A method of approximately simulating buffers based on mathematical equations for accelerating buffer analysis

LI Jun, QIN Qiming, CHEN Chao, ZHAO Yue, XIE Chao

Institute of Remote Sensing and Geographic Information Systems, Peking University, Beijing 100871, *China*

Abstract: Buffer analysis is a very important spatial function of Geographic Information System (GIS). However, its efficiency in terms of time and space when used to process massive amounts of geographic data is very limited. To solve this problem, we define an approximate representation form—equational buffer which uses a mathematical equation to simulate the boundary of a buffer in GIS. Using equational buffer, buffer analysis is converted from a process composed of complicated geometric calculations to a process composed of simple algebraic calculations, leading to increased computational efficiency. This paper introduces the definition, pattern, application method and performance of the equational buffer. We also apply the equational buffer in the process of d ata filtering when estimating the real-time traffic information of China's highways based on floating car data, and the experiment result validates that equational buffers are highly efficient in processing massive amounts of geographic data.

Key words: buffer analysis, buffer representation, data filtering, equational buffer, large geographic dataset

CLC number: P208 **Document code:** A

Citation format: **Li J, Qin Q M, Chen C, Zhao Y and Xie C. 2013. A method of approximately simulating buffers based on mathematical equations for accelerating buffer analysis. Journal of Remote Sensing, 1 7(5): 1131 – 1145 [DOI: 10.11834/jrs.20132166]**

1 INTRODUCTION

Buffer analysis is an important function of Geographic I nformation System (GIS), which may assist the retrieval and processing of geographic information. It is widely used in various aspects such as transportation management, regional e conomic analysis, city planning and environment monitoring. Buffer analysis consists of two steps: buffer generation and buffer application. Buffer generation is the main key process that determines the efficiency and accuracy of buffer analysis. As a result, many researchers studied buffer generation methods in o rder to improve buffer analysis.

For a vector buffer, the core of generation algorithms is the generation of the double-parallel-line algorithm which is presumed to be a very simple process. But when geographic features are complicated or the generated parallel lines are overlapped, the process becomes complicated. Several researches focus on addressing these exceptional cases. Ghosh (1990, 1991) developed an algebraic system of geometric shapes within which one can add and subtract shapes exactly as one adds and subtracts within the integer number system. Wu (1997) introduced the salient arc method based on the double mathematical model. Zalik, et al. (1999, 2003) presented an algorithm for constructing the geometric outlines of a given set of line segments using a sweep-line approach. Dong, et al. (2004) made a further improvement on the generation method of the salient arc buffer using the rotation-point transform formula and recursion method. Wang, et al. (2007) proposed a buffer generation algorithm based on border line tracing, which extracted complete closed curves by tracing and thus avoided complicated geometric calculations.

For a raster buffer, the generation algorithm is simple and easy to implement. The commonly used generation algorithms for a raster buffer include: the run-length brush overlay algorithm (Taloy, 1994), the morphological dilation algorithm (Hu, et al., 2006) and the filling algorithm (Guo, 1997). Xiang and Stratton (1996) proposed a variable buffer generation method in a raster environment, which maps the stream buffer with variable width using the B function. Wang, et al. (2010) proposed the raster-based distance computing algorithm and a novel algorithm of buffer construction based on run-length encoding.

Although much progress has been made in the robustness and efficiency of the buffer generation method, buffer analysis still faces the following challenges. First, the necessary geometric calculations in the generation method of a vector buffer are c

Received: 2012-05-21; **Accepted:** 2012-11-18; **Version of record first published:** 2012-11-25

Foundation: National High Technology Research and Development Program of China (No. 2012AA121305); National Science and Technology Major Projects of High Resolution Earth Observation Systems

First author biography: LI Jun (1987—), male, Ph. D. candidate. He majors in GIS for transportation and geographic data mining. E-mail: newton2069@126.com

Corresponding author biography: QIN Qiming (1955—), male, professor. His current research interests include quantitative remote sensing, spatial analysis modeling and application of GIS. E-mail: qmqink@pku.edu.cn

omplicated and it is difficult to implement. Second, when a vector buffer is applied in an analysis, due to its representation method the judgment of inclusion relation between two features needs to be performed frequently, causing buffer analysis very time-consuming. Finally, a raster buffer uses a collection of raster cells to store the position of the buffer. Therefore, the representation accuracy is susceptible to the raster cell size and the storage of a raster buffer takes a lot of physical memory.

In the past applications, the above mentioned problems are yet not critical ones. Along with the development of GIS, computer technology and telecommunication technology, geographic information technology were used in more and more aspects and the amount of obtained geographic data is unprecedentedly large. With the increased availability of larger geographic datasets, the application of traditional buffer analysis method becomes unsatisfactory because of the above problems. Previous studies mainly focused on the buffer generation methods, and researchers hoped to increase the efficiency of buffer analysis through improving the generation method of buffer. However, to seek a simpler representation method for a buffer is also an effective solution.

After checking the existing method of buffer analysis, and inspired by the view point in analytic geometry that "a mathematical equation represents a graph, in other words it is the mathematical expression or form of a graph", we present the idea of using the graph which can be represented by a mathematical equation to simulate a buffer's boundary. Then based on the idea, we define an approximate representation form of a buffer—equational buffer—inspired by the formula of a gravitational isoline. The application of equational buffers converts buffer analysis from a process composed of complicated geometric c alculations to a process composed of simple algebraic calculations, increasing the computational efficiency.

2 EQUATIONAL BUFFER

2.1 Reflections on the traditional buffer

Because the accuracy of a raster buffer is low and it is susceptible to memory limitation, people often use a vector buffer when dealing with a large geographic dataset. Only the vector buffer is discussed later in the paper, unless otherwise specified. As previously showed, the existing buffer analysis method cannot avoid the complicated judgment of inclusion relation among features. So how can we increase the efficiency of buffer analysis? We seek answers from the process of traditional buffer analysis. For example, there is a task to analyze how many residential areas are within the service area of a hospital (Fig. 1). Point A represents the hospital, each point of point set $\boldsymbol{B}$ represents a residential area and the service distance of the hospital is three kilometers. The traditional analysis is implemented as below: first, we need to generate a buffer around Point A—a series of points $\boldsymbol{P}_A$ are generated to fit Circle O_A whose centre is Point A and radius is three kilometers. This point array $\boldsymbol{P}_A$ is the buffer of point A and intrinsically a polygon vector. Then, we figure out which points in point set $\boldsymbol{B}$ are located within the buffer $\boldsymbol{P}_A$, that is, to judge the inclusion relation between each point in point set $\boldsymbol{B}$ and the polygon vector $\boldsymbol{P}_A$. The whole process is composed of complicated geometric calculations which are time-consuming. In fact, our ultimate aim is to know which points in point set $\boldsymbol{B}$ are within Circle O_A, and in a mathematical point of view it equals to judging if the point coordinate (m, n) meets the inequality $\sqrt{(x-x_0)^2+(y-y_0)^2} \leqslant 3$ which is the equation of Circle O_A, where (x_0, y_0) is the coordinates of Point A. In short, several complicated geometric operations need to be done in GIS to accomplish such a simple algebraic calculation. If we assume that buffer boundaries of all types of geographic features can be simulated by mathematical equations, buffers can be expressed by mathematical equations, and the generation and application of buffers could be accomplished through simple algebraic calculations rather than complicated geometric calculations. If that is the case, the challenges which the existing buffer analysis methods are facing would be solved. The most essential difference between this kind of buffer and the existing buffer is that the new buffer is not represented by entities—a set of discrete points, but by mathematical equations, so we call it equational buffer.

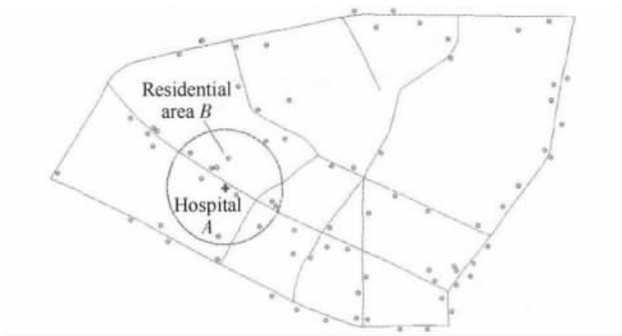


Fig. 1 Traditional buffer analysis

2.2 Definition of the equational buffer

The key to construct the equational buffer is to find the mathematical equation that fits well the boundaries of traditional buffers. The formulas of equational buffers corresponding to various types of features are introduced in this section. Features in GIS include three types: point, polyline as well as polygon. A polygon can also be represented by polyline features and the basic composition unit of a polyline is the line segment. Consequently, the basic geometric unit for GIS is the point and the line segment. This paper presents the definition of equational buffer taking point and line segment as examples.

Because buffer analysis is mainly based on distance, we suggest that equational buffer is an unbroken, closed and distance-related buffer, and of course it should be able to be expressed in a mathematical equation. The formula of gravitational isolines exactly meets such requirements, so we e stablish the equational buffer by borrowing the idea of its f ormula and use a gravitational isoline of a geographic feature to simulate the boundary of the feature's closest buffer. The formulas of equational buffers of point and line segment are d educed as below.

2.2.1 Buffer of point

Suppose that Point A's mass is m, which is located at $(x_0,$

y_0). Another Point B's mass is m', whose distance to Point A is d, and the distance of Point C to Point A is r (Fig. 2).

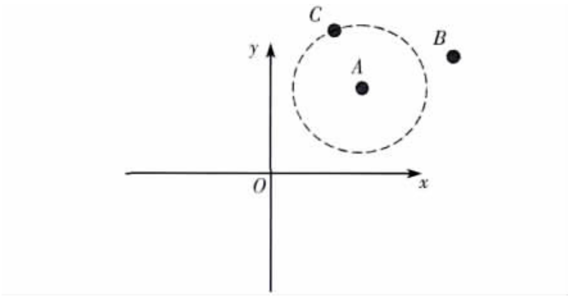


Fig. 2 Deduction of equational buffer of a point

The gravitational force (Ma, 2004) between Point A and Point B is $F = G\frac{mm'}{d^2}$. The gravitational force at Point C is $G\frac{mm'}{r^2}$, and the equation of the gravitational isoline at Poin C is $F = G\frac{mm'}{r^2}$, where r is a constant. For convenient computation, each side of the equation of gravitational isoline is divided by Gmm', and the buffer of the point is defined as $Buffer_{\text{point}}$: $\frac{1}{d^2} \geq \frac{1}{r^2}$, that is also

$$Buffer_{\text{point}}: \sqrt{(x - x_0)^2 + (y - y_0)^2} \leq r^2 \quad (1)$$

where r is the buffer radius.

2.2.2 Buffer of line segment

In order to calculate conveniently, a temporary coordinate system $x'o'y'$ is defined. Line segment AB is put on the x' axis with its middle point at the original point o'. Suppose that the mass of line segment AB is m and length is l, and there is another point C in location (x', y') whose mass is m' (Fig. 3).

The linear density of segment AB is $\rho = m/l$. Line segment

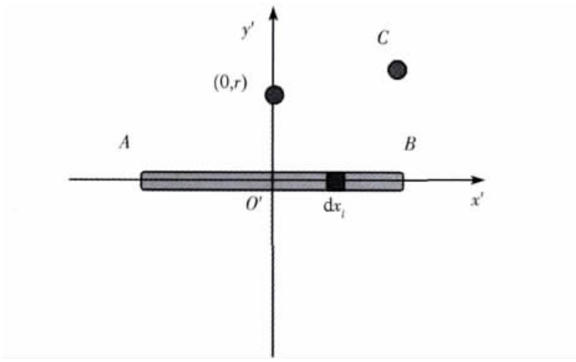


Fig. 3 Deduction of equational buffer of a line segment

AB is divided into numerous small pieces dx_l, and the x' axis and y' axis components of the gravitational force between the piece at $(x_l, 0)$ and Point C are respectively:

$$dF_{x'} = \frac{G\rho m'(x' - x_l)\,dx_l}{\left((x' - x_l)^2 + y'^2\right)^{3/2}}$$

$$dF_{y'} = \frac{G\rho m' y' dx_l}{\left((x' - x_l)^2 + y'^2\right)^{3/2}}$$

The x' axis and y' axis component of the gravitational force between line segment AB and Point C are respectively:

$$F_{x'} = \int_{-\frac{l}{2}}^{\frac{l}{2}} dF_{x'} = G\rho m'\left(\frac{1}{\sqrt{\left(x' + \frac{l}{2}\right)^2 + y'^2}} - \frac{1}{\sqrt{\left(x' - \frac{l}{2}\right)^2 + y'^2}}\right)$$

$$F_{y'} = \int_{-\frac{l}{2}}^{\frac{l}{2}} dF_{y'} = \frac{G\rho m'}{y'}\left(\frac{x' - l/2}{\sqrt{\left(x' - \frac{l}{2}\right)^2 + y'^2}} - \frac{x' + l/2}{\sqrt{\left(x' + \frac{l}{2}\right)^2 + y'^2}}\right), (y' \neq 0)$$

The total gravitational force between Segment AB and Point C is $F_G(x', y') = \sqrt{F_{x'}^2 + F_{y'}^2}$, that is

$$F_G(x', y') = \begin{cases} G\rho m'\sqrt{\left(\frac{1}{\sqrt{(x' + l/2)^2 + y'^2}} - \frac{1}{\sqrt{(x' - l/2)^2 + y'^2}}\right)^2 + \left(\frac{x' - l/2}{y'\sqrt{(x' - l/2)^2 + y'^2}} - \frac{x' + l/2}{y'\sqrt{(x' + l/2)^2 + y'^2}}\right)^2}, (y' \neq 0) \\ G\rho m'\left|\frac{1}{|x' + l/2|} - \frac{1}{|x' - l/2|}\right|, (y' = 0) \end{cases}$$

The gravitational force in location $(0, r)$ is $F_G(0, r)$, and the e quation of gravitational isoline here is $F_G = F_G(0, r)$. Also in order to calculate conveniently, we set $F'(x', y') = \frac{F_G(x', y')}{G\rho m'}$, then the buffer of line segment in the temporary coordinate system $x'o'y'$ is

$$F'(x', y') \geq \frac{F_G(0, r)}{G\rho m'} = \frac{l}{r\sqrt{\frac{l^2}{4} + r^2}}$$

where r is buffer radius.

The above calculations are done in the temporary coordinate system $x'o'y'$. Then the equation of the buffer is transformed to a geographic coordinate system xoy, and the transformation formula is as below:

$$x' = TranX(x) = (x - x_m)\cos t + (y - y_m)\sin t$$

$$y' = TranY(y) = (x_m - x)\sin t + (y - y_m)\cos t$$

where $TranX$ and $TranY$ are transfer function (x_m, y_m) is the g eographic coordinates of middle point of line segment AB, t is the angle between vector $((x_1, y_1), (x_2, y_2))$ and the x axis.

Set $F(x, y) = F'(x', y') = F'(TranX(x), TranY(y))$, the buffer of line segment is defined as

$$Buffer_{\text{line}}: F(x, y) \geq \frac{l}{r\sqrt{\frac{l^2}{4} + r^2}} \quad (2)$$

Point buffer and line segment buffer are the closed curves expressed by Eq. (1) and Eq. (2) respectively. The buffer of the polyline is the union of line segment buffers which are calculated by Eq. (2). The buffer of the multi-feature is the union of buffers of each single feature within the multi-feature. Polygon is not taken into account in this paper presently.

2.3 Pattern of equational buffer

To give more intuitive understanding about equational buffers, patterns of equational buffers with various radiuses are d isplayed. In Fig. 4, there are four buffers of a point, and the buffer radiuses are 20 m, 40 m, 60 m and 80 m from inside to outside. In Fig. 5, there are four buffers of a line segment, and the buffer radiuses are 40 m, 80 m, 120 m and 160 m, respectively.

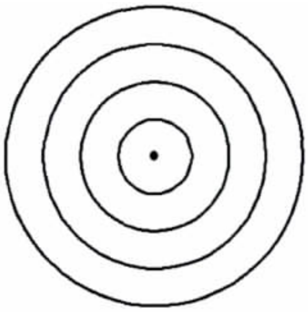


Fig.4 Pattern of a point's equational buffer

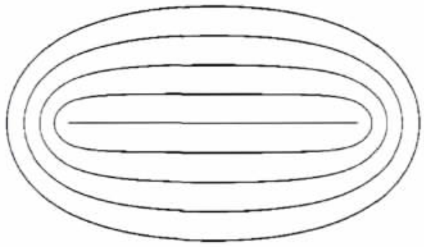


Fig.5 Pattern of a line segment's equational buffer

2.4 Application method of equational buffer

Taking geographic information retrieval (that is to query the geographic features inside a buffer) as an example, the application method of the equational buffer is introduced in this subsection.

(1) Point buffer

When judging if the point feature $T(x_t, y_t)$ is within r buffer radius of Point $A(x_0, y_0)$, we put the coordinate of point feature into Eq. (1), yielding $\sqrt{(x_t - x_0)^2 + (y_t - y_0)^2} \leqslant r$. When the inequality holds, then Point T is within the buffer of Point A; otherwise, Point T is outside the buffer of Point A. When the buffer radius changes, the process is repeated using the corresponding changed parameter r.

(2) Line segment buffer

When judging if the point feature $T(x_t, y_t)$ is within r buffer radius of line segment AB where $A = (x_1, y_1)$ and $B = (x_2, y_2)$, we put the coordinate of point feature into Eq. (2) yielding

$$F(x_t, y_t) \geqslant l \Big/ r\sqrt{\frac{l^2}{4} + r^2}.$$

When the inequality holds, then Point T is within the buffer of line segment AB; otherwise, Point T is outside the buffer of line segment AB.

(3) Polyline buffer

There is a Polyline $Q(P_1, P_2, \cdots, P_n)$. When judging if the point feature $T(x_t, y_t)$ is within r buffer radius of Polyline Q, we need to see if the point feature is within each buffer of the subsegment of Polyline Q. As long as one inequality holds, the point feature T is within the buffer of Polyline Q. Otherwise, it is o utside the buffer of Polyline Q.

3 PERFORMANCE OF EQUATIONAL BUFFER

3.1 Application efficiency

In this section, the application efficiencies of a traditional buffer and an equational buffer are compared, including space e fficiency and time efficiency. Space efficiency is the size of physical memory for storing buffer, and time efficiency is the computation time spent when applying a buffer, which again consists of generation time efficiency and using time efficiency.

(1) Space efficiency of buffer: a traditional buffer is represented in the form of polygon vector. The size of physical m emory for storing the coordinates of node points of polygon is determined by the number, size and shape of features. So a lot of storage space is needed when a large dataset is processed, making the space efficiency of traditional buffer very low. B ecause of the representation method of equational buffer, the buffers of different features are expressed by a common equation with only different coefficients. So only the common mathematical equation needs to be stored in memory, which costs less than 1 KB.

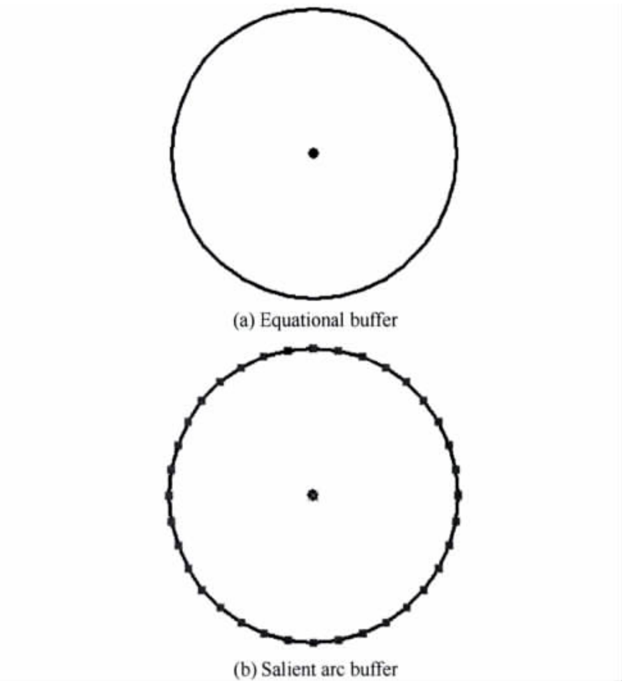
(2) Generation time efficiency. There are several traditional algorithms for buffer generation: the salient arc method and the angular bisector method, and each algorithm costs a certain time. Similarly, when the amount of processing objects is very large, generating buffers of all features will take a lot of time. In particular, frequent changes of feature location or buffer radius will increase the duration of computation time even more, bringing inconveniences to the application of buffer analysis. On the c ontrary, the equational buffer of an arbitrary feature is just a mathematical equation with an unchanged form. Its generation e quals to the parameter selection of equation and takes little time.

(3) Using time efficiency for a traditional buffer, the judgment of inclusion relation between other features and a buffer is to see if other features are within the polygon vector. The c ommonly used algorithms include the ray line algorithm (Preparata & Shamos, 1985; Taloy 1998) and the azimuth s ummation algorithm (Yan, et al., 2000), and both algorithms i nvolve complicated geometric calculations. Nevertheless, the e quational buffer is different from the traditional buffer in the sense that an equational buffer is a mathematical equation. Thus to judge if a feature is within a buffer is just to see if the coordinates of features can make the buffer inequality hold, which takes less time. Its time efficiency is much higher especially when processing features of point type.

3.2 Simulating accuracy analysis

An equational buffer uses an unbroken smooth graph to simulate the buffer boundary, and as a result the simulating error exists. We set the buffer generated by the salient arc algorithm as true value, and analyze the simulating error distribution of equational buffers. The simulating accuracies of equational buffers of the point and line features are discussed respectively.

Although the boundary of a true buffer is a continuous curve, in a vector environment its representation has to be discretised in order to be stored and analyzed in a computer. For a point feature, some discrete points are used to fit a circular buffer. This kind of representation method causes some errors in the traditional buffer. In contrast, the equational buffer of a point is represented by the equation of a circle, so it exactly simulates the true buffer (Fig. 6). It can be concluded that when it comes to a point feature, no simulating error exists in the equational buffer and it is more accurate than the traditional buffer.

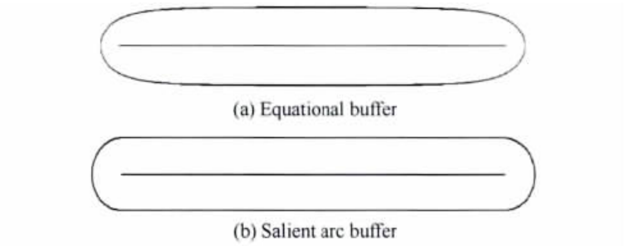

(a) Equational buffer

(b) Salient arc buffer

Fig. 6 The equational buffer and the salient arc buffer of a point

Part of the true buffer of a line segment is lines parallel to the segment, and two semicircles link them at both ends. For a line segment feature, the equational buffer is different from the salient arc buffer: it matches with the salient-arc buffer at middle points aside but is smaller in other parts. Fig. 7 demonstrates p atterns of two kinds of buffers of the same line segment. The length of the segment is 200 m and its buffer radius is 20 m. The simulating error of an equational buffer is analyzed below. Around two end points, the difference between the equational buffer and the true value is fairly big. But the area in this part is comparably small, so in common case the influence is not big and the error can be overlooked. The error in one side of the buffer needs to be analyzed. Two kinds of buffers are put in the same coordinate system and the simulating error where the x c oordinate is from 0 m to 100 m is analyzed (Fig. 8).

The simulating error where the x coordinate is x is $\varepsilon_x = \frac{y_1 - y_2}{y_1}$, where y_1 is the y coordinate of salient-arc buffer whose x coordinate is x, y_2 is the y coordinate of the equational buffer where the x coordinate is x. The simulating errors are calculated with buffer radius 1 m, 10 m, 30 m, 100 m and 400 m respectively, and the results are shown in Fig. 9.


(a) Equational buffer

(b) Salient arc buffer

Fig.7 The equational buffer and the salient arc buffer of a line segment

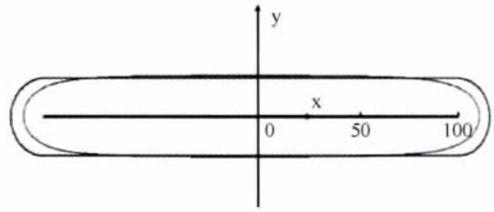


Fig.8 Schematic diagram of accuracy analysis

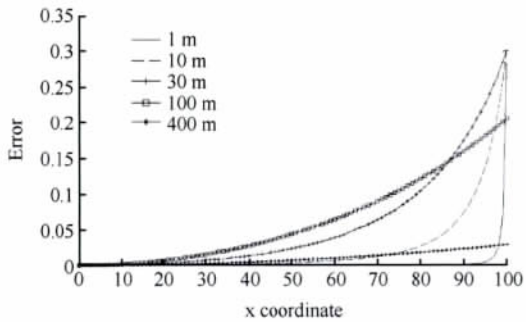


Fig.9 Error distribution with buffer radius 1 m, 10 m, 30 m, 100 m and 400 m

The experiment results show that the simulating error is smaller than 7% in 99% region for 1 m of buffer radius, smaller than 10% in 93% region for 10 m; smaller than 10% in 80% region for 30 m, smaller than 20% in 99% region for 100 m, smaller than 3% for 400 m. The above data indicates that when there is a large difference between buffer radius and the segment length, no matter larger or smaller, the simulating error is small. When the buffer radius is close to segment length, the error b ecomes large. However, the error in nearly 80% of region is less than 10%, and all errors are under 20% except a very small part.

Although simulating errors exist in equational buffers, these errors are acceptable in the buffer analysis without a very strict accuracy requirement. Taking geographic information retrieval as an illustration example, because the area of equational buffer is smaller than that of the true buffer, the existence of simulating errors would lead to some data omitted. But the application of e quational buffers instead of traditional buffers would cause only a little part of data missing, which influences little on the result while greatly increases the computational efficiency.

4 A CASE STUDY

To further demonstrate the equational buffer's advantages in dealing with a large geographic dataset, in this section we apply it in the real-time traffic information estimation based on floating car data (Draijer, et al., 2000). We are currently involved in a r esearch program in which real-time positioning data are collected from the GPS-equipped vehicles across China, and then these signals were used to estimate the real-time traffic information of China's highway. Because the amount of the data in this program is hugely large, it is very suitable for testing the validity of equational buffer in processing massive geographic data. The data source is GPS positioning data of vehicles and road data from digital maps. Both types of data have unavoidable measurement errors, so the first step in traffic information estimating is map matching (Lu & Cui, 1999) —identifying the road segment, to which each GPS positioning point belongs. At this stage the purpose of the program is to estimate the traffic information of highways. However, the vehicles may travel on various levels of roads (highway, national road, provincial road) or park in a parking zone and they do not have effect to estimate highways' traffic information. To improve the computational efficiency, before map matching we conduct a coarse filtering on the original GPS points—weeding out the invalid positioning points which are far away from any highway. The coarse filtering is usually implemented using the buffers of highways. In this section, we realize the filtering procedure using the traditional buffer and the equational buffer respectively and compare their results. The following experiment is done in a computer with a 2.8 GHz dual core CPU and a 2 GB memory.

We select part of North China as the study area with the g eographic range of 37°44′41″N—40°44′47″N and 114°7′10″E —117°53′1″E. If the update frequency of real-time traffic i nformation is half an hour, it means the traffic information of highways' each segment is estimated using the data received within half an hour. All GPS positioning data collected by one data exchange centre from 01:30 to 02:00 on September 19, 2010 are selected as experiment data. There is a total of 194205 GPS positioning points in this dataset and Fig. 10 shows the spatial distribution of those data locating within the study area. A rcEngine, a mature business product in GIS is the use of the salient arc method to generate a buffer. We take it as an example of traditional buffer analysis to implement data filtering. Setting the buffer radius as 500 m, the buffers of highways' all segments are generated and the GPS positioning points falling outside the buffers are removed. Excluding the generation time of buffers, the entire filtering process takes 415.61 s and 11171 effective GPS positioning points are screened out (Fig. 11). We also use e quational buffer to realize the same function. Setting the parameter for buffer radius as 500 m, the GPS positioning points falling inside the buffers are retrieved. The entire process which includes buffer generation takes 76.27 s and 10564 effective GPS positioning points are screened out (Fig. 12).

The number of filtering results of the equational buffer a ccounts for 94.6% of that of the traditonal buffer. After a further analysis, we find that all the positioning points screened out by the equational buffer are included in the filtering result of the traditional buffer, without any point screened out mistakenly. By analyzing the spatial distribution of omitted points by the e quational buffer, we also find that almost all of omitted points

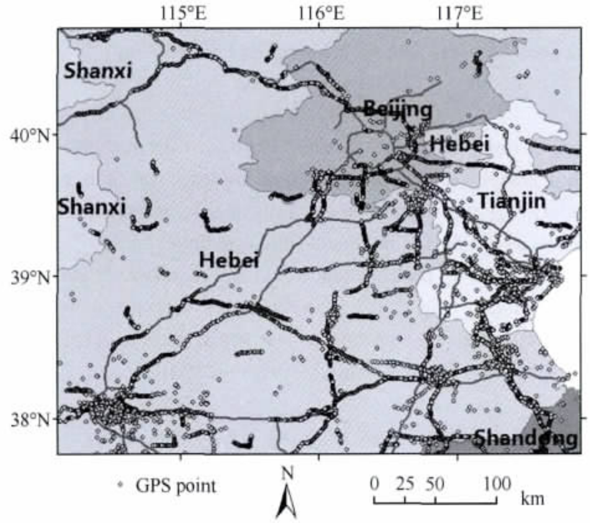


Fig.10 Spatial distribution of GPS positioning data locating within the study area

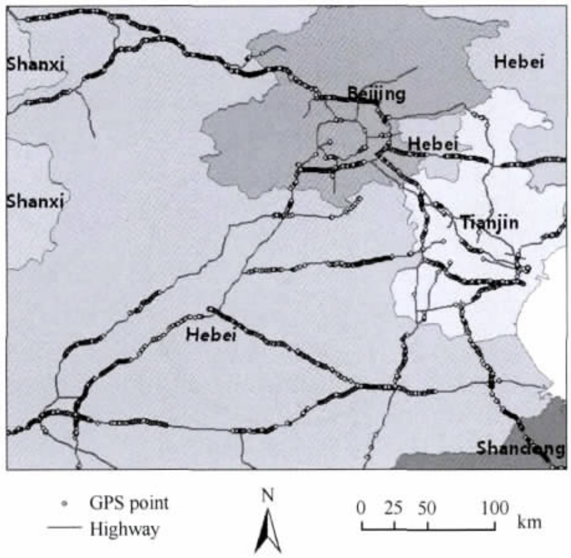


Fig.11 Filtering result of traditional buffer

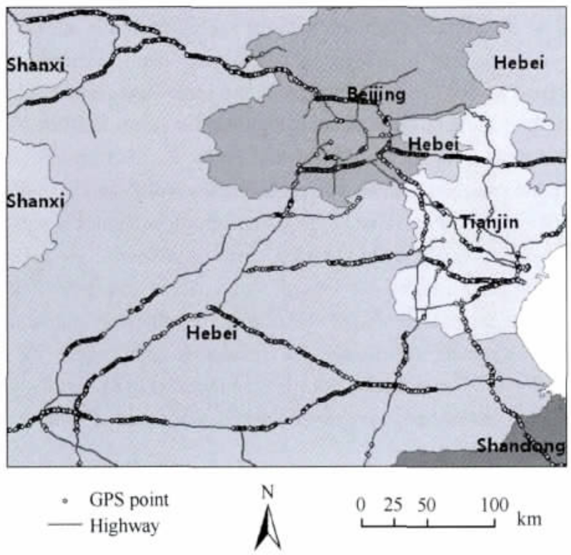
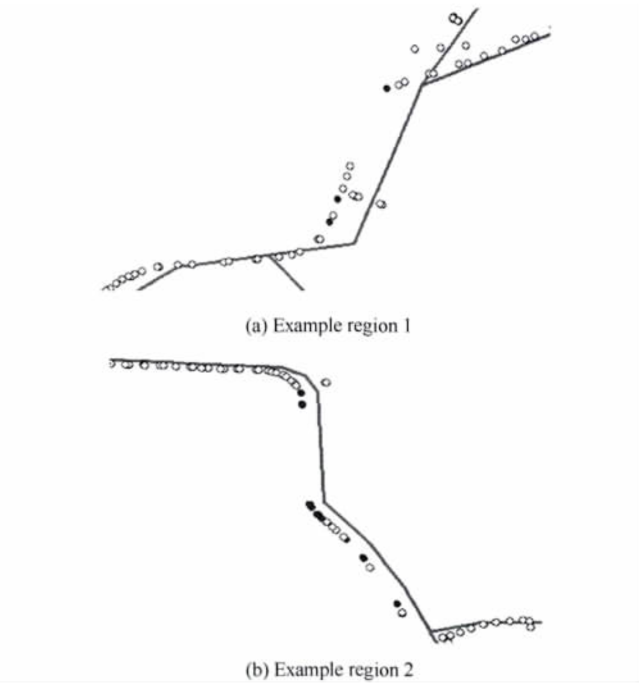


Fig.12 Filtering result of equational buffer

are far away from any highway. Fig. 13 gives two example regions, in which the hollow point is the common filtering result of two buffers, and the solid point is the point screened out by the traditonal buffer while omitted by the equational buffer. From the above analysis, we know that only less than 6% of positioning points are omitted by the equational buffer, and most of these omitted points are invalid data which hardly have influences on the subsequent computation of traffic information. More importantly, the adoption of equational buffer makes the speed of coarse filtering process increase nearly six times, improving the time efficiency of real-time traffic information estimating of highways.



(a) Example region 1

(b) Example region 2

Fig. 13 Spatial distribution of omitted positioning points by the equational buffer

(The hollow point is the common filtering result of two buffers, and the solid point is the point screened out by the traditonal buffer while omitted by the equational buffer)

5 CONCLUSION

The traditional buffer is represented by physical entities. This determines that the generation and application of buffer is a process composed of complicated geometric calculations, causing the time efficiency and space efficiency to be very low when buffer analysis is applied in processing massive geographic datasets. In this paper, we define an approximate representation form of a buffer by imitating the formula of gravitational isoline and call it equational buffer. Using equational buffers converts buffer analysis to algebraic operations on mathematical equations, which speeds up buffer analysis. The experiment result of data filtering of floating car technology has proven the effectiveness of equational buffers in buffer analysis without a very strict a ccuracy requirement.

Although the equational buffer obtains good results in dealing with a large dataset, there are still many aspects deserving further research. For example, the equational buffer is not applicable to polygon features at present while the polygon type is a widely used feature type. It is very necessary to analyze if the equational buffer could be applicable to polygon features. Moreover, an equational buffer is represented by a mathematical equation, making it easy to be transformed into three-dimensional form. And the time and space efficiency do not decrease greatly after the transformation. So it can serve as a promising solution to the challenges that the existing buffer faces when extended to three-dimension space.

Acknowledgements: Thanks to Antoine Lokongo for providing language helps on this article. Gratitude also goes to anonymous reviewers for their effort and useful comments on this article.

REFERENCES

Dong P, Mao D J, Li J and Yang C J. 2004. An effective buffer generation method in GIS. Computer Engineering and Applications, 40(16): 4-8

Draijer G, Kalfs N and Perdok J. 2000. Global positioning system as data collection method for travel research. Transportation Research Record, 1791: 147-153 [DOI: 10.3141/1719-19]

Ghosh P K. 1990. A solution of polygon containment, spatial planning, and other related problems using minkowski operations. Computer Vision Graphics and Image Processing, 49(1): 1-35 [DOI: 10.1016/0734-189X(90)90160-W]

Ghosh P K. 1991. An algebra of polygons through the notion of negative shapes. Computer Vision Graphics and Image Processing, 54(1): 119-144 [DOI: 10.1016/1049-9660(91)90078-4]

Guo R Z. 1997. Spatial Analysis 2nd edn. Wuhan: Wuhan Technical University of Surveying and Mapping Press: 255

Hu P, You L, Yang C Y and Wu Y L. 2006. Map Algebra 2nd edn. Wuhan: Wuhan Technical University of Surveying and Mapping Press: 297

Lu F and Cui W H. 1999. Matching error analysis of GPS positioning and digital maps in vehicle guidance and monitoring. Journal of Remote Sensing, 3(4): 312-317

Ma W W. 2004. Physics Science 4th edn. Beijing: Higher Education Press

Preparata F P and Shamos M I. 1985. Computational Geometry: An Introduction. New York: Springer

Sun J G. 1998. Computer Graphics 3rd edn. Beijing: Tsinghua University Press

Taloy G. 1994. Point in polygon test. Survey Review, 32(10): 479-484

Wang J C, Rui Y K and Li Y Q. 2007. A novel method for buffer generation based on vector boundary tracing. Computer Engineering and Applications, 43(33): 66-68

Wang J C, Cui C, Pu Y X, Ma J S and Chen G. 2010. A novel algorithm of buffer construction based on run-length encoding. Cartographic Journal, 47(3): 198-210 [DOI: 10.1179/000870410X12786821061413]

Wu H H. 1997. Problem of buffer zone construction in GIS. Journal of Wuhan Technical University of Surveying and Mapping, 22(4): 358-366

Xiang W N and Stratton W L. 1996. The *b*-function and variable stream buffer mapping: a note on "A GIS Method for Riparian Water Quality Buffer Generation". International Journal of Geographical Information Systems, 10(4): 499-510 [DOI: 10.1080/02693799608902092]

Yan H W, Yang W F, Cheng Q G and Liang T G. 2000. A fast algorithm of topological polygon auto-construction based on azimuth calculation. Journal of Image and Graphics, 5(7): 563-567

Žalik B and Clapworthy G J. 1999. A universal trapezoidation algorithm for planar polygons. Computers and Graphics, 23(3): 353-363 [DOI: 10.1016/S0097-8493(99)00044-8]

Žalik B, Zadravec M and Clapworthy G J. 2003. Construction of non-symmetric geometric buffer from a set of line segments. Computers and Geosciences, 29(1): 53-63 [DOI: 10.1016/S0098-3004(02)00076-6]

数学方程近似模拟缓冲区分析的加速方法

李军，秦其明，陈超，赵越，解超

北京大学 遥感与地理信息系统研究所，北京 100871

摘 要：缓冲区分析是地理信息系统空间分析的一种重要方法，但应用现有的缓冲区分析方法处理海量数据的时间效率和空间效率非常低。针对存在的问题，本文定义了缓冲区的一种近似表达方式——方程式缓冲区，它以数学方程模拟缓冲区的边界。使用方程式缓冲区代替传统的缓冲区能使缓冲区分析从一个包含复杂几何运算的过程转变成包含简单代数运算的过程，从而提高计算效率。论文阐述了方程式缓冲区的定义、图形、使用方法和性能。最后将方程式缓冲区应用于中国高速公路实时路况采集的数据筛选过程中，实验结果表明方程式缓冲区在海量地理数据分析中的有效性。

关键词：缓冲区分析，缓冲区表达，数据筛选，方程式缓冲区，海量地理数据

中图分类号：P208 文献标志码：A

引用格式：李军，秦其明，陈超，赵越，解超. 2013. 数学方程近似模拟缓冲区分析的加速方法. 遥感学报，17(5)：1131－1145

Li J，Qin Q M，Chen C，Zhao Y and Xie C. 2013. A method of approximately simulating buffers based on mathematical equations for accelerating buffer analysis. Journal of Remote Sensing，17(5)：1131－1145 [DOI：10.11834/jrs.20132166]

1 引 言

缓冲区分析是地理信息系统空间分析的一种重要的方法，它能辅助地理信息检索与综合处理，使用范围非常广，例如，自然灾害损失评估、区域经济分析、城市规划和环境监测等。缓冲区分析包括两部分：缓冲区生成和缓冲区应用。其中缓冲区生成是缓冲区分析的关键过程，它关系到缓冲区分析的效率和准确性，因此很多学者在这方面做了深入研究。

对于矢量缓冲区，核心生成算法是平行双线算法，是一个非常简单的过程，但当出现地理要素形状复杂或平行线自相交时，问题变得复杂。为了处理这些特殊情况，学者们做了大量研究工作。Ghosh(1990，1991)基于数学形态学的思想提出两个多边形求和的边界组合运算。毋河海(1997)介绍了适用于生成凸角圆弧缓冲区的数学模型：形成双线的几何算法模型和针对自相交问题的关系处理模型。Žalik等人(1999，2003)提出了一种使用扫描线方法构建线段集的几何轮廓的算法。董鹏等人(2004)运用旋转点变换公式和递归方法对凸角圆弧生成算法作了进一步改进，简化平行线的生成过程并较好地解决了缓冲区边线的自相交问题。王结臣等人(2007)设计了一种基于矢量追踪技术的缓冲区生成算法，避免一些复杂的矢量计算过程，通过追踪直接获得完整的闭合边界曲线。

对于栅格缓冲区，生成方法相对简单且易于实现。生成栅格缓冲区的常用方法有游程刷叠置算法(Taloy，1994)，数学形态学扩张法(胡鹏 等，2006)和填充算法(郭仁忠，1997)。Xiang 和 Stratton(1996)介绍了在栅格环境下的一种可变化缓冲区生成方法，通过使用 B 函数来实现根据物理和生态条件的差异性绘制具有可变宽度的河流缓冲区。Wang 等人(2010)介绍了基于栅格距离求解缓冲区的实现方法和算法优化策略。

虽然他们在缓冲区生成的稳健性和效率方面取得了不小的进步，但缓冲区分析仍然存在以下问题：一、矢量缓冲区生成算法所涉及的几何计算较为复

收稿日期：2012-05-21；修订日期：2012-11-18；优先数字出版日期：2012-11-25

基金项目：国家高技术研究发展计划(863 计划)(编号：2012AA121305)；高分辨率对地观测系统专项项目

第一作者简介：李军(1987—)，男，博士研究生，主要从事地理数据挖掘和交通地理信息系统研究。E-mail：newton2069@126.com

通信作者简介：秦其明(1955—)，男，教授，主要研究方向为定量遥感，地理信息系统建模与应用。E-mail：qmqin@pku.edu.cn

杂,学者们在提高算法稳健性的同时,也不可避免地提高了算法的复杂度,导致算法具有实现难度高、执行效率低的缺点; 二、矢量缓冲区的表达方式使得在进行缓冲区分析时,要频繁执行缓冲区多边形与要素的包含关系判定,导致分析速度较慢; 三、栅格缓冲区使用一组带有特定属性值的栅格单元记录缓冲区的位置,因此它的表达精度受栅格单元的大小限制,精度不高,并且缓冲区的存储需要耗费大量内存。

在以往的缓冲区分析中,这些问题影响不大。但随着地理信息科学、计算机、通信技术的发展,地理信息技术应用到越来越多的行业中,经常会面临海量地理信息数据的处理,此时传统缓冲区分析方法所存在的问题严重制约着缓冲区分析的应用效果。以往研究主要集中在缓冲区的生成方法上,希望通过改进缓冲区生成方法来提高缓冲区分析效率。本文认为寻找更简单的缓冲区表达方式也是提高缓冲区分析效率的一种有效途径。

通过分析目前的缓冲区分析过程,并受到解析几何中"一个方程表示一个图像,方程是图像的数学表达形式"观点的启发,本文提出了通过用数学方程表示的图形来模拟缓冲区边界的思想,并仿照万有引力等值线公式的形式定义了缓冲区的一种近似表达方式——方程式缓冲区。方程式缓冲区的使用使缓冲区分析从复杂的几何运算过程转变为简单的代数运算过程,提高了缓冲区分析的计算效率。

2 方程式缓冲区

2.1 对传统缓冲区分析过程的思考

由于栅格缓冲区精度低且易受内存限制,因此在处理海量地理数据时人们一般使用矢量缓冲区,如无特别说明,下文中只讨论矢量缓冲区。正如前文介绍的,现有的缓冲区分析方法需要进行多边形的空间包含关系判断,过程较复杂。如何提高缓冲区分析效率呢? 我们从传统的缓冲区分析过程中寻找答案,例如一项任务是分析有多少居民区在一个医院的服务范围内(图1),其中点要素 A 表示医院,点要素集合 B 中的每一个点都代表一个居民区,假设医院的服务距离是 3 km。传统的分析过程大致是: 第 1 步生成点 A 的缓冲区,无论使用何种缓冲区生成方法,都要生成一系列点集 $\boldsymbol{P}_A = \{P_1, P_2, \cdots, P_n\}$ 拟合以点 A 为圆心半径为 3 km 的圆 O_A,这个点集 $\boldsymbol{P}_A$ 就是点 A 的缓冲区,本质上也是一个矢量多边形。第 2 步,判断点集 B 中的哪些点在缓冲区 $\boldsymbol{P}_A$ 内,这需要计算点集 B 中每一个点与矢量多边形 $\boldsymbol{P}_A$ 的包含关系。整个分析过程包括了多个复杂的几何计算。事实上,最终目的是想知道点集 B 中哪些点在圆 O_A 内,在数学角度就等效于点(x,y)是否满足圆 O_A 的方程 $\sqrt{(x-x_0)^2+(y-y_0)^2} \leqslant 3$,其中$(x_0, y_0)$为点 A 的坐标。这样一个简单的代数计算,在 GIS 中却需要多个复杂的几何运算来完成。

设想如果能用数学方程来拟合地理要素的缓冲区,则缓冲区可以用数学方程来表达,缓冲区的生成与应用就可以通过简单的代数运算完成,避免了复杂的几何运算,那么现有缓冲区分析面临的问题就迎刃而解了。这种缓冲区与现有缓冲区最本质的区别是它不使用实体要素——离散的点表达,而是用数学方程表达,因此称之为方程式缓冲区。

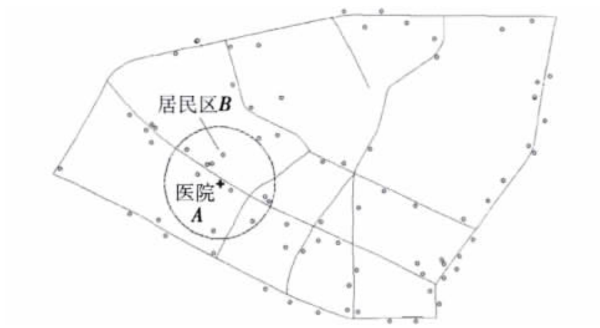


图1 传统缓冲区分析

2.2 方程式缓冲区的定义

构建方程式缓冲区的关键是寻找能较好地拟合传统缓冲区边界的数学方程,下面分别介绍对应 GIS 中不同类型要素的方程式缓冲区的公式。GIS 中要素类型分为 3 种: 点要素、线要素和面要素。面要素也可以用线要素来表达,区别仅在于面要素是闭合的,而线要素通常是不闭合的。线要素的基本组成单元是直线段,由此可知 GIS 中最重要的是点和线段,为此分别以点和线段为例着重介绍方程式缓冲区的定义。

缓冲区分析主要是以距离为依据,因此方程式缓冲区需要与距离有关系,并且具有连续封闭的特点,当然最重要的特点是能用一个数学公式描述。而万有引力的等值线恰好能满足这些要求,为此借助万有引力等值线公式的形式构造方程式缓冲区,用地理要素的一条引力等值线拟合与之最接近的缓冲区边界。下面分别推导点和线段的方程式缓冲区的数学公式。

(1)点的缓冲区

假设点 A 的质量为 m,位于(x_0,y_0),另有一点 B 与它相距 d 质量为 m',点 C 与它相距 r(图2)。

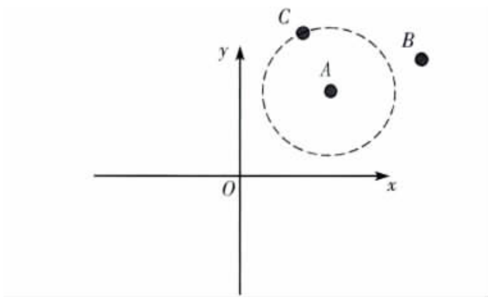


图2 点要素方程式缓冲区的推导

点 A 与点 B 间的引力大小(马文蔚,2004)为 $F=G\frac{mm'}{d^2}$。点 C 处的引力为 $G\frac{mm'}{r^2}$,则点 C 处的引力等值线公式为 $F=G\frac{mm'}{r^2}$,其中 r 为常数。

为了计算方便,等值线公式的两边同时除以 Gmm',定义点的缓冲区为 $Buffer_{\text{point}}$: $\frac{1}{d^2}\geqslant\frac{1}{r^2}$,即

$$Buffer_{\text{point}}: \sqrt{(x-x_0)^2+(y-y_0)^2}\leqslant r^2 \quad (1)$$

式中,r 是缓冲区半径。

(2)线段的缓冲区

为了计算方便,定义一个临时坐标系 $x'o'y'$,使线段位于 x'轴上且中点处于原点 o'处,假设线段 AB 质量为 m,长度为 l,另有一质量为 m'位于(x',y')处的点 C(图3)。

计算可得线段的线密度为 $\rho=m/l$,将线段 AB 划微分成无数小段 $\mathrm{d}x_l$,位于$(x_l,0)$处的小段 $\mathrm{d}x_l$ 对点 C 在 x'和 y'方向的引力大小分别为:

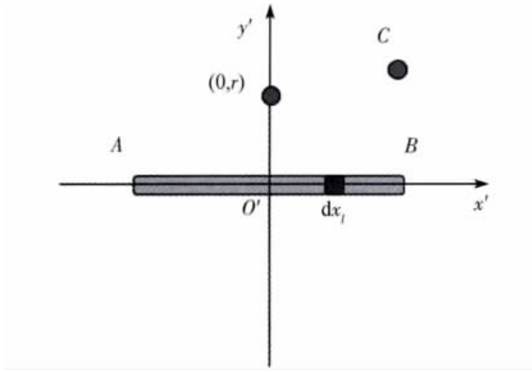


图3 线要素方程式缓冲区的推导

$$\mathrm{d}F_{x'}=\frac{G\rho m'(x'-x_l)\mathrm{d}x_l}{((x'-x_l)^2+y'^2)^{3/2}}$$

$$\mathrm{d}F_{y'}=\frac{G\rho m'y'\mathrm{d}x_l}{((x'-x_l)^2+y'^2)^{3/2}}$$

线段 AB 对点 C 在 x'和 y'方向引力大小分别为

$$F_{x'}=\int_{-\frac{l}{2}}^{\frac{l}{2}}\mathrm{d}F_{x'}=$$

$$G\rho m'\left(\frac{1}{\sqrt{\left(x'+\frac{l}{2}\right)^2+y'^2}}-\frac{1}{\sqrt{\left(x'-\frac{l}{2}\right)^2+y'^2}}\right)$$

$$F_{y'}=\int_{-\frac{l}{2}}^{\frac{l}{2}}\mathrm{d}F_{y'}=$$

$$\frac{G\rho m'}{y'}\left(\frac{x'-l/2}{\sqrt{\left(x'-\frac{l}{2}\right)^2+y'^2}}-\frac{x'+l/2}{\sqrt{\left(x'+\frac{l}{2}\right)^2+y'^2}}\right),$$

$$(y'\neq 0)$$

线段 AB 对点 C 的引力总大小为

$F_G(x',y')=\sqrt{F_{x'}^2+F_{y'}^2}$,即

$$F_G(x',y')=$$

$$\begin{cases} G\rho m'\sqrt{\left(\frac{1}{\sqrt{(x'+l/2)^2+y'^2}}-\frac{1}{\sqrt{(x'-l/2)^2+y'^2}}\right)^2+\left(\frac{x'-l/2}{y'\sqrt{(x'-l/2)^2+y'^2}}-\frac{x'+l/2}{y'\sqrt{(x'+l/2)^2+y'^2}}\right)^2}, & (y'\neq 0)\\ G\rho m'\left|\frac{1}{|x'+l/2|}-\frac{1}{|x'-l/2|}\right|, & (y'=0)\end{cases}$$

在$(0,r)$处的引力大小为 $F_G(0,r)$,则此处的引力等值线为 $F_G=F_G(0,r)$。同样为了计算方便,令 $F'(x',y')=\frac{F_G(x',y')}{G\rho m'}$,则线段在临时坐标系 $x'o'y'$中的缓冲区为 $F'(x',y')\geqslant\frac{F_G(0,r)}{G\rho m'}=\frac{l}{r\sqrt{\frac{l^2}{4}+r^2}}$

式中,r 是缓冲区半径。

上述计算是在临时坐标系 $x'o'y'$中完成。然后将计算所得的缓冲区方程从临时坐标系 $x'o'y'$中转换到地理坐标系 xoy 中,坐标转换公式如下:

$$x'=TranX(x)=(x-x_m)\cos t+(y-y_m)\sin t$$

$$y'=TranY(y)=(x_m-x)\sin t+(y-y_m)\cos t$$

式中,$TranX$、$TranY$ 为坐标转换函数,(x_m,y_m)为线段 AB 的中点,t 为 AB 线段$((x_1,y_1),(x_2,y_2))$与 X 轴的夹角。

令 $F(x,y) = F'(x',y') = F'(TranX(x), TranY(y))$,则定义线段的缓冲区为

$$Buffer_{\text{line}}: F(x,y) \geqslant \frac{l}{r\sqrt{\frac{l^2}{4}+r^2}} \quad (2)$$

点和线段的缓冲区分别是式(1)和式(2)所表示的闭合曲线,曲线的缓冲区是由式(2)计算出的各子线段缓冲区的并集,多要素的缓冲区则是多要素中所有单要素的缓冲区的并集。本文中暂不讨论面缓冲区。

2.3 方程式缓冲区的图形

方程式缓冲区以数学方程表达,使用它进行分析时一般不需要将其显示出来,为了使读者对方程式缓冲区有更直观的认识,下面分别以点和线段为例,显示具有不同缓冲区半径的方程式缓冲区。图4为点要素的不同大小的方程式缓冲区图形,从里到外缓冲区半径分别为 20 m,40 m,60 m,80 m。图 5 为长度 400 m 的线段的不同大小的方程式缓冲区图形,从里到外缓冲区半径分别为 40 m,80 m,1 20 m,160 m。

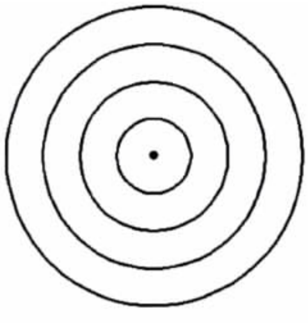


图 4 点的方程式缓冲区

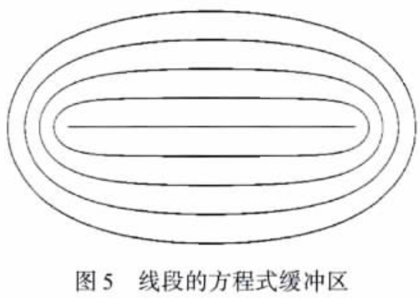


图 5 线段的方程式缓冲区

2.4 方程式缓冲区的应用方法

以地理信息检索为例,介绍方程式缓冲区的应用方法,即查询在缓冲区内的地理要素。

(1) 点缓冲区 判断某要素点 $T(x_t,y_t)$ 是否在点 $A(x_0,y_0)$ 半径为 r 的缓冲区内时,将要素坐标代入式(1)中得到 $\sqrt{(x_t-x_0)^2+(y_t-y_0)^2} \leqslant r^2$,若不等式成立,则要素点 T 在点 A 的缓冲区内;若不成立,则要素点 T 在点 A 的缓冲区外。缓冲区半径变化时只需要改变 r 的值重新计算即可。

(2) 线段缓冲区 判断某要素点 $T(x_t,y_t)$ 是否在线段 AB 半径为 r 的缓冲区内,A、B 的坐标分别为 (x_1,y_1),(x_2,y_2),将要素坐标代入式(2)中得到 $F(x_t,y_t) \geqslant \frac{l}{r\sqrt{\frac{l^2}{4}+r^2}}$,若不等式成立,则要素点 T 在线段 AB 的缓冲区内;若不成立,则要素点 T 在线段 AB 的缓冲区外。

(3) 曲线缓冲区 曲线 $Q(P_1,P_2,\cdots,P_n)$,判断某要素点 $T(x_t,y_t)$ 是否在曲线半径为 r 的缓冲区内时,需要将要素坐标代入曲线所有子线段(P_tP_{t+1},$t=1,2,\cdots,n-1$))的缓冲区不等式中进行判断,只要有一个不等式成立,则要素点 T 在曲线 Q 的缓冲区内,反之则不在曲线 Q 的缓冲区内。

3 方程式缓冲区的性能

3.1 应用效率

对比传统缓冲区和方程式缓冲区的应用效率,应用效率包括:空间效率和时间效率。空间效率即进行缓冲区分析时所耗费的存储空间;时间效率即使用缓冲区时所涉及的计算过程耗费的时间,又分为生成时间效率和使用时间效率。

下面分别对比方程式缓冲区和传统缓冲区的 3 种效率。

(1) 缓冲区空间效率:传统缓冲区是以矢量多边形存在,多边形的节点坐标需要存储在物理内存中,耗费空间的大小由要素的数量、尺寸和形状决定。一般的缓冲区应用对存储空间要求不高,但对海量数据进行处理时,需要耗费大量存储空间,导致传统缓冲区的空间效率非常低。而方程式缓冲区是由数学方程表达,不同要素的缓冲区都由一个共同的方程表达,只是系数不同,因此只需要在程序中保存共同的数学方程即可,耗费内存不超过 1 kB。

(2) 生成时间效率:传统缓冲区的生成有多种算法:凸角圆弧法和角平分线法,各个算法都需要耗费一定时间。在处理对象数量较少时不存在问题,但对海量数据进行缓冲区分析时,生成所有要素的缓冲区需要耗费大量时间。特别是要素位置或者缓

冲区半径频繁变化时,耗费时间成倍地增加,给应用带来诸多不便。而任意要素的方程式缓冲区都只是一个数学方程,而且方程形式是固定不变的,因此缓冲区的生成就只是方程参数的选择,时间效率极高。

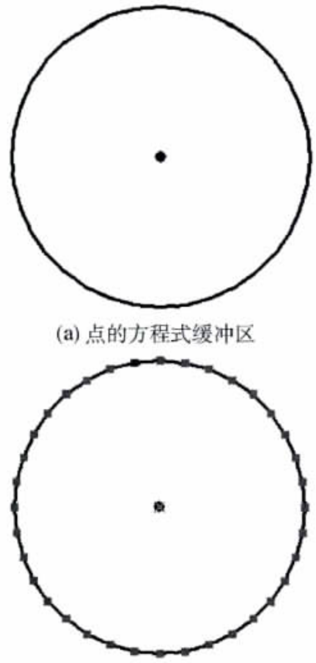
(3)使用时间效率:对传统缓冲区来说,判断任意一个要素是否包含在一个缓冲区内实质上就是点与多边形的包含关系判断,常用方法有射线法(Preparata和Shamos,1985;孙家广 等,1998)、方位角求夹角和法(闫浩文 等,2000)等,无论哪一种都是一个复杂的几何运算。与传统缓冲区不同,方程式缓冲区是一个数学表达式,判断某要素是否在缓冲区内即判断该要素的坐标是否都能使缓冲区方程成立,这是一个相对简单的过程,耗时较少。尤其是处理对象为点类型要素时,方程式缓冲区的使用效率比传统缓冲区高很多。

3.2 拟合精度分析

方程式缓冲区是通过连续光滑的图形拟合传统缓冲区的边界,因此有可能存在拟合误差,本节中将分析方程式缓冲区的拟合精度。在GIS应用中,缓冲区生成方法最常用的是凸角圆弧法,以凸角圆弧法所定义的缓冲区作为真实值,分析方程式缓冲区的拟合精度。下面分别讨论点要素和线要素的方程式缓冲区的拟合精度。

真实的点缓冲区是圆形区域。尽管真实缓冲区的边界是连续的,但在矢量环境中,为了使计算机能对它进行存储与分析,必须采用离散化表达方式。因此地理要素的传统缓冲区是使用一组带有坐标的离散点来表示。对点要素而言,使用一个离散点集拟合它的圆形缓冲区。这种表达方式导致传统缓冲区存在一定的误差。然而,方程式缓冲区是由圆的方程来表达,它准确地拟合了真实缓冲区(如图6所示)。因此,对于点要素,方程式缓冲区不存在误差,拟合精度优于传统缓冲区方法。

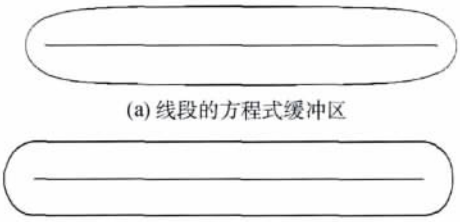
线段的真实缓冲区在侧面是一条与线段平行的直线,端点处是半圆。方程式缓冲区与凸角圆弧缓冲区有所不同,在侧面中点处与凸角圆弧缓冲区重合,其他地方稍小。图7所示的是长度为200 m的线段半径为20 m的两种缓冲区图形。下面分析方程式缓冲区的拟合误差分布,方程式缓冲区的两端与真实值相差较大,但由于所占面积较小,一般情况下影响不大,这里不作分析。又因缓冲区在沿线段方向和沿线段垂直方向上是轴对称的,因此本文中只对缓冲区一侧的区域进行拟合误差分析,如图8中所示的横坐标0—100 m的范围。



(a)点的方程式缓冲区

(b)凸角圆弧缓冲区

图6 点的方程式缓冲区和凸角圆弧缓冲区



(a)线段的方程式缓冲区

(b)凸角圆弧缓冲区

图7 线段的方程式缓冲区和凸角圆弧缓冲区

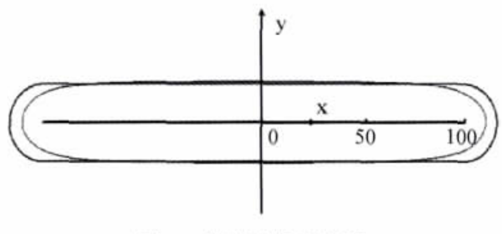


图8 精度分析示意图

定义方程式缓冲区横坐标为x处的拟合误差为$\varepsilon_x = \frac{y_1 - y_2}{y_1}$,其中$y_1$为凸角圆弧缓冲区在$x$处的纵坐标,$y_2$为方程式缓冲区在$x$处的纵坐标。计算缓冲半径分别为1 m、10 m、30 m、100 m和400 m时的拟合误差大小,结果如图9所示。

结果显示,缓冲区半径为1 m时,99%的区域拟合误差小于7%;缓冲区半径为10 m时,93%的区

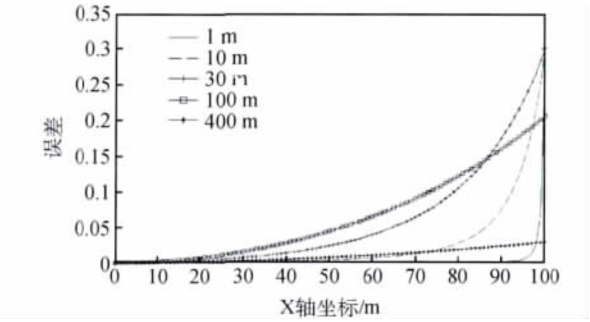


图9　拟合误差计算结果

域拟合误差小于10%；缓冲区半径为30 m时，80%的区域拟合误差小于10%；缓冲区半径为100 m时，99%的区域拟合误差小于20%；缓冲区半径为400 m时，100%的区域拟合误差小于3%。这些数据表明，当缓冲区半径与线段长度相差较大时，方程式缓冲区的拟合误差都很小，绝大部分区域几乎与凸角圆弧缓冲区吻合。当缓冲区半径与线段长度接近时，拟合误差有所增大，但接近80%左右的区域拟合误差都小于10%，除了非常少一部分，所有范围的拟合误差都控制在20%以内。

虽然方程式缓冲区存在拟合误差，但在精度要求不是非常严格的缓冲区分析应用中这些误差是可以接受的。以地理信息检索应用为例说明，方程式缓冲区相对凸角圆弧缓冲区的面积较小，因此拟合误差的存在会导致漏掉部分数据。当有海量数据需要进行缓冲区分析时，用方程式缓冲区代替传统缓冲区进行分析仅仅会损失少量数据，对结果影响不大，但带来的好处是计算效率得到了非常大的提高。

4　应用实例

为了更好地展示方程式缓冲区在处理大量地理数据的优势，本节将方程式缓冲区应用于基于浮动车的实时交通信息采集中（Draijer 等，2000）。目前正在参与一项有关浮动车的研究，接收来自全中国装有车载全球定位系统GPS设备的重点营运车辆上的实时定位数据，以此估算中国高速公路的实时路况信息。该项目所涉及的数据量庞大，非常适合验证方程式缓冲区在处理海量地理数据的有效性。研究中使用的数据源是车辆的GPS定位数据和数字地图中的道路数据，由于二者都不可避免地存在测量误差，因此路况计算的第1步就是地图匹配（陆锋和崔伟宏，1999）——将GPS定位点与道路图的路段关联起来或确定每个GPS定位点所在的路段。现阶段本项目目的是估算高速公路的路况信息，而车辆行驶位置可能在各种等级的道路（高速公路、国道、省道和其他道路）、停车场等，这样导致所接收到的部分GPS定位点不是在高速公路上采集的，对估算高速路路况信息没有作用。接收到的GPS点数据量庞大，如果对每一个点都进行地图匹配会需要较长时间，为了提高计算效率，我们对原始的GPS点进行粗略筛选，将无效数据即与高速公路相距较远的定位点剔除掉，然后对剩下的有较大可能性在高速路上的点进行地图匹配。通常GPS点的粗略筛选由高速公路的缓冲区来完成，此节中分别利用传统缓冲区和方程式缓冲区来实现筛选过程，对比二者的结果，下列实验都在具有2.8 GHz双核CPU和2 GB内存的计算机上完成。

本文选取中国华北地区的部分区域为实验区，地理范围为37°44′41″N—40°44′47″N和114°7′10″E—117°53′1″E。若实时路况信息的更新频率为半小时一次，即通过处理半小时内所接收的GPS定位数据估算高速公路各路段的路况。选取2010年9月19日01:30—02:00期间内某个数据交换中心所接收到的全部GPS定位数据为研究对象，该数据集共有194205个GPS定位点，覆盖到实验区内的GPS定位数据空间分布如图10所示。

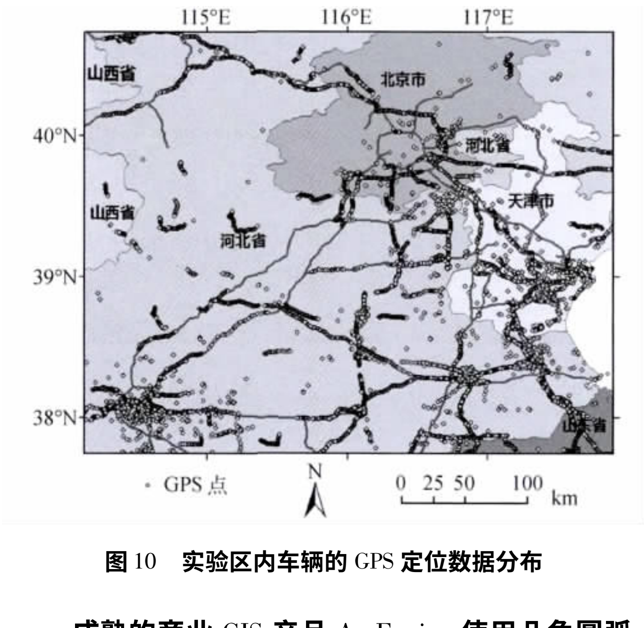


图10　实验区内车辆的GPS定位数据分布

成熟的商业GIS产品ArcEngine使用凸角圆弧法生成缓冲区，本文以它作为传统缓冲区的代表实现数据筛选。设置缓冲区半径为500 m，先生成所有路段的缓冲区，然后剔除掉落在缓冲区以外的GPS定位点，不计缓冲区的生成时间，整个筛选过程

耗时415.61 s。通过筛选得到11171个有效的GPS定位点,结果如图11所示。然后使用方程式缓冲区实现同样的功能,输入缓冲区半径参数为500 m,然后检索落在方程式缓冲区内的GPS定位点,整个筛选过程耗时76.27 s,通过筛选得到10564个有效的GPS定位点,结果如图12所示。

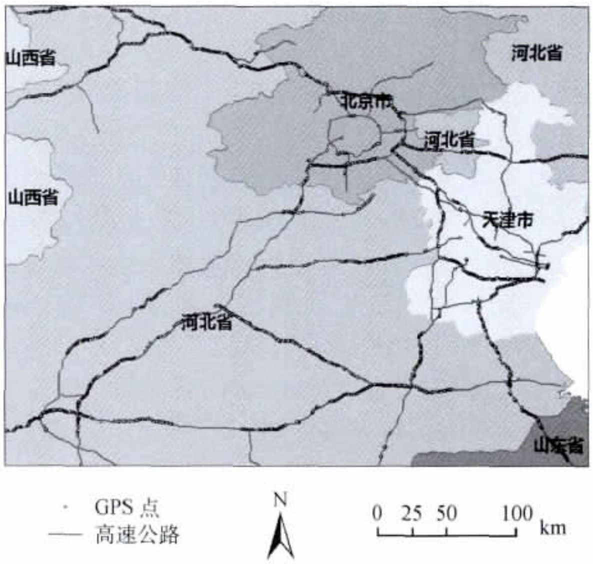


图11 凸角圆弧缓冲区的数据筛选结果

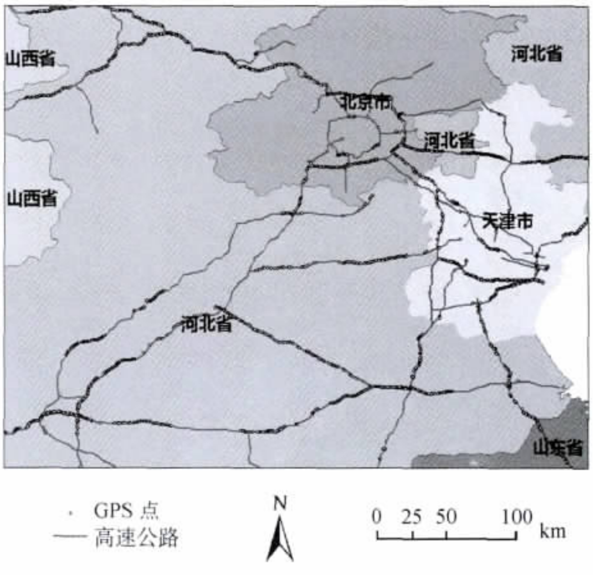


图12 方程式缓冲区的数据筛选结果

方程式缓冲区的筛选结果的数量占凸角圆弧缓冲区的94.6%,经过进一步分析,方程式缓冲区筛选出的全部定位点都包括在凸角圆弧缓冲区的筛选结果中,没有筛选出任何错误点。通过分析方程式缓冲区漏选数据的空间分布,还发现绝大部分漏掉的数据都距离道路较远。图13给出了两个示例区域,其中空心点为两种缓冲区共同的筛选结果,实心点为凸角圆弧缓冲区筛选出方程式缓冲区没有筛选出的点。由以上分析可知,方程式缓冲区仅漏选了不到6%的定位点,而且这些点中的绝大部分属于无效数据,对后面的路况计算几乎没有影响。更重要的是方程式缓冲区的应用使粗略筛选过程的速度提高了近6倍,进而提高了道路实时路况计算的时间效率。

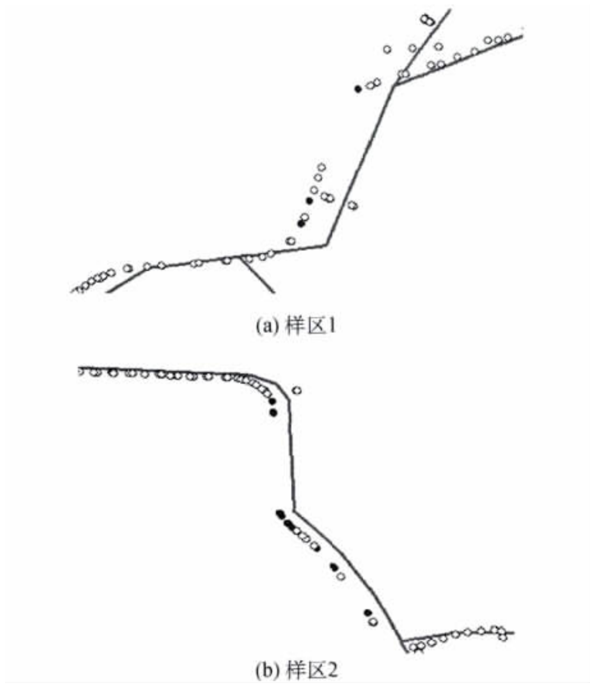


图13 方程式缓冲区漏选点的空间分布
(空心点为两种缓冲区共同的筛选结果,实心点为方程式缓冲区的漏选点)

5 结论

传统的缓冲区是以物理实体来表达,决定了缓冲区的生成与应用都不能避免复杂的几何运算,导致对海量地理数据进行缓冲区分析时,时间效率和空间效率都很低。本文仿照万有引力等值线的公式形式定义了缓冲区的一种近似表达方式,并称之为方程式缓冲区。在进行缓冲区分析时,使用方程式缓冲区代替传统缓冲区,从而使缓冲区分析变成数学方程的运算,进而提高缓冲区分析速度。实验结果证明了方程式缓冲区在精度要求不是非常严格的缓冲区分析应用的有效性。

虽然方程式缓冲区在处理大数据集时取得了较好的效果,但仍然有许多方面需要进一步研究。例如,目前方程式缓冲区还不适用于面状矢量要素的分析,而面状要素也是GIS中的一种常用数据类型,

研究适用于面状矢量要素的方程式缓冲区显得非常有必要。此外,由于方程式缓冲区是以数学方程表达,使得它很容易变换成3维形式,并且在变换后的时间和空间效率都没有明显的降低,因此它非常有希望成为解决目前缓冲区难以扩展到3维空间的困难的一种途径。

参考文献(References)

董鹏,毛东军,李军,杨崇俊. 2004. 一种有效的GIS缓冲区生成算法. 计算机工程与应用,40(16):4-8

Draijer G, Kalfs N and Perdok J. 2000. Global positioning system as data collection method for travel research. Transportation Research Record, 1791: 147-153 [DOI: 10.3141/1719-19]

Ghosh P K. 1990. A solution of polygon containment, spatial planning, and other related problems using minkowski operations. Computer Vision Graphics and Image Processing, 49(1): 1-35 [DOI: 10.1016/0734-189X(90)90160-W]

Ghosh P K. 1991. An algebra of polygons through the notion of negative shapes. Computer Vision Graphics and Image Processing, 54(1): 119-144 [DOI: 10.1016/1049-9660(91)90078-4]

郭仁忠. 1997. 空间分析 第二版. 武汉: 武汉测绘科技大学出版社: 255

胡鹏,游涟,杨传勇,吴艳兰. 2006. 地图代数 第二版. 武汉: 武汉大学出版社: 297

陆锋,崔宏伟. 1999. 车辆导航与监控中GPS/GIS实时定位配准误差分析. 遥感学报,3(4):312-317

马文蔚. 2004. 物理学 第四版. 高等教育出版社

Preparata F P and Shamos M I. 1985. Computational Geometry: An Introduction. New York: Springer

孙家广. 1998. 计算机图形学 第三版. 北京: 清华大学出版社

Taloy G. 1994. Point in polygon test. Survey Review, 32(10): 479-484

王结臣,芮一康,李永全. 2007. 一种基于矢量边界追踪的缓冲区生成方法. 计算机工程与应用,43(33):66-68

Wang J C, Cui C, Pu Y X, Ma J S and Chen G. 2010. A novel algorithm of buffer construction based on run-length encoding. Cartographic Journal, 47(3): 198-210 [DOI: 10.1179/000870410X12786821061413]

毋河海. 1997. 关于GIS缓冲区的建立问题. 武汉测绘科技大学学报,22(4):358-366

Xiang W N and Stratton W L. 1996. The *b*-function and variable stream buffer mapping: a note on "A GIS Method for Riparian Water Quality Buffer Generation". International Journal of Geographical Information Systems, 10(4): 499-510 [DOI: 10.1080/02693799608902092]

闫浩文,杨维芳,陈全功,梁天刚. 2000. 基于方位角计算的拓扑多边形自动构建快速算法. 中国图象图形学报,5(7):563-567

Žalik B and Clapworthy G J. 1999. A universal trapezoidation algorithm for planar polygons. Computers and Graphics, 23(3): 353-363 [DOI: 10.1016/S0097-8493(99)00044-8]

Žalik B, Zadravec M and Clapworthy G J. 2003. Construction of non-symmetric geometric buffer from a set of line segments. Computers and Geosciences, 29(1): 53-63 [DOI: 10.1016/S0098-3004(02)00076-6]