Novel approaches to the multiscalar display of vector data

NIU Fangqu* and CHENG Changxiub

Key Laboratory of Regional Sustainable Development Modeling, Institute of Geographical Sciences and Natural Resources Research, Chinese Academy of Sciences, 11A Datun Road, Chaoyang District, Beijing 100101, China; bLREIS, Institute of Geographical Sciences and Natural Resources Research, Chinese Academy of Sciences, 11A Datun Road, Chaoyang District, Beijing 100101, China

(Received 10 September 2011; final version received 11 December 2011)

There are many challenges to achieving quality multiscalar displays of vector data in geographical information system (GIS). Acknowledging and making use of the differences between traditional cartographic generalization and multiscalar display in GIS, this paper focuses on novel approaches to the definition of small objects and the establishment of scale sequence. The ease of implementation and efficacy of the solutions proposed are exemplified and analyzed. Possibilities for further research into the multiscalar display of vector data in GIS are thereby suggested.

Keywords: data generalization; multiscale; GIS; generalization rectangle; scale sequence

1. Introduction

One of the goals of modern information society is to provide flexible access to information for anyone, anywhere, anytime (1). With the development of network and multimedia visualization technologies, spatial data no longer need be envisioned statically at a single resolution. Geovisualization is increasingly multiangle, multipoint, and multilevel, consistent with calls to focus on human demands, individuation, and self-adaptation (2).

As such, a geographical information system (GIS) should be able to adaptively provide different data for different users, especially within large vector data-sets. More details and features should be displayed when the user zooms in. To achieve that requirement, questions of multiple scales and multiple representations have become important topics of theoretical and applied GIS research (2, 3). Various techniques have been proposed to realize the multiscalar display of GIS data (4–15).

The ideal multirepresentation of vector data is that of a scaleless database. With a scaleless database, a set of vector data at a larger scale is stored, and then a representation at any smaller scale could be obtained in real time when required. But different scales correspond to different rules depending on application requirements. Because of intelligence requirements, relatively few schemes have implemented scaleless databases. On the other hand, given that the zoom operations of users are discrete, display scales are often themselves discrete, so a database of discrete scales can be used to simulate a scaleless display, becoming the commonly used method for realizing a multiscalar display.

Several methods have been favored for implementation of multiscalar databases: the prestorage of multiscalar snapshots, real-time derivation of data at a given smaller scale from stored data, and storage of multiscalar indexes of increment. One of the drawbacks to the first method is that of redundant data; a key drawback to the second method is long processing time; while the third requires storage of the new objects created at a smaller scale, building an index of all the data, such as in an extended generalized area partitioning tree (GAP)-tree (7).

Results from previous studies into the multiscalar display of vector data may lead to the conclusion that most research has focused on the framework, strategy, storage, and evaluation methods for multiscalar displays (16), with some other research focusing on data structures and indices (17). In fact, the key to realize multiscalar display of vector data is to decide which objects are to be displayed at a certain scale, so, there are some problems that need further research. There are significant gaps in the research such as the definition of small objects, the establishment of the relative importance of objects, and the establishment of scale sequence. The resolution, or lack thereof, of such problems affects the quality of multiscalar displays for vector data. For multiscalar display, parameter choices must consider the particularities of the screen. Small objects are commonly deemed to be objects that need to be generalized at smaller scales. Since many definitions of small objects only consider area while ignoring shape, we offer a novel approach to defining small objects by considering both...
shape and area. However, the decision as to whether or not to generalize a small object should make reference to notions of importance defined according to the requirements of the application domain.

In implementing our method for the multiscalar display of vector data, we propose the concept of generalized scale (GS), an additive attribute of objects. The GS of an object is the scale at or below which the object should be generalized. Treating GS as an attribute will help to determine whether to generalize an object or not within a multiscalar display. To some extent, GS embodies the importance of objects. To implement multiscalar display, a few scales must be sampled. Cheng et al. gave a quantitative scale-setting approach, which focused on the geometry level (18). As to the object level, three methods will be discussed to build the scale sequence. A prototype system is also presented, based on solutions proposed in this paper.

The paper is organized as follows; Section 2 gives the criterion for small objects, exploring several concepts associated with the definition of small objects, including pixel area and generalization rectangle (GR). Section 3 proposes the GS attribute for objects, and discusses its usefulness in implementation of multiscalar vector visualization. Section 4 discusses how to select the scale to be displayed when users zoom in or out and describes methods for the establishment of the scale sequence. Section 5 presents our experiments and statistical results based on the solutions mentioned above. The final section draws conclusions and discusses prospects for future work.

2. Criterion of small objects

Traditionally, objects that are too small to be displayed are called small objects. Small objects will be generalized so to not explicitly exist in maps at smaller scales. In cartographic generalization, thresholds are often used. A threshold is the smallest value for the area or the length of an object before the object is considered to be “small”. For multiscalar visualization within a GIS, the settings for such a parameter should be influenced by the screen used for display.

2.1. Pixel area

Pixel area is the screen area measured by the pixel. For example, an object’s pixel area is the area occupied by the object on the screen when displayed. It is not suitable to use geographical area as the only criterion to determine the “smallness” of polygons in the multiscalar display of vector data. First, of course, multiscalar display requires more than one scale. At different scales, the permitted smallest value of polygons will be different. Second, as the map is displayed, it is not just the geographic area but also the on-screen impact that should be considered. In other words, small polygons should be those that would be too small for the eye to find them meaningful on the screen. Here, for the purposes of multiscalar display, small objects are defined according to their pixel area occupied on the screen. Thus, whatever the geographical area of a polygon and the display scale, if the pixel area value of a polygon is smaller than a specified value (called the “threshold”), the polygon is considered to be a “small polygon” that should be generalized.

2.2. Generalization rectangle

It is still insufficient, however, to take only the pixel area of a polygon as the criterion defining a small object, shape remains another important facet. Polygons of different shapes have different display effects on eyes. Consider, for example, long and narrow polygons. Such polygons may have less pixel area but larger affected range, so they should be displayed in many cases (Figure 1(a)).

Consider both aspects of the shape and also that of the pixel area, the concept of GR is proposed. A GR is a rectangle on the screen that is used as the criterion for small polygons. For example, an \( M \times N \) GR is a rectangle that is \( M \) pixels wide and \( N \) pixels long. In a multiscalar display, when a polygon could be exactly enclosed in the GR, that is to say when its width or height is shorter than \( M \) or \( N \), it would be generalized. Of course, at different display scales, the GR corresponds with different geographical areas.

3. The GS

Several further concepts are proposed here to improve the implementation of multiscalar displays – “display scale” and “Generalized Scale” (GS). The attribute GS of an object helps to determine whether it should be generalized at a certain display scale, thereby partially embodying the importance of an object. The calculated GS value should differ according to the application considered.

3.1. Scales and multiscalar display

When spatial data are displayed as a map and the user zooms in or out, the display scale changes accordingly. For an electronic map, there is no fixed scale.

For purposes of screen display, the concept of “display scale” is more useful and intuitive. Display scale can be obtained as follows: suppose ScreenWidth is the width of the map view in terms of pixels, whereas the ScreenWidth’s corresponding geographical distance is CorWidth, measured in km. Then the display scale (ScreenScale) can be calculated as:

\[
\text{ScreenScale} = \frac{\text{ScreenWidth}}{\text{CorWidth}}.
\]

3.2. Generalized scale

GS, as proposed here, is an attribute of an object. In multiscalar displays, when the display scale is below the
GS of an object, suppose that object A’s GS is “imp” and the display scale is Screen Scale, then:

\[
\text{If}(\text{ScreenScale} \leq A \cdot \text{imp}) \{A \cdot \text{generalize();}\}
\]

\(A \cdot \text{generalize()}\) is a function to generalize object \(A\).

In this manner, we determine whether an object should be displayed or generalized in a multiscalar display.

In setting the value of GS for an object, we first set a GR on the screen. If object \(A\) could be exactly enclosed in the GR at a certain display scale, the display scale is called the GS of object \(A\). Suppose that the GR is \(M/C_1\) (\(M\) pixels wide, \(N\) pixels long), and that object \(A\)’s geographical width and height are Geo WIDTH and GeoLENGTH, respectively, then \(A\)’s GS is calculated as follows:

\[
A \cdot \text{GS} = \text{Min}(M/\text{GeoWidth}, N/\text{GeoLength}).
\]

\(\text{Min()}\) is a function returning the smaller valued of the two parameters. As shown in Figure 1, the object’s width and height are equal to \(M\) and \(N\), so the value of \(M/\text{GeoWidth}\) equals that of \(N/\text{GeoLength}\). But all objects are not always the case. As shown in Figure 1(b), the object could be exactly enclosed in GR, and the present display scale should be the smaller one of \(M/\text{GeoWidth}, N/\text{GeoLength}\). When the display scale is below \(A\cdot\text{GS}\), object \(A\) should be generalized.

As usual, in multiscalar display and cartographic generalization, more important objects would be generalized at smaller scales; less important objects would be generalized at larger scales. As mentioned above, an object’s GS determines at what scale it should be generalized, so GS partially embodies the importance of an object. However, the importance of an object may differ from application to application. For example, for land use application, bodies of water (such as lakes) may not be allowed to be generalized at certain scales. For dry lands, bodies of water are quite important, so even small bodies might be displayed at any scale. Thus the GS of an object should not be set only by size. If an object must be displayed at any scale, a special value could be specified for its GS, such as 0, which would naturally lead it to be displayed at all times, according to the procedures described above.

In conclusion, the importance and size of an object determine if it should be generalized or displayed at a certain display scale. GS can be used to indicate the importance of an object by indicating the scale range at which an object should be generalized. Yet “importance” involves many factors beyond the shape and area of an object. For different applications, different mathematical formulas will be needed to calculate GS values for objects, a topic into which further research will be beneficial.

4. Sampling discrete scales

For traditional cartographic generalization, there is only one specified objective scale, but for multiscalar display, a fixed number of scales are needed. A scale sequence must be established. The establishment of a scale sequence should take into account the fact that the map needs to be changed fluently and fit to view during zoom.

4.1. What scale data to display in real time

During the zoom (in or out), the system must determine in real time what scale data to display according to the display scale. Our solution chooses the scale that is closest to the current display scale. For example, if there are \(n\) scales of data in the multiscalar database, the scale sequence is Scale\(_1\), Scale\(_2\), ... Scale\(_n\), and the current display scale is ScreenScale, then the value of Scale\(_x\) which will be chosen to be displayed should meet the requirement of:

\[
|\text{ScreenScale} - \text{Scale}_{x}| = \text{min}(|\text{ScreenScale} - \text{Scale}_{1}|, |\text{ScreenScale} - \text{Scale}_{2}|, \ldots, |\text{ScreenScale} - \text{Scale}_{n}|),
\]

where function \(\text{min()}\) determines the smallest of all the parameters.

4.2. Establishment of the scale sequence

The purpose of multiscalar display is to ensure that the map view is clear enough to be viewed during the zoom,
thus the establishment of the scale sequence should be user oriented. Based on many experiments, several possible methods for establishing the scale sequence are proposed here.

4.2.1. Equally dividing all the objects that may be generalized during the multiscalar display

For such a procedure, the minimal and maximal display scales must be specified exogenously. Our experiments only concern the object’s size; therefore, the GS is calculated by the GR such that an object’s GS is the scale at which the object could be exactly enclosed in the GR. Consequently, the maximal scale is the maximal GS of all the objects. When the display scale is greater than the maximal scale, no object will be generalized. At the other extreme, the minimal scale is the overview display scale. After the GS of all objects is calculated, we rank the objects in accordance with the GS in the descending order and divide them into equal groups. For example, assume that at the minimal scale, there are 10,000 polygons needed to be generalized. Then we divide all the polygons into five equal groups, each having 2000.

Then we need to acquire the scale sequence (Figure 2). The value of Scale1 is the GS of No. 2000, and the GS of each polygon of the first 2000 is greater than or equal to Scale1. These polygons must be generated at the scale Scale1 or below. Likewise, the other scales are acquired, establishing the scale sequence.

We have experimented using polygon data of a land-use map for Beijing, China. We set a sequence of 10 scales and a GR was used to calculate the minimal display scale as well as the number of generated polygons at each scale (Figure 3). This sampling method could guarantee that at each scale equal numbers of objects would be generalized. Yet, the size of all the objects has no distribution rule, so this scale sequence has no distribution rule accordingly.

4.2.2. Arithmetic sequence

Alternatively, to establish an arithmetic sequence of scales, first the value of the minimal scale needs to be set. In experiments we chose the value of the overview display scale as the minimum. Suppose that the minimal display scale in our application is minScale, then the minScale can be obtained as follows:

\[
\text{minScale} = \text{Min}(\text{HeightOfView}/\text{HeightOfLayer}, \\
\text{WidthOfView}/\text{WidthOfLayer})
\]

where \(\text{Min()}\) is a function to get the smaller of the two parameters. HeightOfView and WidthOfView are the height and width of the map view in screen, respectively. HeightOfLayer and WidthOfLayer are the geographical height and width of the map displayed in the view.

In overview, if a user continues to zoom out when the display scale is below the minimum, there is no map at a smaller scale, so only the data of the minimal scale is displayed. The maximal scale is the scale greater than which there are no polygons to be generalized. The maximal GS of all the objects could be set as the maximal scale. Suppose that the value of the maximal GS is maxGS, when the display scale is greater than maxGS, no objects will be generalized; as a result the original data will be displayed. Actually, maxGS is critical, but it does not affect the establishment of the scale sequence. The arithmetic sequence is then obtained by inserting a sequence of scales between the minimal and maximal scales with equal intervals.

Suppose the minimal display scale is minScale, and the maximal is maxScale. To obtain a sequence of six elements, four scales must be sampled between minScale and maxScale. The value of Scale\(_i\) is calculated by:

\[
\text{Scale}_i = \text{minScale} + i \times (\text{maxScale} - \text{minScale})/5
\]

In this way, the arithmetic sequence of scales is established (Figure 4). We can then determine those polygons that need to be generalized at each scale according to their GS.

Although the scales have even distribution, there are large differences in the numbers of objects generalized between scales. This method and the former have the same drawback – neither is as user oriented as the one that follows in Section 3.

4.2.3. Geometric sequence

To establish a geometric sequence, the minimal scale, the element count of sequence and the geometric factor must be specified. The minimal scale can be obtained in the same way as mentioned above. The geometric factor may vary from topic to topic. However, in our experi-
ments, we used the geometric factor 1.4, which is commonly used in GIS software. In fact, the value does not matter, because it is here only for testing purposes.

Suppose that in our application, the minimal display scale is $\text{minScale}$, nine other scales are desired, and the geometric factor is 1.4. The sequence is established as follows:

$\text{minScale}, \text{minScale} \times 1.4, \text{minScale} \times 1.4^2, \text{minScale} \times 1.4^3, \ldots \text{minScale} \times 1.4^9$

In our demonstration, the user interface enabled users to set the geometric factor themselves. The relationship between display scales and the count of objects generalized at each scale are illustrated by cartogram shown in Figure 5.

### 4.2.4. Discussion about scale sampling methods

The first method (equal division) establishes the scale sequence in a way that ensures the number of polygons is equal from portion to portion, but has not considered the distribution of the sizes of all polygons. Generally, in a map, there are relatively fewer objects that are very small or very large. Thus, moderate scales should often present a denser distribution by this method. When
zooming, the view map may easily skip many moderate scales, so the view map may change quickly zooming across some scales and slowly zooming across others. The second method (arithmetic sequence) ensures that the scales of the sequence have the same interval, but at some scales there are too many objects generalized, other scale too few generalized.

What about the third method – that of the geometric sequence? As usual, a map is fully displayed in the overview. Suppose the display scale is minScale in overview, and the geometric factor is $C$. After the user zooms in once, the display scale changes to $\text{minScale} \times \frac{1}{C}$. Likewise, when the user continues to zoom in, the display scales will be $\text{minScale} \times \frac{1}{C^2}$, $\text{minScale} \times \frac{1}{C^3}$, and so on. The zoom function is based on the current map view, with a geometric scale sequence. If only considering the zoom operation, the geometric sequence may be a better choice. Of course, users may zoom in or out by windowing. If a user zooms in/out by drawing the same rectangle, the display scale sequence is still geometric. But if the user draws a rectangle of random size, the display scale sequence would have no distribution rule. Because the user’s zoom function is based on the current map view, we hold that the geometric sequence is more feasible.

5. Case study
Given the aforementioned criteria for small objects, the GS of objects, and the method of scale sampling, several experiments were done. In the prototype system, the GAP-tree index was used, as described below.

5.1. Review of GAP-tree
The GAP-tree is an area-partitioning hierarchy and is used to decide not only which area is removed but also which area will fill the gap of the removed feature $(7, 12, 19)$ The GAP-tree was introduced to avoid holes on a map created by the Reactive-tree. An example in Figure 6 illustrates the GAP-tree. It shows a generalization example in five steps, ranging from detailed to coarse displays. Each polygon is given an ID and a computed importance value (shown in a smaller font). The importance value is calculated by Equation (1) $(11)$. Less important objects receive lower values, while more important objects receive higher values. At every step, the least important object (i.e. feature 6 in step 0) is removed, and its area is assigned to its neighbor (i.e. feature 2 in step 0) with the highest value (calculated by Equation (2)). Also, a new ID is assigned to the new polygon, which is enlarged and becomes more important. The importance value of the enlarged object will be recomputed according to Equation (1). This process continues until one large polygon is left. Therefore, the hierarchy of Figure 6 can be stored as the GAP-tree shown in Figure 7.

$$I(o) = f(\text{type}, \text{attributes}, \text{size}) \quad (1)$$

$$\text{Collapse}(a, b) = f(L(a, b), \text{CompatibleTypes}(a, b), \text{WeightFactor}(b)) \quad (2)$$

where $L(a, b)$ is the common boundary between $a$ and its neighbor $b$, CompatibleTypes$(a, b)$ determines how close the two feature types of $a$ and $b$ are, WeightFactor$(b)$ is weight factor of the feature type of $b$.

5.2. The revised GAP-tree
In our experiments, we implemented a revised approach to the GAP-tree. As we can see in Figure 7, the original GAP-tree is binary. A binary GAP-tree can be so expansive that it requires a large disk space to store the data. It records in detail the generalization procedures of every two features from many nodes to one root. This procedure would make the GAP-tree deeper, and then slow

Figure 6. Generalization scene (from Van Oosterom (12)).
the system’s response time. However, it is unnecessary to detail the generalization process for every two objects. Alternatively, we might select key representations to describe the generalization process. Our solution sampled a few scales to realize a multiscalar display. We revised the GAP-tree from binary to multiway (Figure 8). There are three levels in the multiway GAP-tree as Figure 8 shows. The three levels correspond to three scales. Level 0 corresponds to the original scale, whereas Level 1 and Level 2 correspond to the other two generalizations. Thus, the revised GAP-tree will use three scales to realize multiscalar display. However, the implementation of the revised GAP-tree in a relational database management system (RDBMS) remains a topic for future discussion and reflection.

5.3. Test data and setting parameters

The test data are Beijing land-use data at 1:10,000 scale. It contains 186,329 polygons and 11,668,862 vertices. The SDBMS used in our test was PostgreSQL and PostGIS, running on a Dell 2600 computer. Client software was developed and run on a P4 computer with a 1-GHz CPU and 256 MB of RAM. Named Multi-viewer, the client software was programmed in Java, though the module building the revised GAP-tree index was coded with C.

To build the revised GAP-tree, a few parameters needed to be defined, and were set as described above. In our experiments, we set the map view to be 200 by 150 pixels. The geometric sequence was adopted. The overview display scale (0.000836) was preset, as was the geometric factor (at 1.4), with the other five elements of the sequence of display scale computed as described. We used $2 \times 3$ GR and $4 \times 6$ GR, respectively to do the multiscalar display, as an arbitrary test. We also displayed spatial data without generalization. Response times can thus be compared with each other. Using GR, the GS of every object was computed. As mentioned above, the value of an object’s GS is the display scale at which the object could exactly be enclosed in a GR. Because the GS is the scale at or below which an object will be generalized, the lesser the GS, the smaller the display scale, the more important the object. Our solution used objects’ GS as the measure of importance. Based on the scale sequence and GS, the revised GAP-tree was built. After setting the parameters, multiscalar display would be implemented as follow:

1. Build and store the revised GAP-tree.

   (i) Establish the display scale sequence. Suppose the overview display scale is minScale, then the sequence is: 
   
   minScale, minScale $\times 1.4$, minScale $\times 1.4^2$, minScale $\times 1.4^3$, minScale $\times 1.4^4$, minScale $\times 1.4^5$.

   (ii) Calculate the GS of every object. An object’s GS is the scale at which the object could be exactly enclosed in the GR ($2 \times 3$, $4 \times 6$). Suppose object A’s geographical width and length are GeoWidth, GeoLength, then

   $A \cdot GS = \min(3/GeoWidth, 2/GeoLength)$ or

   $A \cdot GS = \min(6/GeoWidth, 3/GeoLength)$.

   (iii) Generalize all the objects whose GS is below minScale $\times 1.4^5$. That is to say, when the display scale is below or equal minScale $\times 1.4^5$, the generalized objects will not be displayed. Mark these generalized objects as level 0 and store the relationship of affiliations between objects.

   (iv) As (iii), generalize all objects whose GS is below minScale $\times 1.4^4$. Repeat this process until all the objects whose GSs are below minScale are generalized.

   This way, the tree is created. But the tree may not have one root. It may be a forest. We call it a revised GAP-tree.

2. Select all the objects of a certain scale in real time according to the revised GAP-tree when user zoom in/out.

   (i) Calculate the present display scale.

   (ii) Choose the data of the scale which is nearest to the display scale and display them.

   Selection of the objects must be implemented according to the logic of revised GAP-tree.

5.4. Analysis of testing results

After building the revised GAP-tree, we took the center point of the map data as the center of the map view. We
Table 1. Statistics of multiscalar display.

|   | No generalization | Using 2 × 3 GR | Using 4 × 6 GR |
|---|------------------|---------------|---------------|
| 0 | Overview         |               |               |
|   | Time used: 95.84 S | Time used: 4.87 S | Time used: 2.71 S |
|   | Number of records: 186,329 | Number of records: 5009 | Number of records: 1405 |
| 1 | Zoom × 1         |               |               |
|   | Time used: 57.16 S | Time used: 8.15 S | Time used: 6.94 S |
|   | Number of records: 108,549 | Number of records: 9111 | Number of records: 2872 |
| 2 | Zoom × 2         |               |               |
|   | Time used: 17.84 S | Time used: 3.32 S | Time used: 2.72 S |
|   | Number of records: 38,606 | Number of records: 8389 | Number of records: 2958 |

(Continued)
|   | No generalization | Using 2 × 3 GR | Using 4 × 6 GR |
|---|------------------|---------------|---------------|
| 3 | Zoom × 3         | ![Image](image1) | ![Image](image2) | ![Image](image3) |
|   | Time used: 4.62 S | Time used: 1.25 S | Time used: 0.95 S |
|   | Number of records: 10,061 | Number of records: 4865 | Number of records: 2097 |
| 4 | Zoom × 4         | ![Image](image4) | ![Image](image5) | ![Image](image6) |
|   | Time used: 1.51 S | Time used: 0.50 S | Time used: 0.36 S |
|   | Number of records: 2440 | Number of records: 1848 | Number of records: 1147 |
| 5 | Zoom × 5         | ![Image](image7) | ![Image](image8) | ![Image](image9) |
|   | Time used: 0.32 S | Time used: 0.36 S | Time used: 0.32 S |
|   | Number of records: 417 | Number of records: 417 | Number of Records: 417 |
first displayed the map fully and then adjusted the display scale to the above-mentioned 5 scales. The number of queried records, the average response time of system, and the screen snapshots are recorded in Table 1.

From this table, it can be concluded that the multiscale display dramatically reduces the response time while keeping a comparable visual quality. The geometric scale sequence makes the map view change more fluently and naturally during a zoom operation. With an adequate scale sequence, we can thus simulate scaleless display adequately. From the comparison of screen snapshots between “using 2×3 GR” and “using 4×6 GR,” we can see that when using 4×6 GR, the response time is shortened, the number of records is smaller, and the map is clearer, consistent with what we concluded before.

6. Conclusion and prospect

To realize a multiscale display, several issues must be resolved, including the definition of “small” objects able to be generalized as well as the method for sampling a sequence of discrete scales. Some such problems have long been raised by traditional cartographic generalization, yet they often have new complexities within a multiscale display, an issue we address. Finding that existing methods to define small objects were open to improvement, we proposed a new method. Further, with respect to the establishment of the scale sequence, all three methods proposed have their advantages and disadvantages, respectively, and the quality of results must be decided by reasoning with respect to criteria desirable within a situation as much as by eye alone. Our experiments chose a geometric sequence according to the rule of the user zoom function, for the fluency of the map view change. In such a manner, we have laid an effective foundation for future research into multiscale display of vector data in GIS.

Moving forward, at least three areas of future research are suggested. First, it should be possible to continue improving the definition of small objects. Only the sizes of objects are considered in our solution, but the proportions of the areas of objects to the areas of GRs (refer to Section 2.1) have not been taken into account. Nor have the results of examining GRs rotated between “using 2×3 GR” and “using 4×6 GR,” we can see that when using 4×6 GR, the response time is shortened, the number of records is smaller, and the map is clearer, consistent with what we concluded before.

Secondly, there are opportunities to develop additional scientific sampling methods to select reasonable representations. The selection of key scales should be deduced by the characteristics of vector data, the characteristics of map view, and the users’ requirements for response times. The sampling methods in Section 4.2 are only rough approximations. More scientific sampling methods are needed to make the multiscale display more fluent and rational.

Thirdly, the calculation of the GS may be improved. According to our solution, an object’s GS determines the scale at which it must be generalized, thus the GS partially embodies the importance of an object with respect to certain applications. However, our experiments only consider the size of objects. Attention should also be focused on how to better relate the GS of an object to its importance, which requires the development of models based on expert knowledge of application domain requirements. Finally, an ideal system should provide interfaces to enable different users to define their own equations to compute the GS.

Our study has thus helped demonstrate the potential for multiscale display of vector data, such that these issues now would benefit from further investigation, which would further illuminate the promising applications of GIS visualization.

Acknowledgements

Supported by the Knowledge Innovation Program of the Chinese Academy of Sciences (No. kzcx2-yw-345).

Notes on contributors

Niu Fangqu is an assistant professor. His major research interests are in Geographic Information Science as well as its applications in human geography, including (1) spatial analysis and visualization, (2) visualization and modeling of human geography process, and (3) urban boundary growth.

Cheng Changxiu is an associate professor doing research in GIS including multi-scale visualization of vector data and multi-scalar database.

References

(1) Spaccapietra, S.; Vangenot, C.; Parent, C.Z.; Zimanyi, E. MurMur: A Research Agenda on Multiple Representations. Proceedings of DANTES’99, Kyoto, Japan, 1999.
(2) Ai, T.H. Key Technologies and Strategies in the Development of Multi-Scalar Spatial Database. Sci. Technol. Rev. 2004, 2, 4–8.
(3) Wang, Y.H.; Cheng, J.; Jiang, J. Multi-Scalar Conceptual Model for GIS Geographical Features. Min. Sci. Technol (China) 2003, 32 (4), 376–382.
(4) Becker, B.; Six, H.W.; Widmayer, P. Spatial Priority Search: An Access Technique for Scale-Less Maps. Proceedings of ACM SIGMOD, 126–137, 1991.
(5) Cai, Z.L.; Wang, Q.; Weng, M.; Jiang, W.P. Information Organization and Visualization Mechanism of Electronic Map. Geo-spatial Inform. Sci. 2008, 11 (4), 262–268.
(6) Chan, E.F.; Chow, K.K.W. On Multi-Scale Display of Geometric Objects. Data Knowl. Eng. 2002, 40, 91–119.
(7) Cheng, C.X.; Fangqu, N.; Jun, C.; Yanlu, Z. Extensions of GAP-Tree and its Implementation Based on a Non-Topological Data Model. Int. J. Geogr. Inf. Sci. 2008, 22 (6), 657–673.
(8) Kwon, J.; Yoon, Y. An Access Method for Integrating Multi-Scalar Geometric Data. Proceedings of the 6th East European Conference on Advances in Databases and Information Systems, September 8–11, 2002, pp. 204–217.
(9) Li, Z.; Openshaw, S. Algorithms for Automated Line Generalization Based on a Natural Principle of Objective Generalization. *Int. J. Geogr. Inform. Sys.* 1992, 6, 373–389.

(10) Prasher, S.; Zhou, X. Efficient Update and Retrieval of Objects in a Multiresolution Geospatial Database. *Proceedings of 15th Int. Conf. on scientific and Statistical Database Management (SSDBM)*, July, 2003; pp. 193–201.

(11) Van Oosterom, P. The Reactive-Tree: A Storage Structure for a Seamless, Scaleless Geographic Database. *Proceedings of Auto-Carto 10: Technical Papers of the 1991 ACSM-ASPRS Annual Convention*. Baltimore: ACSM-ASPRS, 1991; 6: 393–407.

(12) Van Oosterom, P. Variable-Scale Topological Data Structures Suitable for Progressive Transfer: The GAP-Face Tree and GAP-Edge Forest. *Cartogr. Geogr. Inf. Sci.* 2005, 32, 331–346.

(13) Wang, H.; Ai, T.H.; Zhu, G.R. Adaptive Strategy on the Visualization of Electronic Map (in Chinese). *J. Wuhan Univ.* 2004, 29 (6), 525–528.

(14) Wei, Z.; Cho, S.; Kim, J.; Bae, H. A new spatial data representation in GIS, 2000. http://www.Geocomputation.org/2000/GC019/Ge019.htm (accessed March 21, 2003).

(15) Zhou, X.; Prasher, S.; Masaru, K. Database Support for Spatial Generalization for WWW and Mobile Applications. *Proceedings of 3rd International Conference on Web Information Systems Engineering (WISE 2002)*, Singapore: IEEE Computer Society, December, 2002; pp. 239–246.

(16) Wu, H.Y.; Zhu, H.J.; Liu, Y. A raster-Based Map Information Measurement for Qos. *Proceedings of ISPRS*, Istanbul, Turkey, 2004; pp. 365–370.

(17) Yang, B. A Multi-Resolution Model of Vector Map Data for Rapid Transmission Over the Internet. *Comput. Geosci.* 2005, 31, 569–578.

(18) Cheng, C.; Lu, F.; Cai, J. A Quantitative Scale-Setting Approach for Building Multi-Scalar Spatial Databases. *Comput. Geosci.* 2009, 35, 2204–2209.

(19) Van Oosterom, P.; Schenkelaars, V. The Development of an Interactive Multiscale GIS. *Int. J. Geogr. Inform. Sys.* 1995, 9, 489–507.