer zone. The same operation is performed for other points which are on the boundary line of polygon. All boundary points of generalized polygon buffer zone are collected and put into a set in order.

III. EXPERIMENTS AND ANALYSIS

Traditional buffer analysis usually took the noise pollutions as paradigm. This paper also takes the noise pollutions paradigm in the two cities of China as example to explain how the genialized buffer is used for spatial analysis and what is difference between the generalized and traditional buffer algorithm.

A. DATA SETS

Data set-1: The investigation area is the electronic vector data of Bao'an District, Shenzhen, China, and obtained from the Geographical Information Monitoring Cloud Platform in 2015 with ArcGIS shp file. The Data set-1 contains more than 30 million pieces of information (e.g., hospitals, buildings, supermarkets, gas stations, banks, etc.), in which there are 24 types of point features consisting of 1876 point elements, (e.g., airports, parking, gas stations, etc.), 11 types of line feature consisting of 26135 line elements (e.g., in-state free highway, provincial highway, pedestrian, etc.), and 7 types of polygon features consisting of 1201 polygon elements (e.g. provincial boundaries, regional boundaries, lake, etc.).

Data set-2: The investigation area is the electronic data of Beijing in 2015 and obtained from the Geographical Information Monitoring Cloud Platform, with ArcGIS shp format. The data set-2 contains more than 30 million pieces of information (e.g., hospitals, buildings, supermarkets, gas stations, banks, etc.). The Data set-2 contains 18 types of point features consisting of 124252 point elements (e.g., village, school, bus station, etc.), 9 types of line features consisting of 112057 line elements (e.g., expressways, provincial highways, pedestrian, etc.), and 4 types of polygon features consisting of 6933 polygon elements (e.g., lakes, rivers, boundaries of regions, etc.).

In the two data sets, the point noise source is assumed from in-situ construction, such as excavating, blasting, cement

TABLE 2. The information of spatial data.

Attributes	Dataset-1	Dataset-2
X, Y coordinate	Geographical coordinate (system: Lat Long for MAPINFO type) Datum: D_MAPINFO Prime meridian: Greenwich Angular Unit: Degree Spheroid: World_Geodetic_System_of_1984 Semimajor Axis: 6378137.00 Semiminor Axis: 6356752.314 Inverse Flattening: 298.257	Geographical coordinate (system: GCS_WGS_1984) Datum: D_WGS_1984 Angular Unit: Degree (0.0174532925199433299) Prime Meridian: Greenwich Spheroid: WGS_1984 Semimajor Axis: 6378137.00 Semiminor Axis: 6356752.314 Inverse Flattening: 298.257

TABLE 3. Instance non-spatial attributes.

Type of data		Non-spatial attribute
Point, Line, Polygon	Quantitative attributes	Time, Length, Width, Height, Area, Quantity, Size, Level, Density, Integrated areal density (IAD)
	Qualitative attributes	Name, Category, Number, Utilization type, Characteristics

mixing, etc. The line noise source is assumed from moving vehicles on the highway. The polygon noise source is assumed from the manufacture, in which the machines are operated continuously. Many buildings are surrounding these noise sources, i.e., surrounding the point element, line element, and the polygon element.

- ***Spatial data:*** The spatial data for describing point, line and polygon features are X, Y coordinates. The datum and the projection coordinate system for spatial data for Dataset-1 and Dataset-2 are listed in TABLE 2.
- ***Attribute data:*** The attribute data of point, line, and area features include quantitative and qualitative attributes (see TABLE 3). In order to investigate the extent of noise propagation using point, line and polygon buffer algorithms. Without losing generality, only one attribute, namely, Integrated areal density (IAD) is selected, and other attributes are not taken into account in this step. This assumption is consistent with Kang (2004), who though that the IAD of a building hinders noise propagation.

B. GENERALIZED BUFFER ANALYSIS

The experiments were conducted under a computer with an Intel Xeon E5645 Six core 2400 MHz (12MB Cache) and 4GB of RAM.

1) EXPERIMENT FOR DATASET 1

a: GENERALIZED POINT BUFFER ANALYSIS

The purpose of noise pollution using point buffer analysis is to investigate how big area is impacted by noise source. Usually, the volume of a sound higher more than 80 dB is considered as noise, and of lower than 45 dB - 60 dB is NOT considered as urban environmental noise. The noise attenuation is expressed

$$\Delta dB = 10\log(\frac{1}{4}\pi r^2) \tag{26}$$

where r is the noise propagation distance (m) and ΔdB is the noise attenuation (dB), which decreases with increasing distance square at a logarithmic function.

This paper selects 45 dB as a threshold of either sound or noise, *i.e.,* bigger than 45 dB as noise. With the threshold, the buffer radius in traditional point buffering algorithm is

$$r = \sqrt{4 \times 10^{(80-45)/10} \Big/ 3.1415926} = 63.5(\text{m}) \tag{27}$$

In order to compare the traditional point buffering algorithm (TPIA), taking point, A as noise source, which is a mechanical manufacturing factory, to create point buffer analysis and depicted in Figure 14. As observed from Figure 8, the traditional point buffer area is a circle centered at Point A, i.e., the noise source. This means that traditional point buffer analysis thinks that the noise propagation is along a flat terrestrial area in all directions without any obstacle. In fact, the noise propagation is impacted by many factors, such as buildings, as a result, the noise propagation is NOT a complete circle, *i.e.*, the noise propagation distances are NOT equal in all directions.

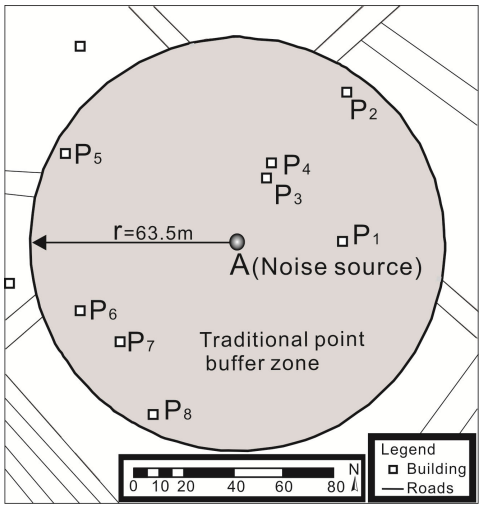

FIGURE 8. Traditional point buffer zone with a buffer radius of 63.5 m.

In the generalized point buffer (GPIB) algorithm, also taking Point A as the noise source, 45 dB as a threshold of either sound or noise, buildings noted as Point P_i, ($P_i = P_1, \ldots, P_8$) in Figure 8 as obstacles, whose attributes are listed in TABLE 3, IAD as factor of noise attenuations. The steps of the GPIB algorithm are as follows.

Step 1: *Establish a polar coordinate system.* The Point A is taken as the pole, and Easting as polar axis, noted, X-axis. φ is defined "*polar angle*", whose value is determined relative to the X-axis counterclockwise, $\Delta\varphi$ is defined as the angle increment at a counterclockwise rotation. In this experiment,

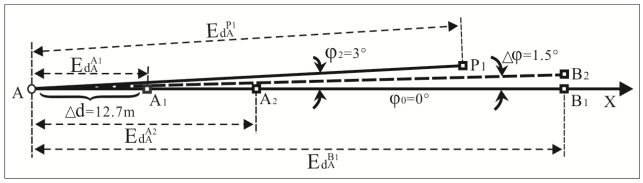

FIGURE 9. Generalized point buffer analysis algorithm.

the polar angle is expressed using φ_i, *i.e.*, $\varphi_{i+1} = \varphi_i + \Delta\varphi$, Δd is defined as increment distance (distance step) along rotational axis starting from pole Point A (see Figure 9).

Step 2: *Calculating the sound propagation course.* Starting Point A, along the X axis, *i.e.*, $\varphi_0 = 0°$. Set a distance increment of 12.7 m (The 63.5/5 = 12.7m, noted as, $\Delta d = 12.7$m, other lengths is discussed below). The starting point is located in Point A_1. The Euclidean distance between Point A and Point A_1 can be calculated using Eq. (7), noted as, $Ed(A, A_1) = 12.7$ and noise attenuation is also calculated using Eq. (26), which is 20.8dB. Meanwhile, it is determined whether the sound propagation encounters any building. It has been indicated from the data set that no buildings have blocked the nose propagation (see Figure 9). Because $Ed(A, A_1)$ is less than the distance threshold, noted $E_\theta = 63.5$ m, Point A_1 is considered to be homogenous with Point A, i.e., Point A_1 with Point A has a *R-proximity relationship*. In order to further determine their homogeneous, the *NPI* is calculated using Eq. (9), where the *total_nA* in Eq. (9) is the number of all attributes, which is 3 (IAD and XY coordinates) in TABLE 2 and TABLE 3, *i.e.*, $total\_nA = 3$. The *similar_nA* is the number of attributes for each point (see TABLE 3). Point A has 2 attributes (XY coordinates), $similar\_nA = 2$. *i.e.*, $NPI_A = 2/3$. Therefore, the threshold of *NPI* is calculated using Eq. (9), *i.e.*, $NPI_\theta = 2/3$. When the *NPI* value of the candidate instance is equal to or greater than NPI_θ, this means that the candidate instance is homogeneous with Point A. On the other hand, the *NPI* value of Point A_1 is $NPI_{A1} = 2/3$, which is equal to NPI_θ. This means that Point A_1 and Point A are homogeneous. Moreover, the noise value at Point A_1 is 59.2dB (≥45dB), so the noise at Point A_1 will continues to propagate along X-axis.

Step 3: Repeat the same operation at the same distance increment, i.e., ($\Delta d = 12.7$ m), the sound is propagated to Point A_2, and calculate the Euclidean distance between Point A and Point A_2, and noise values, and also determine whether encountering any building. With calculations, the volume of sound is 27.0 dB, and no building is encountered. The Euclidean distance of Point A_2 and Point A is $Ed(A, A_2) = 12.7$ m +12.7 m = 25.4 m, which is less than $E_\theta = 63.5$ m and $NPI_{A2} = 2/3$, which is equal toNPI_θ. This means that Point A and Point A_2 are homogeneous. In addition, the noise value at Point A_2 is 53 dB (≥45 dB). This means that the sound will continuously propagate along the X-axis. Repeat the ***Step 2***, until the volume of sound is equal to and less than 45 dB, and/or encounter the buildings which can block the sound propagation. Therefore, a boundary point, far away 63.5 m from Point A is found, and noted, Point B_1. and $Ed(A, B_1) = 63.5$ m, which is the boundary of the GPIA at $\varphi_0 = 0°$ (see Figure 9).

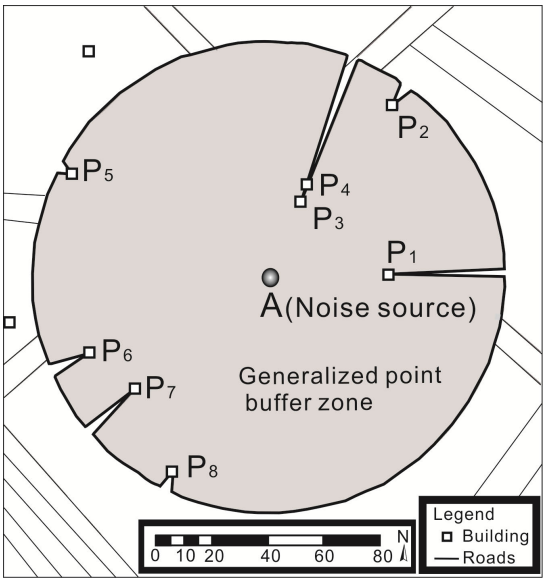

FIGURE 10. Generalized point buffer using Dataset 1 with the increment angle at $\Delta\varphi = 1.5°$.

Step 4: Anticlockwise rotating the polar axis with an increment angle of 1.5°, *i.e.*, $\Delta\varphi = 1.5°$ and $\varphi_1 = \varphi_0 + \Delta\varphi = (0° + 1.5°) = 1.5°$ (see Figure 9). Repeating ***Step 2*** and ***Step 3*** for the same operations along new X-axis at polar angle of 1.5°. With the repeated operations and the same calculations, during which no buildings/blocks are, until the boundary Point B_2 is recognized.

Step 5: Repeating ***Step 4*** with anticlockwise rotating the polar axis at an increment angle of 1.5°, i.e., $\Delta\varphi = 1.5°$, $\varphi_2 = \varphi_1 + \Delta\varphi = (1.5° + 1.5°) = 3°$ (see Figure 9). Repeating ***Step 2*** and ***Step 3*** along X-axis at polar angle of 3°, it is found that the sound propagation meets a blocking building, noted, Point P_1. Calculating the Euclidean distance from Point A to Point P_1 using Eq. (7), *i.e.*, $Ed(A, P_1) = 44$ m, which is less than $E_\theta = 63.5$ m. This implies that Point P_1 is homogenous with Point A, since it satisfies *R-proximity relationship* constraint condition in Eq. (8). In addition, for Point P_1 NPI_A = (similar_nA)/(total_nA) = 1/3, which is smaller than the threshold, $NPI_\theta = 2/3$. Therefore, it can be concluded that Point P_1 and Point A are not homogeneous. This implies that the sound propagation ends at Point P_1. Thereby, Point P_1 is considered as the boundary point of the GPIB algorithm.

Step 6: Repeating ***Step 2*** through ***Step 5*** with anticlockwise rotating the polar axis with an increment angle of 1.5°, until $\varphi = 360°$, the GPIB algorithm is finished. The result is shown in Figure 10.

***Comparison analysis with the different angles increments*:** In order to compare the accuracy of the GPB algorithm, the different increments of angle are set up at 3°, 6°, and 9°. The same operations are carried out as ones above,

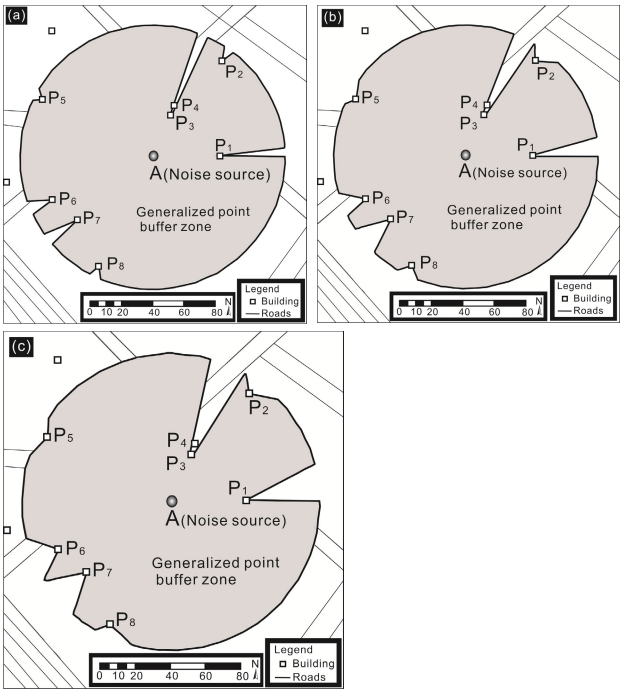

FIGURE 11. Generalized point buffer using Dataset 1 with the different increment angles at (17a) $\Delta\varphi_2 = 3°$, (17b) $\Delta\varphi_3 = 6°$, and (17c) $\Delta\varphi_4 = 9°$.

and the results are shown in Figure 11(a), Figure 11(b), and Figure 11(c), respectively.

b: GENERALIZED LINE BUFFER ALGORITHM

The validation takes a vehicle's noise along the highway as noise source to investigate how big area is impacted by vehicle nose source using the generalized and traditional line buffer algorithm. In order to detailedly explain the GLB algorithm, two endpoint and one inflection point are selected as examples. Also, assuming that the vehicle's noise is 80 dB (The vehicle's noise is usually between 60-80 dB [25]), the traditional line buffer zone is generated using a buffer distance of 63.5 m along the highway (line element). The result is depicted in Figure 12

Similar to the GPIB algorithm, the GLB algorithm also takes the buildings as obstacle, which block the noise propagation upon their properties (e.g., IAD), and are represented by Point P_i ($i = 1, \ldots, 24$) in Figure 12. The steps for the GLB algorithm are described as follows.

Step 1: The line is first re-sampled into the line segments at a given distance increment, noted $\Delta L = 1m$, without losing generality. The points are called *sample points*. The 6 sample points, noted A_i ($i = 1, \ldots, 6$), located in the starting, the inflection of line (highway), respectively, are taken as example to explain the process of the GLB algorithm (see Figure 13(a)). The spatial attributes and non-spatial attributes are listed in TABLE 2 and TABLE 3.

Step 2: Starting from Point A_1 (see Figure 13(b)), determining whether Point A_1 is either endpoint, or inflection point, or none of them using the method in Section II.C. Because Point A_1 is an endpoint, a line perpendicular to

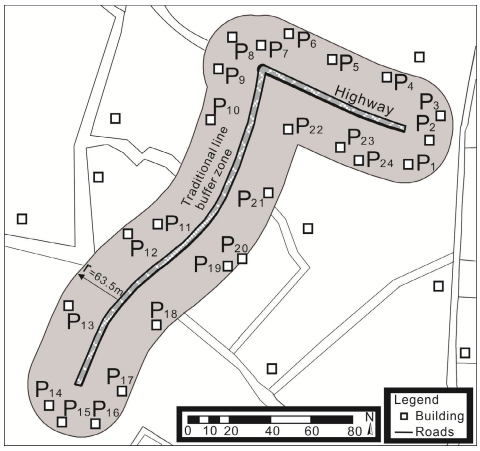

FIGURE 12. Traditional line buffer analysis with a buffer distance of 63.5 m.

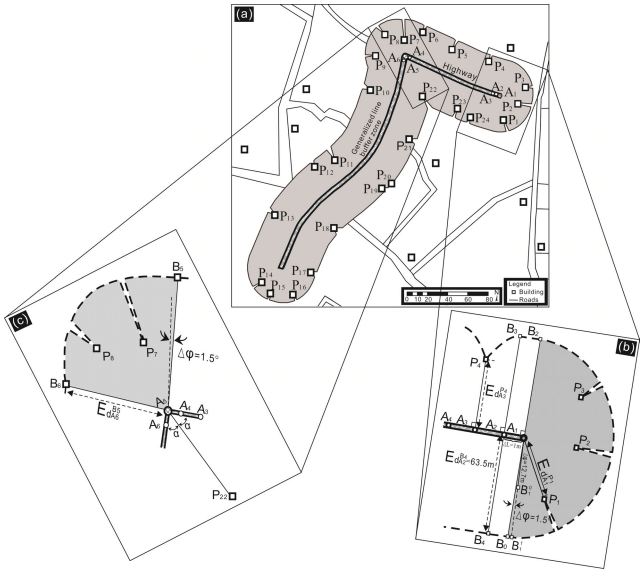

FIGURE 13. (a) Generalized line buffer using Dataset 1 with the increment length at $\Delta L = 1.0$ m, (b) and (c) Generalized line buffer analysis algorithm implementation step diagram.

the line passing Point A_1 is made, *i.e.,* $B_1A_1B_2$, where the Point B_1 and Point B_2 are the boundary points of the noise propagation in the GLB algorithm. Similarly, the same calculation and process are made for Point A_1 using the method described in ***Step 3*** of the GPIB algorithm in Section III.B. With the calculated results, it is found that the Point B_0 and Point B_2, are traditional boundary points without encountering any buildings (blocks). Thereby, their Euclidean distances, $\mathrm{Ed}(A_1, B_0)$ and $\mathrm{Ed}(A_1, B_2)$, are 63.5 m, respectively.

Step 3: With an increment angle of $\Delta\varphi = 1.5°$ and rotating anticlockwise the axis AB_1^0 until $\varphi = 1.5°$. Set a increment distance of 12.7 m along the axis $A_1B_1^0$ at B_1^0, repeat the ***Step 2*** thru the ***Step 5*** in the GPB algorithm described in Section III.B to determine whether any building (block) is encountered during the noise propagation. When the noise propagates to point b, it is found that no buildings (blocks) are encountered with the Point b. Will the repeated operation,

no buildings (blocks) are encountered until reaching Point B_1^1., thus Point B_1^1 is considered as boundary point.

Step 4: Repeat the ***Step 3*** above with an increment angle of $\Delta\varphi = 1.5°$ until the rotation angle reach $\varphi = 31.5°$, it is found that the sound propagation encountered a building, noted, Point P_1. Calculating the Euclidean distance from Point A_1. to Point P_1. using Eq. (7), *i.e.*, $Ed(A_1, P_1)$. It satisfies *R-proximity relationship* constraint in Eq. (8) (i.e., less than 63.5 m), and are considered to be homogenous with, and candidate of homogeneous with Point A_1. Further calculating $NPI_{P1} = (similar\_nA)/(total\_nA) = 1/3$, which is smaller than the threshold $NPI_\theta = 2/3$ to determine whether Point P_1 is homogeneous with Point A_1 or not using Eq. (9) in Section III.B.. With the calculated result, it is found that Point P_1 and Point A_1 are NOT homogenous. Thus, Point P_1 is considered as the boundary point of the GLB zone.

Step 5: Continue increasing the increment angle of $\Delta\varphi = 1.5°$, repeat the ***Step 4***, it is found that Point P_2 and Point P_3 are NOT homogeneous with Point A_1, which means that they are boundary points.

Step 6: Continue increasing the increment angle of $\Delta\varphi = 1.5°$, repeat the steps above until Point B_2.

Step 7: With the assumed length increment, $\Delta L = 1.0$ m, Point A_2 is operated to determine whether it is either an endpoint, or an inflection point, or none of them by the methods described in Section II.C. With the computation, Point A_2 is none of them. Two line-segments (A_2B_3, A_2B_4) are made, respectively on the both sides of the highway (line element), meanwhile the homogeneous along the both of line segments, A_2B_3, A_2B_4 are determined (see Figure 19(b)). It can be concluded that no buildings are encountered in the both of line segments, i.e., Point B_3 and Point B_4 as the boundary points.

Step 8: Repeat the same operation with the same length increment, until Point A_3, it is found that the sound propagation encounters a building, noted, Point P_4. Calculating the Euclidean distance between Point A_3 and Point P_4 using Eq. (7), *i.e.*, $Ed(A_3, P_4) = 37$ m, which satisfies *R-proximity relationship* constraint condition in Eq. (8) (i.e., less than 63.5 m), and are considered as a candidate of homogeneous with Point A_3. Further determining whether Point P_4 is homogeneous with Point A_3 using ***Step 4*** above, Point P_4 is considered as the boundary point of the generalized buffer analysis. Similarly, the Point B_5 is boundary point.

Step 9: Repeat ***Step 7*** and ***Step 8*** above, until Point A_5 (see Figure 13(c)), With the computation, Point A_5 is a inflection point by the methods described in Section II.C. Two vertical lines along two line segments (A_4A_5, A_5A_6) are made. The boundary points are determined using the same ***Step 3***. The Point B_6 and Point B_7 are boundary points. their Euclidean distances, $Ed(A_5, B_6)$ and $Ed(A_5, B_7)$, are 63.5 m, respectively.

Step 10: Assuming that Point B_6 is a starting point, repeat the ***Step 4*** with an increment angle of $\Delta\varphi = 1.5°$ to determine the boundary points. Point P_7 and Point P_8 are selected.

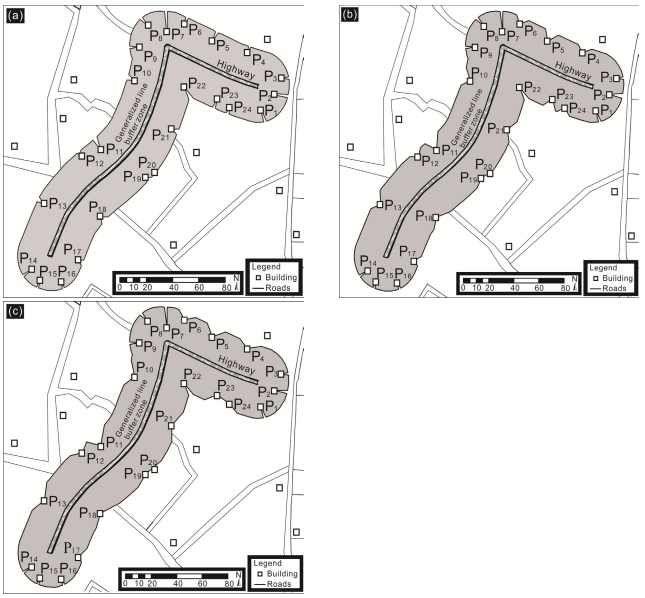

FIGURE 14. Generalized line buffer using Dataset 1 with the different increment length at (a) $\Delta L_2 = 5$ m, (b) $\Delta L_3 = 5$ m, and (c) $\Delta L_4 = 15$ m.

Step 11: Continue increasing the increment angle of $\Delta\varphi = 1.5°$, repeat the steps above until Point B_7.

Step 12: On the angular bisector of $\angle A_4A_5A_6$, repeat the same ***Step 4***, Point P_{22} as the boundary point.

Step 13: Connect all boundary points to form a generalized line buffer analysis area. The results are shown in the Figure 13(a).

Generalized Line Buffer Generation using Different Length Increments ΔL: In order to compare the difference when using different distance increments, ΔL,the difference value of ΔL at 5m, 10m, and 15m are set. Repeat the same operations above, the results of the GLB zones with the different length increment are depicted in Figure 14(a), Figure 14(b), and Figure 14(c).

c: GENERALIZED POLYGON BUFFER (GPLB) GENERATION

Assume that the noise source is produced by a factory at 80dB (Xia *et al*. 2001). The region of the factory with noise is depicted in Figure 15. Without losing the generality, the noise at any point within the factory is the same. The traditional polygon buffer zone is generated using a buffer distance of 63.5 m, and is shown in Figure 15.

The operations of GPLB algorithm is almost the same as the one of the GLB, the unique difference is the starting point is the same the endpoint. For this reason, any point at curve is selected as starting point. For example, Point Q_1 is selected as starting point, and the boundary of the polygon is re-sampled into cure segments starting from Point Q1 at a given curve of 1.0 m, noted as, $\Delta S = 1.0$ m. The points are called *sampled points*. The 4 sampled points, noted as Point Qi ($i = 1, \ldots, 4$), which are located on the boundary of polygon (factory with noise) are taken as examples to explain the algorithm of the GPLB zone generation (see Figure 16).

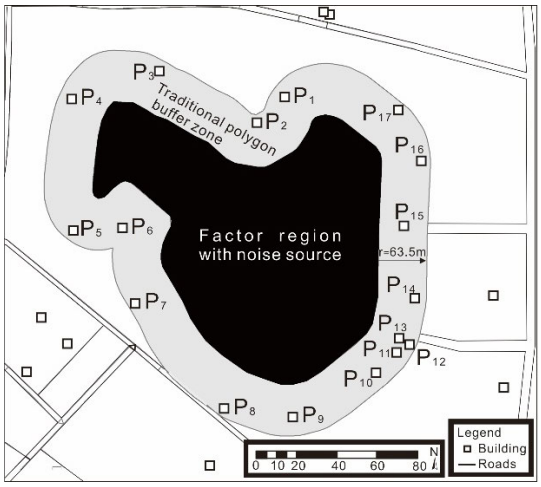

FIGURE 15. Traditional polygon buffer analysis with a distance of 63.5 m.

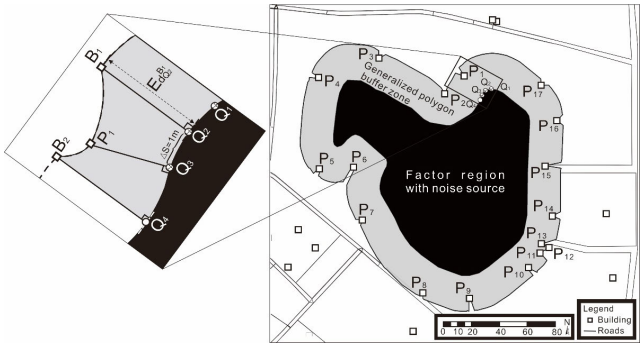

FIGURE 16. Generalized polygon buffer using Dataset 1 with the increment curve at $\Delta S = 1.0$ m.

The spatial attributes and non-spatial attribute for each of the sampled points are the same as ones above.

for Point Q_2 (see Figure 16), whether is it an inflection point or not is determined by the method described in Section II.C, with which Point Q_2 in an inflection point. The results are shown in the Figure 16.

GPB Zone Generation using different curve increments ΔS : In order to investigate the accuracy of the GPB algorithm with the different increment of curves, ΔS, the increment of cures at 5 m, 10 m, and 15 m are set up for the experiments. The same operations are repeated the ones above. The results are shown in Figure 17(a), Figure 17(b), and Figure 17(c).

2) EXPERIMENT FOR DATA SET 2

a: GENERALIZED POINT BUFFER ZONE GENERATION

For Dataset 2, the same assumption of noise resource with 80 dB happens at Point A. The same operations are made as one for Dataset 1, as described in Section III.B. The traditional and generalized point buffer zone are generated. The results are depicted in Figure 18 and Figure 19, respectively.

b: GENERALIZED LINE BUFFER ANALYSIS

Similarly, a vehicle's noise along the highway is considered as noise source. The same operations are made as one for Dataset 1, as described in Section III.B. The experimental results are shown in Figure 20 and Figure 21, respectively.

FIGURE 17. Generalized polygon buffer using Dataset 1 with the different increment curves at (17a) $\Delta S = 5.0$ m, (17b) $\Delta S = 10.0$ m, and (17c) $\Delta S = 15.0$ m.

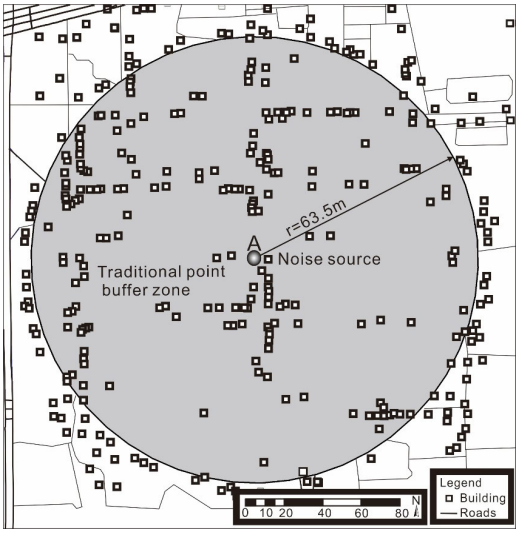

FIGURE 18. Traditional point buffer analysis with a radius of 63.5 m.

c: GENERALIZED POLYGON BUFFER ANALYSIS

Similarly, a noise at a factory is considered as noise source. The same operations are made as one for Dataset 1, as described in Section III.B. The experimental results are shown in Figure 22 and Figure 23.

C. COMPARISON ANALYSIS AND REMARKS

1) COMPARISON ANALYSIS

In order to compare the difference between the traditional and generalized buffer algorithms, the parameters below are used for indexes.

Area difference, which is the difference of the areas between traditional buffer zone and generalized buffer zone,

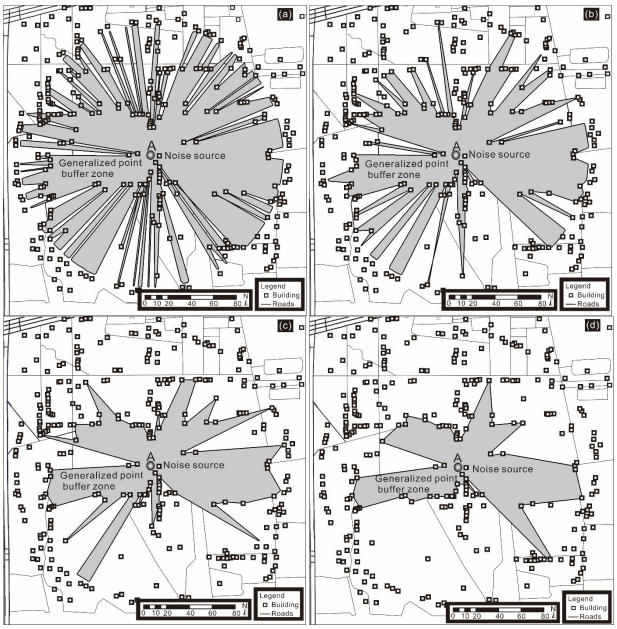

FIGURE 19. Generalized point buffer using Dataset 2 with the different increment angles at (a)$\Delta\varphi_1 = 1.5°$, (b) $\Delta\varphi_2 = 3°$, (19c) $\Delta\varphi_3 = 6°$, and (d) $\Delta\varphi_4 = 9°$.

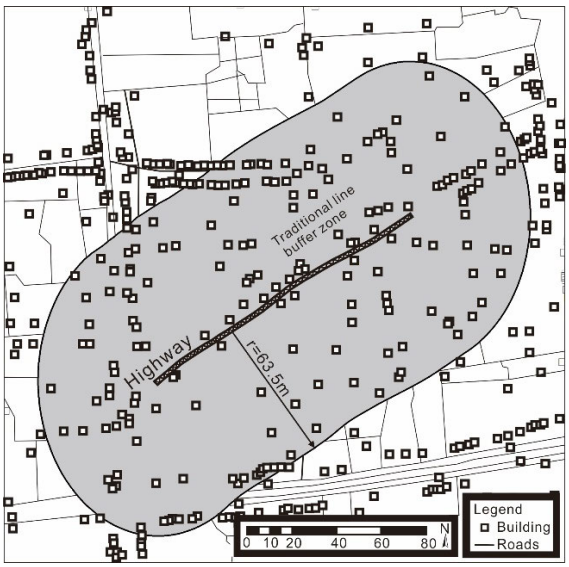

FIGURE 20. Traditional line buffer analysis with a distance of 63.5 m.

i.e.,

$$\Delta_{Area} = Area_T - Area_{G_i} \tag{28}$$

where Δ_{Area} is the difference of areas(m^2); $Area_T$ is the area from traditional buffer zone; $Area_{Gi}$ is area from generalized point/line/polygon buffer zone; $G_i = Point, line$ and $Polygon$.

Perimeter difference, which is the difference of the perimeter between traditional buffer zone and generalized buffer zone, i.e.,

$$\Delta_{Perimeter} = Perimeter_T - Perimeter_{G_i} \tag{29}$$

where $\Delta_{Perimeter}$ is the difference of perimeters (m); $Perimeter_T$ is the perimeter from traditional buffer zone; $Perimeter_{Gi}$ is perimeter from generalized point/line/polygon buffer zone.

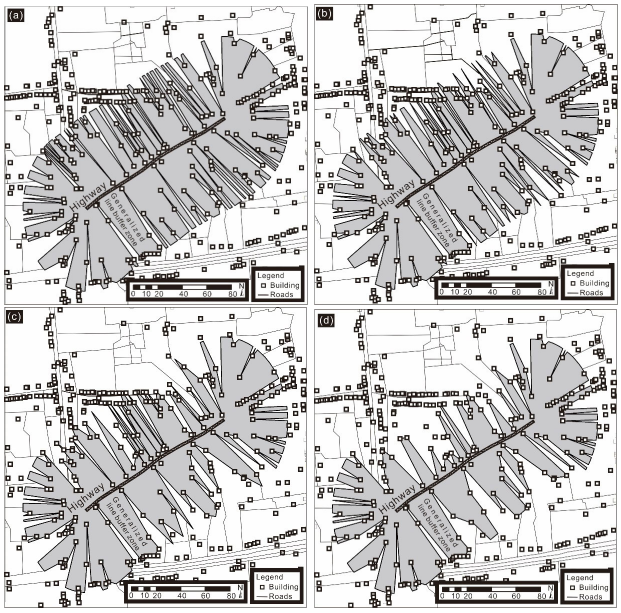

FIGURE 21. Generalized line buffer using Dataset 2 with the different increment length at (21a) $\Delta L_1 = 1.0$ m, (21b) $\Delta L_2 = 5.0$ m, (21c) $\Delta L_3 = 10.0$ m, and (21d) $\Delta L_4 = 15.0$ m.

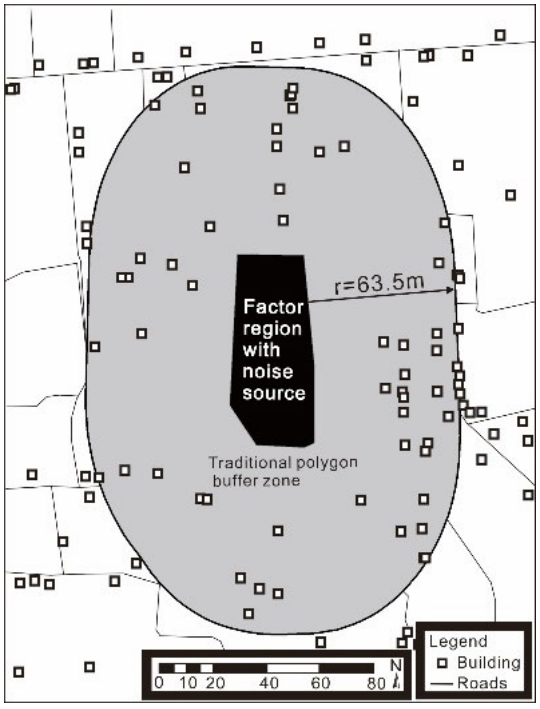

FIGURE 22. Traditional polygon buffer analysis with a distance of 63.5 m.

Relative position similarity, which is defined as the distance between the centroids of the two buffers (Zhang *et al.* 2018), i.e.,

$$D(|a\ ,\ b|) = \frac{1}{1 + |a\ ,\ b|^2} \tag{30}$$

TABLE 4. The difference value and the shape similarity index of the *GPIB* relative to *TPB* for Dataset 1.

	Δ_{Area} (m^2)	$\Delta_{perimeter}$ (m)	Δ_f(m)	D(|a, b|) (%)	S_{area} (%)	S_{per} (%)
$\Delta\varphi_1$=1.5°	29	-205	1.4	41.4	97.7	66.1
$\Delta\varphi_2$=3°	46	-191	1.5	39.8	96.4	67.6
$\Delta\varphi_3$=6°	83	-167	2.2	31.2	93.5	70.5
$\Delta\varphi_4$=9°	112	-146	2.8	26.3	91.2	73.3

TABLE 5. The difference value and the shape similarity index of the *GLB* relative to *TLB* for Dataset 1.

	Δ_{Area} (m^2)	$\Delta_{perimeter}$ (m)	Δ_f(m)	D(|a, b|) (%)	S_{area} (%)	S_{per} (%)
ΔL_1=1 m	3697	-986	0.9	53.7	98.5	72.9
ΔL_2=5 m	5897	-878	1.0	50.9	97.6	75.2
ΔL_3=10 m	8805	-790	1.6	38.9	96.4	77.1
ΔL_4=15 m	12286	-725	2.1	33.8	94.9	78.6

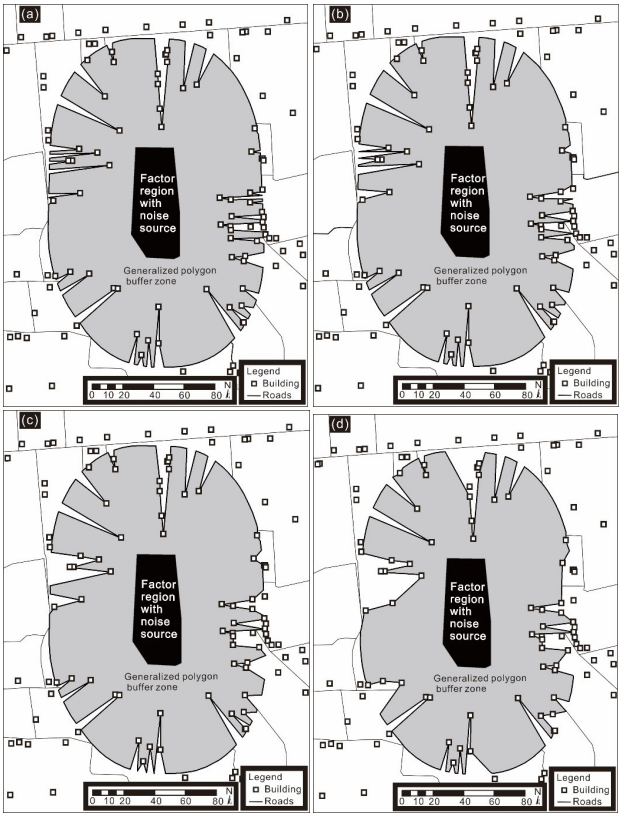

FIGURE 23. Generalized polygon buffer using Dataset 2 with the different increment curves at (a)$\Delta S_1 = 1.0$ m, (b) $\Delta S_2 = 5.0$ m, (c) $\Delta S_3 = 10.0$ m, and (d) $\Delta S_4 = 15.0$ m.

where D(|a, b|) is the relative position and $|a, b|$ is the distance between the two centroids.

Relative area similarity, which is defined as follow (Zhang *et al.* 2014):

$$S_{area} = 1 - \frac{|S_T - S_G|}{Max(S_T, S_G)} \tag{31}$$

where S_{area} is the relative area similarity, $|S_T - S_G|$ is the area difference of two buffer regions, and Max(S_T, S_G) is the maximum area of them.

Relative perimeter similarity, which is defined as follow (Zhang *et al.* 2014):

$$S_{per} = 1 - \frac{|P_T - P_G|}{Max(P_T, P_G)} \tag{32}$$

where S_{per} is the relative perimeter similarity, $|P_T$-$P_G|$ is the difference between the perimeter of the traditional buffer and the generalized buffer, and Max(P_T, P_G) is the maximum perimeters of them.

Offset of centroid of mass, which is defined as

$$\Delta_f = \sqrt{\Delta X^2 + \Delta Y^2} \tag{33}$$

where Δ_f is the offset of centroid mass, $\Delta X = X_T$-X_{Gi} ($i = 1, 2, 3, 4$), and is the difference between the X_T (X-coordinate of the centroid of the traditional buffer analysis) and the X_{Gi} (X-coordinate of centroid for generalized buffer analysis). $\Delta Y = Y_T$-Y_{Gi} ($i = 1, 2, 3, 4$) and is the difference between the Y_T (Y-coordinate of the centroid of the traditional buffer analysis) and the Y_{Gi} (Y-coordinate of centroid for generalized buffer analysis).

Comparison analysis for Dataset 1: For Dataset 1, the 6 parameters for point, line and polygon buffer zones are calculated, associated with the different parameters and shown in TABLE 4 through TABLE 6.

Comparison analysis for Dataset 2: For Dataset 2, the 6 parameters for point, line and polygon buffer zones are calculated, associated with the different parameters and shown in TABLE 7 through TABLE 9.

2) REMARKS FROM THE COMPARED RESULTS

- *From Traditional/Generalized Point Buffer Algorithm*

With TABLE 4 and TABLE 7, a few remarks can be drawn up:

(***a***) If the angle increment $\Delta\varphi$ limits to 0, the area of *GPIB* is close to the area of *TPB*, i.e.,

$$S_{GPIB} = \lim_{\Delta\varphi \to 0} \int_0^{2\pi} ds = S_{TPB} \tag{34}$$

This means that the generalized buffer zone is close to the traditional buffer zone when he angle increment $\Delta\varphi$ limits to 0.

TABLE 6. The difference value and the shape similarity index of the *GPLB* relative to *TPLB* for Dataset 1.

	Δ_{Area} (m^2)	$\Delta_{perimeter}$ (m)	Δ_f(m)	D(|a, b|) (%)	S_{area} (%)	S_{per} (%)
ΔS_1=1 m	5034	-1425	17.3	5.8	97.7	68.3
ΔS_2=5 m	8695	-1572	17.0	5.9	95.9	65.0
ΔS_3=10 m	11796	-1652	16.9	5.9	94.6	63.2
ΔS_4=15 m	16703	-1733	17.5	5.7	92.3	61.4

TABLE 7. The difference value and the shape similarity indexes of the *GPIB* relative to *TPB* for Dataset 2.

	Δ_{Area} (m^2)	$\Delta_{perimeter}$ (m)	Δ_f(m)	D(|a, b|) (%)	S_{area} (%)	S_{per} (%)
$\Delta\varphi_1$=1.5°	394	-3565	34.1	2.8	68.9	10.1
$\Delta\varphi_2$=3°	690	-1502	47.6	2.1	45.6	20.9
$\Delta\varphi_3$=6°	867	-494	44.5	2.0	31.5	44.7
$\Delta\varphi_4$=9°	974	-84	52.6	1.9	23.1	82.6

TABLE 8. The difference value and the shape similarity index of the *GLB* relative to *TLB* for Dataset 2.

	Δ_{Area} (m^2)	$\Delta_{perimeter}$ (m)	Δ_f(m)	D(|a, b|) (%)	S_{area} (%)	S_{per} (%)
ΔL_1=1 m	16050	-12493	8.5	10.6	80.3	11.1
ΔL_2=5 m	24757	-11085	11.3	8.1	69.6	12.3
ΔL_3=10 m	26349	-8741	14.7	6.4	67.6	15.1
ΔL_4=15 m	47936	-3446	26.1	3.7	41.0	31.1

TABLE 9. The difference value and the shape similarity index of the *GPLB* relative to *TPLB* for Dataset 2.

	Δ_{Area} (m^2)	$\Delta_{perimeter}$ (m)	Δ_f(m)	D(|a, b|) (%)	S_{area} (%)	S_{per} (%)
ΔS_1=1 m	7253	-2471	1.5	41.7	91.9	32.7
ΔS_2=5 m	8539	-2372	1.4	41.3	90.6	33.2
ΔS_3=10 m	10099	-2035	1.7	37.3	88.9	36.7
ΔS_4=15 m	32034	-1792	2.0	33.7	64.7	39.6

(***b***) With increasing the incremental angles ($\Delta\varphi$) from 1.5° to 9°, the area from GPB algorithm decreases from 1.6 times to 3.9 times for Dataset-1, from 1.8 times to 2.5 times for Dataset-2 relative to ones from TGB algorithm; the perimeters from GPB algorithm dramatically increases from 0.7 times to 0.9 times for Dataset-1, and rom 0.02 times to 0.4 times for Dataset-2. The offsets of mass centroid move from 1.4 m to 2.8 m for Dataset-1, and from 34.1 m to 52.6 m for Dataset-2.

(***c***) With increasing the incremental angle ($\Delta\varphi$) from 1.5° to 9°, the relative position similarities decrease from 41.4% to 26.3% for Dataset-1, and 2.8% to 1.9% for Dataset-2, respectively. The relative area similarities decrease from 97.7% to 91.2% for Dataset-1, and 68.9% to 23.1% for Dataset-2, respectively. The relative shape similarities increase increases from 66.1% to 73.3% for Dataset-1, and 10.1% to 82.6% for Dataset-2, respectively. This means that the relative area similarity, relative perimeter similarity and the relative position similarity slightly increase for Dataset-1, while the relative position similarity largely increases for Dataset-2.

- *From Traditional/Generalized Line Buffer Algorithm*

With TABLE 5 and TABLE 8, a few remarks can be drawn up:

(***a***) If the distance increment ΔL limit to 0, the area of *GLB* zone is close to the area of *TLB* zone, i.e.

$$S_{GLB} = \lim_{\Delta L \to 0} \int_0^L ds = S_{TLB} \tag{35}$$

This means that the generalized line buffer zone is close to the traditional buffer zone when the line increment, ΔL limit to 0.

(***b***) With increasing the incremental distance (ΔL) from 1 m to 15 m, the areas of *GLB* zone decrease decreases from 1.6 times to 3.3 times for Dataset-1, and from 1.5 times to 3 times for Dataset-2, respectively; the perimeters increase from 0.7 times to 0.9 times for Dataset-1, and from 0.3 times to 0.9 times for Dataset-2, respectively. The offsets of mass centroid move from 0.9 m to 2.1 m for Dataset-1 and from 8.5 m to 26.1 m for Dataset-2, respectively.

(***c***) With increasing the incremental distance (ΔL) from 1 m to 15 m, the relative position similarities decrease from 53.7% to 33.8% for Dataset-1, and 10.6% to 3.7% for Dataset-2, respectively. The relative area similarities decrease from 98.5% to 94.9% for Dataset-1, and 80.3% to 41.0% or Dataset-2, respectively. The relative perimeter similarities increase with increasing the incremental distance increases from 72.9% to 78.6% for Dataset-1, and 11.1% to 31.1% or Dataset-2, respectively. This means that the relative area similarity, relative perimeter similarity and the relative position similarity slightly increase for Dataset-1, while the relative position similarity largely increase for Dataset-2.

- *From Traditional/Generalized Polygon Buffer Algorithm*

With TABLE 6 and TABLE 9, a few remarks can be drawn up:

(***a***) If the arc increment ΔS approaches to 0, the area of *GPLB* zone is close to the area of *TPLB* zone, i.e.

$$S_{GPLB} = \lim_{\Delta S \to 0} \int_0^S ds = S_{TPLB} \tag{36}$$

This means that the generalized polygon buffer zone is close to the traditional polygon buffer zone when the arc increment ΔS limits to 0.

(***b***) With increasing the incremental arc length (ΔS) from 1m to 15 m, the areas of *GPLB* decrease from 1.7 times to 3.3 times for Dataset-1, and from 1.2 times to 4.4 times for Dataset-2, respectively. and the perimeters increase from 1.1 times to 1.2 times for Dataset-1, and 3 from 0.7 times to 0.9 times for Dataset-2, respectively. The offsets of mass centroid move from 17.3 m to 17.5 m for Dataset-1 and from 1.4 m to 2.0 m for Dataset-2.

(***c***) With increasing the incremental arc length (ΔS) from 1 m to 15 m, the relative position similarities decrease from 5.8% to 5.7% for Dataset-1, and 41.7% to 33.7% for Dataset-2, respectively. The relative area similarities decrease from 97.7% to 92.3% for Dataset-1, and 91.9% to 64.7% for Dataset-2, respectively. The relative perimeter similarities increase from 68.3% to 61.4% for Dataset-1, and 32.7% to 39.6% for Dataset-2, respectively. This means that the relative area similarity and relative perimeter similarity slightly increase, while the relative position similarity quickly increase for Dataset-1, and the relative area similarity, relative perimeter similarity and relative position similarity quickly increase for Dataset-2.

With the analysis above, the conclusions below can be drawn up:

(***a***) The differences between traditional and generalized buffer zone for the point, line and polygon buffer zone generation are different, upon the variables, such as incremental angles ($\Delta\varphi$), incremental length(ΔL), and incremental arc length (ΔS), respectively. In particular, when these increments approach to 0, the traditional and generalized buffer zones are the same.

(***b***) With increasing incremental angle ($\Delta\varphi$), incremental length (ΔL), and incremental arc length (ΔS), the relative position similarity in GB zones for the point, line and polygon buffering zone generation decreases. The smaller the data density is, the bigger the relative position similarity changes, and vice versa. The higher the relative position similarity is, the simpler the generalized buffer shape is, even approach to the traditional buffer shape, and visa versa.

(***c***) With increasing incremental angle ($\Delta\varphi$), incremental length (ΔL), and incremental arc length (ΔS), the relative area similarity of the generalized buffering zone decreases. The smaller data density is, the smaller relative area similarity changes, and vice versa. The higher the relative area similarity is, the closer the generalized buffer shape is to the traditional buffer shaper is, and vice versa.

(***d***) With increasing incremental angle ($\Delta\varphi$), incremental length (ΔL), and incremental arc length (ΔS), the perimeters for point, line and polygon buffer zones increase. In addition, the data density is small, and the change in the relative perimeter similarity is small. The data density is large, and the relative perimeter similarity changes greatly. The higher the relative perimeter similarity, the simpler the generalized buffer shape, and the closer the generalized buffer shape is to the traditional buffer.

IV. CONCLUSION

The main contribution of this research is the development of a radial and breakthrough algorithm, called *generalized buffer algorithm (GBA)* for point, line, and polygon buffer generation. This algorithm is challenging the traditional buffering algorithm which has been for over 60 years. This algorithm stems from the fact that traditional buffering algorithms are based on a fixed buffer distance without considering difference of neighbor instance's attributes in practice. The proposed *GBA* simultaneously considers homogeneous and the correlation of both spatial data and attribute data of two instances, consequently, the buffer distance varies upon the characteristics of two instances.

The details of the proposed *GBA* is described in text. Summarily, first, spatial and no-spatial attributes are selected. Second, the *R*-proximity relationships between two instances are determined in accordance with the selected spatial attributes. Third, candidates of boundary points of buffering zone are selected based on the *R-proximity* relationship. Forth, boundary points of buffering zone are determined using non-spatial attributes to decide if the candidates of boundary points of buffer are prevalent events. Finally, the boundary points are connected to form the boundary of the generalized buffer zone. To validate the advances of the proposed method, the point, line and polygon data sets from Beijing and Bao'an District, City of Shenzhen, China are used. The experimental results and comparison analyses, using 6 indexes calculated from traditional and generalized buffer algorithms, discovered that:

(i) The proposed GBA can accurately reflect the real situation of the buffering zone, and improve the deficiency and accuracy of traditional buffering algorithm.

(ii) From 6 indexes, GBA approaches to the traditional point/line/polygon buffering algorithms when the incremental angle ($\Delta\varphi$), the incremental length (ΔL), and the incremental arc length (ΔS) approach to zero.

ACKNOWLEDGMENT

The authors would like to thank the editor-in-chief, the associate editor, and the reviewers for their insightful comments and suggestions.

REFERENCES

[1] M. Shimrat, "Algorithm 112: Position of point relative to polygon," *Commun. ACM*, vol. 5, no. 8, p. 434, Aug. 1962, doi: 10.1145/368637.368653.

[2] V. Chvátal, "On certain polytopes associated with graphs," *J. Combinat. Theory*, vol. 18, no. 2, pp. 138–154, 1975, doi: 10.1016/0095-8956(75)90041-6.

[3] F. P. Preparata and M. I. Shamos, *Computational Geometry: An Introduction*. Berlin, Germany: Springer-Verlag, 1985.

[4] A. Aggarwal, B. Chazelle, L. Guibas, "Parallel computational geometry," in *Proc. IEEE Symp. Found. Comput. Sci.*, 1985, pp. 293–327. [Online]. [Online]. Available: https://link.springer.com/article/10.1007/BF01762120

[5] H. Wu, "On the establishment of GIS buffer zone," *J. Wuhan Univ. Surv. Mapping*, vol. 22, no. 4, pp. 358–366, 1997.

[6] H. Wu, J. Gong, and D. Li, "Buffer curve and buffer generation algorithm in aid of edge-constrained triangle network," *ACTA Geodaetica Cartographica Sinica*, vol. 28, no. 4, pp. 75–79, 1999.

[7] J. Chen, R. Zhao, and Z. Li, "Voronoi-based K-order neighbour relations for spatial analysis," *ISPRS J. Photogramm. Remote Sens.*, vol. 59, nos. 1–2, pp. 60–72, Aug. 2004, doi: 10.1016/j.isprsjprs.2004.04.001.
[8] P. Dong, "An effective buffer generation method in GIS," in *Proc. IEEE IGARSS*, Aug. 2003, pp. 3706–3708, doi: 10.1109/IGARSS.2003.1295244.
[9] Y. Ren, "A way to speed up buffer generalization by Douglas-Peucker algorithm," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, Sep. 2004, pp. 2916–2919. [Online]. Available: https://ieeexplore.ieee.org/document/1370304
[10] E. Er, I. Kilinc, G. Gezici, and B. Baykal, "A buffer zone computation algorithm for corridor rendering in GIS," in *Proc. 24th Int. Symp. Comput. Inf. Sci.*, Sep. 2009, pp. 432–435. [Online]. Available: https://ieeexplore.ieee.org/document/5291855
[11] W. Pan and Y. Li, "Random algorithm for buffer generation," *Comput. Eng.*, vol. 36, no. 14, pp. 70–73, 2020.
[12] L. Liu and J. Min, "Analysis of traditional space buffer generation algorithm," *Comput. CD Softw. Appl.*, vol. 35, no. 20, p. 35, 2011.
[13] F. Bai and F. Zhang, "Multi-objective buffer generation algorithms," *GNSS World China*, vol. 36, no. 1, pp. 38–41, 2011.
[14] X. Zhang, T. Ai, J. Stoter, and X. Zhao, "Data matching of building polygons at multiple map scales improved by contextual information and relaxation," *ISPRS J. Photogramm. Remote Sens.*, vol. 92, pp. 147–163, Jun. 2014, doi: 10.1016/j.isprsjprs.2014.03.010.
[15] Q. Chen and G. Liu, "Algorithm for generating irregular buffer zone with constraints," *Geol. Sci. Technol. Inf.*, vol. 33, no. 4, pp. 213–218, 2014.
[16] X. Xu and W. Liu, "Research on vector-raster mixed algorithm of linear buffer generation," *Comput. Eng. Appl.*, vol. 50, no. 4, pp. 152–156, 2014.
[17] C. J. Yang, Y. Zhao, F. Wang, C. Fang, and S. Wu, "Algorithm for rapid buffer analysis in three-dimensional digital Earth," *J. Remote Sens.*, vol. 18, no. 2, pp. 353–364, 2014, doi: 10.11834/jrs.20143108.
[18] J. Fan, M. Ji, G. Gu, and Y. Sun, "Optimization approaches to mpi and area merging-based parallel buffer algorithm," *Boletim de Ciáncias Geodésicas*, vol. 20, no. 2, pp. 237–256, Jun. 2014, doi: 10.1590/S1982-21702014000200015.
[19] W. Enze, G. Baolu, X. Cheng, and C. Junjie, "Edge match algorithm based on attributes similarity and multiple buffers," *Bull. Sci. Technol.*, vol. 32, no. 4, pp. 174–177, 2016.
[20] W. Tuo-Di, Z. Ling-Jun, W. Li-Zhe, and C. La-Jiao, "Parallel research and optimization of buffer algorithm based on equivalent arc partition," *Remote Sens. Inf.*, vol. 31, no. 4, pp. 147–152, 2016.
[21] Q. Dong and J. Chen, "A tile-based method for geodesic buffer generation in a virtual globe," *Int. J. Geograph. Inf. Sci.*, vol. 32, no. 2, pp. 302–323, Feb. 2018, doi: 10.1080/13658816.2017.1372764.
[22] N. Chen, "Influence of conversion on the location of points and lines: The change of location entropy and the probability of a vector point inside the converted grid point," *ISPRS J. Photogramm. Remote Sens.*, vol. 137, pp. 84–96, Mar. 2018, doi: 10.1016/j.isprsjprs.2018.01.011.
[23] J.-H. Lee and C.-H. Won, "Topology preserving relaxation labeling for nonrigid point matching," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 2, pp. 427–432, Feb. 2011, doi: 10.1109/TPAMI.2010.179.
[24] M. Ma, Y. Wu, L. Chen, J. Li, and N. Jing, "Interactive and online buffer-overlay analytics of large-scale spatial data," *ISPRS Int. J. Geo-Inf.*, vol. 8, no. 1, p. 21, Jan. 2019, doi: 10.3390/ijgi8010021.
[25] X. Ma and G. Chang, "Influence of Vehicle Sound Source Height on Traffic Noise Forecast and Sound Barrier Design," *Sci. Technol. Eng.*, vol. 36, pp. 180–184, May 2018.

GUOQING ZHOU (Senior Member, IEEE) received the Ph.D. degree from Wuhan University, Wuhan, China, in 1994.

He was a Visiting Scholar with the Department of Computer Science and Technology, Tsinghua University, Beijing, China, and a Postdoctoral Researcher with the Institute of Information Science, Beijing Jiaotong University, Beijing. From 1996 to 1998, he continued his research as an Alexander von Humboldt Fellow with the Technical University of Berlin, Berlin, Germany. From 1998 to 2000, he was a Postdoctoral Researcher with The Ohio State University, Columbus, OH, USA. He was an Assistant Professor, an Associate Professor, and a Full Professor with Old Dominion University, Norfolk, VA, USA, in 2000, 2005, and 2010, respectively. He has authored five books and more than 402 refereed articles.

RONGTING ZHANG received the Ph.D. degree in instrumentation science and technology from the School of Precision Instrument and Opto-Electronics Engineering, Tianjin University, China, in 2020. He is currently a Lecturer with the College of Geomatics Science and Technology, Nanjing Tech University, Nanjing, China.

SHENGXIN HUANG was born in 1994. She received the M.S. degree from the College of Geomatics and Geoinformation, Guilin University of Technology, Guilin, China, in 2020. Her research interest includes spatial analysis.