Interactive Generalization on Large-Scale Topographical Map Supported by a Database Platform

CAI Zhongliang, WU Hehai, DU Qingyun, LIAO Chujiang

1 Introduction

The traditional manual cartographic generalization needs the operators to master enough professional knowledge and skills and to possess certain work experience, which is very difficult to be fulfilled. Furthermore, this method brings about low efficiency, a long time consuming as well as big contrived errors. With the transformation of user’s consumption concept, as far as map production is concerned, a swift reaction to market is demanded. Obviously, for its poor efficiency, this kind of map production whose majority is constituted by paper map is difficult to meet the users’ demands. In addition, it can not keep up with the quick development of information at all.

At present, to a great extent, the manual method still prevails in map generalization. That is to say, on one hand, we have achieved automatic cartography, on the other hand, we are adopting manual cartographic generalization. It will be too difficult for the performers to simultaneously work under two different environments, which will also destroy the system of automation and integration.

In order to meet new requirements presented by digital environment, it should be a development direction of map making to adopt new technology to improve the efficiency of map compilation.
2 Large-scale topographic map and interactive cartographic generalization

2.1 Characteristics of large-scale topographic map

By means of graphics decomposition, large-scale topographic map may consist of the following two parts: map graphics and character annotation. The former includes three kinds of graphics elements: point, line and area. A digital topographic map must cover all spatial information and attribute information of six types of features: single feature, residential area, drainage basin, relief, pipe line and border, vegetation. In terms of large-scale topographic map, because almost all human and natural trivial elements on the earth surface require to be precisely and really reflected and added to the considerable amount of information. Due to the quick changing and updating speed, the verisimilitude these elements is very bad, which embodies more obviously when man-made elements are represented.

In large-scale maps, the entities represented by features are difficult to form objects, the majority of them consist of complex geometric structures, and this kind of extraordinarily elaborate express, to certain extent, affects the simplification and abstraction of data model based on the objective world. When added into the database, this kind of data representation is little similar to a full replication; the scattered structure of the entity world still remains unchanged. For example, in 1:1,000 scale map, a house is not only denoted as a simple polygon, but attaches some other linear structures, such as balcony, porch and its pole, ladder, stairway etc. That is to say, the majority of objects are compound, which aggravates the difficulty of cartographic semantics identification, and make operators designed for cartographic generalization excessively focus on the generalization of geometric information and the disposal of topological relations between elements. Urban large-scale topographic map is characterized by numerous man-made map elements (such as house, street, pipe net, construction facility, etc). Differing from natural features, these features take on obvious human traits. For instance, the borders of streets can not embody fractal characteristics as natural features (rivers, coastlines, etc) do. On the whole, the features distributing with human traits develop a unique structure. For example, the angle of a house polygon is a right angle, and the houses lying beside the two sides of roads are regular and in arrays.

2.2 Interactive generalization based on large-scale topographic map

Looking through the characteristics of large-scale topographic map mentioned above, we can conclude that its representation of content elements is more detailed, and the relations between them are also more complex. Presently, the basic theory of automatic generalization is not mature. Therefore, realizing purely automatic generalization based on such complex information relations is hardly possible at all. By taking insight into the characteristic of large-scale topographic map, it is not difficult to find out the majority of characteristics and relations of map content elements.

Interactive generalization, however, is a human-computer cooperative working style. During the course of generalization, the performer is required to participate in such work as selecting generalization operators, setting parameters, feeding back the executing results, etc. And computer, depending on certain software arithmetic, executes the basic generalization operation. The generalization process as a whole may be summarized as the problem of 3W+H. That is, When——when does the user put forward the condition of generalization and simplification to execute the generalization transformation of shorte-
Where do the spatial occupancies generate conflicts, and where are the features too dense? Which features are important? How to execute cartographic simplification, replacement, and conflation etc. In these problems, 3W problems belong to the deeds involving in powerful intellective generalization, reasoning, and judgment, which can presently only be solved by hand. But H problems can be accomplished by perfect generalization operators. Indeed, in the whole task, the disposals of 3W problems account for a larger proportion, however, our brains’ thinking, as is known to us, can swiftly arrive at an answer, which consumes a little time. H problems, dealing with the compilation and maintenance of geometric graphs, should be categorized into physical labor. Provided they are solved by hand, it might need a very long time. So adopting this generalization method, human and computer achieve a kind of mutual complement through cooperative work. Compared to the traditional manual operation, its efficiency proves to be an evident improvement.

Consequently, it can be seen that the design and construction of data structure and data model directly determine whether the characteristics and demands of interactive generalization based on large-scale topographic map can be rightly and flexibly reflected.

3 Construction of map generalization-oriented database platform

3.1 Hierarchical organization of generalization objects

While designing the basic platform of map database (which includes designing the database model and organizing the data structure), the development of the subsequent application functions should be sufficiently taken into account, and the designed platform should present data objects with all kinds of structure relations that the map generalization operations need, among which the hierarchical organization method is regarded most important.

(1) The hierarchies of the generalization objects should be partitioned logically. In the traditional cartographic generalization, we execute it on different hierarchies. This idea is still of significance in software designing. It is essential to construct versatile map layer manager, which is used to provide the compilation functions orienting operations. After the hierarchies of objects to be generalized are partitioned, the generalization operators toward these hierarchies and the corresponding control parameters will be determined eventually, which are the foundation of application module development in the subsequent generalization process.

During the course of partitioning the hierarchies, such respects as the basis of classifications of map content elements, geometric features (including point, line, area, net, etc.), spatial relativity etc. should be given enough consideration.

The generalization element hierarchy should approximately embody three characteristics: operational sequence, structural singularity and element hierarchies’ overlapability. Operational sequence suggests that while solving spatial conflicts among elements of different hierarchies, the priority of generalization must be taken into account, keeping the features with the higher priority immobile, and deleting, cutting, replacing the features with less priority. The purpose of structural singularity is to meet the operational requirements of generalization operators. Element hierarchies’ overlapability is designed to realize all the element hierarchies’ overlapping after they are generalized, so that we can adjust spatial relations between them and eliminate the conflict contradictions.

(2) In the software system, how to organize the hierarchy relations before generalization and after generalization is also a problem that should
be solved. Generalization operations, not the same as those common graphics editing operations, which may either directly substitute new objects for old objects, or resume the original state by using undoing operations, involve in considerable complicated computations. Furthermore, under most situations, the operational object is always towards multi-item. In the interactive undertakings, in fact, whether a generalization operation is in accordance with criterion is mainly justified by performer. Because this kind of justice is achieved through comparing the state before the generalization with the one after the generalization, it is entirely wrong to adopt the editing operations similar to replacement. The generalization results ought to be derived from the base map data. For adapting to this requirement, it is necessary to simultaneously store two hierarchies of data before generalization and after generalization.

3.2 Design of system based on object-oriented (OO) method

The OO method takes objects as the most fundamental elements, overcoming the disadvantages that the relation between data structure and behavior is not very compact. Meanwhile, the OO method develops such outstanding peculiarities as modularity, information encapsulation and hiding, abstraction inheritance, polymorphism etc., which offers a most valid instrument and approach for managing large software and advancing software reliability, reusability, expansibility and maintainability.

When building large-scale topographic map database, generally, we pay little attention to the need of cartography, but focus on considering it as a geographical information database. Thereby it is spatially vital to execute the abstraction of data type. This system database platform, based on the characteristics of cartographic generalization and GIS, CAD technology, can be abstracted as graphics, layer, object, geometrical class (including point class, line class, area class, annotation class, path class, region class, and group class). The hierarchical structure can be described as: graphic->layer->object->geometrical class, combining with the operation class related to the cartographic generalization. The design of the system may be interpreted in Fig. 1.
The object class of this system mainly consists of four parts: system interface, database management platform, basic operators and arithmetic, element generalization process. Besides, it includes graphic symbol design and data interface, etc.

The map database management platform class includes graphics class, point class, line class, area class, annotation class, path class, region class, and group class (Table 1).

The element generalization process class includes building class, drainage basin class, vegetation class, and relief class (Table 2).

The generalization operator class may be divided into Delaunay triangulation network class, overlay analysis class, etc (Table 3).

While carrying out module design, this system makes the best use of the inheritance and polymorphism. For example, the element generalization class, through multi-parent inheritance, can obtain the inheritances of all kinds of operations defined by a set of generalization operators.

### Table 1 Classes of map database platform

| Object class | Descriptions of main attributes | Main operations |
|--------------|---------------------------------|-----------------|
| Graphics class | scale denominator, graphics name, scope, underling layers and geometric elements of spatial index, coordinate system, saved file name | create, data input, data output, index constructing, delete, read and write, show, save |
| Layer class | State, layer name, operation characteristics, geometric properties, underling geometric elements | create, copy, move, delete, show, data input, construction maintenance |
| Geometric object class | Geometric coordinates, attributes, state, keyword, boundary rectangles, index grids, structure relations | add, delete, move, read and write, show, register, grid index building, topology organizing |

### Table 2 Feature Generalization classes

| Object class | Descriptions of main attributes | Main operations |
|--------------|---------------------------------|-----------------|
| Building class | Coordinates of building polygon, the layers of building, structure of building, adjoining buildings, shape, the smallest boundary rectangle | Partition of buildings groups, contiguous buildings recognition, replacement of building, simplification of shape, deletion, conflation, evaluation |
| Drainage basic class | Coordinates of polygon, properties of triangulated network, minimum bounding rectangle, the description of shape, the relations of polygons | Recognition, filtration, deletion of small lake, bi-line river is converted into single-line river, elimination, conflation, simplification, and evaluation of islands |
| Road class | Coordinates, length, properties of road, the description of part convex, contiguity relations, the characteristics of bends | Deletion, conflation, join, replacement, extraction of axis, simplification, summary of bends properties, elevation |
| Relief class | Coordinates, characteristics, elevation, contiguity relations of contour lines, valleys, ridges, elevation points | Filtration, join, deletion of contour lines, simplification of bends characteristics, constructing Voronois, smoothness |
| Vegetable class | Coordinates, area, perimeter, attribute characteristics, contiguity relations, the characteristics of bends on boundary, and shape of polygon | Deletion, simplification, combination, replacement, constructing Delaunay Triangular network, evaluation |

### Table 3 Generalization operator classes

| Object class | Descriptions of main attributes | Main operations |
|--------------|---------------------------------|-----------------|
| Triangulated irregular network class | Coordinates of group points, conditions of triangulated network, the vertexes of triangle, neighboring triangles, the center of triangular gravity | Contraction of Delaunay triangulated network, predisposal of data, extraction of axis, conflation Voronoi diagram, triangulate network maintenance |
| Overlay analysis class | Boundaries of polygons, the attribute conditions, islands, contiguity relations | Overlay analysis, computation of minimum bounding rectangle, difference combination, simplification, uniting, conversion between vector and raster |
3.3 Logical organization of generalization-oriented map database

Logistic hierarchical structure of database is organized in accordance with the system: graphics->layer->element class->object->geometric attribute description, which may be expressed as in Fig. 2.

![Diagram of Logical organization of database](image)

The idea of from up to down adopting tree structure to build database, ensures the consistence between the physical storage of program realization and the application-oriented logic structure, that is, as early as the time of storage and managing database, the hierarchical relations is embodied, so we can directly acquire the information we retrieve with no excess search calculations needed, improving the usage efficiency of the database.

In Fig. 2, all objects in this database are registered in a spatial grid index, which will sequentially quicken the feedback speed of the object identification and retrieval.

Similar to the hierarchical organization of database, we may build the hierarchical structure of object classes based on object-oriented design idea. In every class, the encapsulation are exerted on the descriptions of class and the operations to data members, meanwhile, according to the affiliations of the data members and the object characteristics, we build a serial of inheritance relations among graphics, layer, element class, object.

In the seven kinds of element class defined above, point, line, area, and annotation belong to the simple object types, which are used to describe and save such simple entity objects as single facility, road, communication line, building, vegetation, lake, illumination text, enterprise name, etc. but path, region, and group are used to express compound objects. Path is used for the storage of drainage basin network and road network; region is used for the storage of such group polygon structure as the buildings group and lakes group, etc; group is used to express those compound structures of objects, which may be any type. However, here it should be emphasized that path, region and group only provide compound objects with the storage of structure frame, because the compound objects mentioned above are all derived from simple objects by means of relational operators and additive information. We need not save the simple objects a second time. We only save the basic structures of these compound objects in database, through which we can certainly get concrete data. For example, in the buildings generalization, according to the spatial contiguity relations, some simple polygon structures of buildings are identified forming a group structure. At this time a compound object is derived from it,
and we only need save it using region group structure.

The design of path and region structure in this paper shares the idea of using network to analyze path and using region to analyze region in Arc/Info system. The compounding relation between simple objects and compound object is also approximately coincident with the idea in Arc/Info system.

3.4 Coordinate system

Coordinate systems involved in this database include: geodesic coordinate system, drawing coordinate system before generalization, drawing coordinate system after generalization, database coordinate system, output device coordinate system, etc.

The map database generalization is the operation on graphics using virtual geodesic coordinate, but it is necessary to decide the executions of operations through the user’s vision under drawing coordinate system. The index rules of generation are also described under the drawing coordinate system. For example, we rule the minimum interval among houses is 1 mm, twenty single facility objects per sq. dm, etc. In AutoMap software, the logical descriptions of data, in the drawing coordinate system before generalization, are designated to adopt mm as unit, and during the course of physical storage of data, the coordinates describing data are transformed into database coordinates. The discrepancy between the two kinds of coordinates is only a multiple which takes the resolution in the system as a coefficient, and the coordinate origin lies in the center of drawing.

3.4.1 Transformation from geodesic coordinate to database coordinate

The transformation formula from the geodesic coordinate to the database coordinate is:

\[
\begin{align*}
X & = \frac{R}{2} \left[ x - (x_{\text{max}} + x_{\text{min}}) \right] \\
Y & = \frac{R}{2} \left[ y - (y_{\text{max}} + y_{\text{min}}) \right] \\
R & = \frac{1000U}{M} S
\end{align*}
\]

where \((X, Y)\) is the database coordinate; \((x, y)\) is the virtual geodesic coordinate; \((x_{\text{min}}, x_{\text{max}}, y_{\text{min}}, y_{\text{max}})\) is the range of virtual geodesic coordinate; \(R\) is the zoom coefficient of the transformation from geodesic coordinate to database coordinate; \(U\) is the unit of geodesic coordinate, such as meter; \(S\) is the system resolution; \(M\) is the denominator of scale.

3.4.2 Transformation from database coordinate to drawing coordinate

The transformation formula from the database coordinate to the drawing coordinate is:

\[
\begin{align*}
X & = \frac{x}{S} \\
Y & = \frac{y}{S}
\end{align*}
\]

where \((X, Y)\) is the database coordinate; \((x, y)\) is the drawing coordinate; \(S\) is the system resolution.

3.4.3 Transformation from base layer coordinate to generalization layer coordinate

The transformation formula from the base layer coordinate to the generalization layer coordinate is:

\[
\begin{align*}
X & = \frac{M_2}{M_1} x \\
Y & = \frac{M_2}{M_1} y
\end{align*}
\]

where \((X, Y)\) is the generalization layer coordinate; \((x, y)\) is the base layer coordinate; \(M_1\) is the denominator of scale after generalization; \(M_2\) is the denominator of scale before generalization.

Because AutoMap software is mainly used to carry out cartographic generalization based on large-scale topographic map, it is not necessary to consider the projection transformation. While outputting data, this system will question the selection from the geodesic system, drawing system before generalization, and drawing system after generalization. The data type of this system database is integer type, owing to the characteristic of real 32 bit in Windows NT system. The expression of integer type data is four bytes. Meantime, in the drawing coordinate system, we select 0.01 mm as resolution, both the two accuracies can meet the demand of the cartographic generalization.
3.5 Spatial grid index

In order to improve the speed of querying the objects in this database, the spatial grid index technology is widely adopted in the spatial database development. The bottom query in relate to spatial localization includes two aspects: one is to query which objects there exist in a grid, and the other is to query which grids a object lies in. The former is mainly applied to objects identifying and retrieval by window, while the latter is mainly applied to objects registering and grids index maintenance after some objects are deleted. To realize bi-directional query, a good method is building a bit matrix that takes the serial numbers of grids as the rows and takes the keywords of objects as the lines. However, in this software, due to the enormous amount of objects (if twenty-five 1,000 map sheets are joined, the number of points, lines and areas will add up to more than 80,000), the amount of storage needed by the matrix is so large that the bi-directional query is difficult to achieve.

This system selects the single-directional storage, that is, for the n by n grids, we do not register which grids a object goes through, but register the objects keywords contained in every grid, which meet the demands of the objects identifying and retrieval. When the user delete a object, this system executes the real-time register calculations, then logs out the corresponding keywords contained in the grids that this object goes through, because the registration calculations need not spend much time, and this kind of real-time calculation has no effects on the running efficiency of the system.

When the system registers objects, the point objects, according to the grid locations they lie in, are registered point objects keywords; line objects are registered line objects keywords in terms of the grids serial numbers they go through; but for area objects, this system registers their keywords in the grids that their minimum bounding rectangles cover; annotation objects are registered with respect to the little square determined by the localization point and the size of every word; and path, region, group objects are not built a grid index.

3.6 Building and maintenance of map database

The data source building database is approximately data files containing topological structure and attribute information, and the operations building database are mainly completed by correlative functions defined in map class, here we only make several rough rules for the process of building database:

① reading information content files of data source,
② regrouping topological relation, getting the information about the arcs forming polygon and the external ring of islands,
③ performing the register of point, line, area, and annotation objects, and registering them in the grid index,
④ saving the series of keywords, coordinate strings and header information of objects.

After the data are read into AutoMap, the system does not immediately create external memory files, but saves all of them in memory buffer to perform management operations. Only when the storage operations are activated, are the external memory files: *.amg, *.xy, *.key created and saved.

It is a complicated process to maintain the database, after the user deletes an object, the corresponding operations include:

① removing the object keyword,
② removing the registers of this object in the correlative grid indexes,
③ maintaining the information of the relations between this object and other objects.

Taking arc as an example, after an arc is deleted, the polygon containing this arc will not exist, either. It must be emphasized that all objects' deletion operations do not include immediately calling back their storage spaces. As the user performs undoing operations, the system only needs to change 0 into 1.
3.7 Design of database query

The query of this system is designed as the three manners: query according to spatial positions (localization identification, windowing, arbitrary polygon), query in terms of logic conditions (element code, layer, geometric character class, area, perimeter, etc.) and query with respect to structure relations. The query results of different processes can be carried out AND, OR, XOR compounding for any times.

The query result employs bit strings to express; an integer of integer type denotes 32 bit strings, then 6 000 integers of integer type can denote 192 000 bit strings. The position of bit represents the value of the keyword. That the value of bit is 1 denotes that the keyword has already been selected; or it has not been selected. The method of bit express perfectly supports the logical calculations of the results.

In query class, such a group of bit operation functions are defined: READ bits, WRITE bits, MODIFY 1 or 0, AND, OR, XOR calculations, TRANSFORM bit strings into selected objects keywords, etc.

4 Experiments and analysis

Taking the $1:1000$ and $1:10000$ database data of Shenzhen city as examples, the authors producted a number of experiments. The results proved that this kind of cartographic generalization oriented database platform had high stability, and this kind of human-computer interactive generalization environment based on this database can well help performers finish the generalization work under the digital environment (Fig. 4 and Fig. 5). Though the automatic degree is not very high,
the effect of generalization basically accords with the demand of generalization. As far as the magnitude of the task is concerned, computer instead of human finishes burdensome and repeated labors. In addition, the total time can reduce to one fourth of the original time or less. At the same time, the human-computer environment brings about a big advance in the performance accuracy.

References

1. Wu H H (1991) Cartography database. Beijing: Publishing House of Surveying and Mapping. (in Chinese)
2. Guo Q S (1998) Research on new theories and methods of automatic map generalization; [Ph. D dissertation]. Wuhan; Wuhan Technical University of Surveying and Mapping. (in Chinese)
3. Cai Z L (1999) Research on environment about interactive generalization on large-scale topographical map; [Master thesis]. Wuhan; Wuhan Technical University of Surveying and Mapping. (in Chinese)
4. Weibel R (1991) Amplified intelligence and rule-based systems. In: Barbara B, McMaster R eds, Map Generalization: Make Rules for Knowledge Representation. London: Longman Scientific & Technical.
5. Jones C B, Buady G L, Ware J M (1995) Map generalization with a triangulated data structure. Cartography and Geographical Information Systems, 22 (4): 317-331

(Continue from Page 9)

5. Molenaar M (1998) An introduction to the theory of spatial object modeling. London, Taylor & Francis.
6. Fuller R, Brown N (1996) Techniques for automatic generalization of the land cover map of Great Britain. International Journal of Geographic Information Systems, 10(8): 937-953
7. Muller J C, Zoschen W (1992) Area-patch generalization: a competitive approach. Cartographic Journal, 29: 137-144
8. Smaalen J W N (1996) A hierarchic rule model for geographic information abstraction. SDH, Delft, the Netherlands.
9. Ware J M, Jones C B (1996) A spatial model for detecting (and resolving) conflict caused by scale reduction. SDH, Delft, the Netherlands.
10. Richardson D E (1994) Generalization of spatial and thematic data using inheritance and classification and aggregation hierarchies. Spatial Data Handling Conference 94, Edinburgh, Scotland.
11. Wang Z S (2001) A Hybrid approach for automated area aggregation. CD of ICA Conference, Beijing.
12. Jaakkola O (1998) Multi-scale categorical database with automatic generalization transformation based on map algebra. Cartography and Geographic Information Systems, 25(4): 195-207
13. Jones C B, Buny G L, Ware J M (1995) Map generalization with a triangulated data structure. Cartographic and Geographic Information Systems, 22(4): 317-331
14. Weibel R, Dutton G H (1998) Constraint-based automated map generalization. The 8th International Symposium on Spatial Data Handling, Vancouver.
15. Bat P, Weibel R (1999) Using vector and raster-base techniques in categorical map generalization. The 3rd ICA Workshop on Progress in Automated Map Generalization, Ottawa.
16. Rodriguez M A, Egenhofer M J, Rugg R D (1999) Assessing semantic similarities among geospatial feature class definitions. Interop '99, Zurich, Switzerland.
17. Jones C B, Ware M J (1998) Nearest Neighbor search for linear and polygonal objects with constrained triangulations. The 8th International Symposium on Spatial Data Handling, Ottawa.
18. Tversky A (1997) Features of similarity. Psychological Review, 84(40), 327-352