DESIGN AND IMPLEMENTATION OF HIGH PRECISION MAP SYMBOL LIBRARY BASED ON GDI+

Wu Xiaofang¹, Du Qingyun², Cai Zhongliang³ and Xu Zhiyong⁴

School of Resources and Environment Science, Wuhan University, Wuhan, 430079, China, Tel: 027-87664557
¹somewxf@163.com, ²qydu@telecarto.com, ³zlcai@telecarto.com, ⁴XuSirZY@163.com

Abstract

The rapid development of computer and the emergence of digital map and electronic map is motivating Geographic Information System greatly. At the same time, with the progress of data capturing technologies, voluminous geographical data has been accumulated in databases and still increasing rapidly. We are on the verge a necessary-solved problem how to visualize the map data to the users in a direct, clear and visual way. In order to meet these requirements, map symbol library has appeared. As the symbolization of the vector map, the map features are transformed to the visual symbols that the users can read and extract more information. Therefore, symbolizing map features can communicate information more clearly and efficiently.

In this paper, the authors introduce open map symbol design, the oriented-object system design and implementation. The software is developed by Visual C++ 6.0 and has friendly interface, convenient operation and great function; and it uses the parametric data structure and high precision geometry data to organize the symbols and uses GDI+ to draw the symbols. The users can design the relevant symbols easily by using the program. And it meets with the needs of the national topographic map specification.

Firstly, the research and development situation of the map symbol library are summarized, the significances of design and implementation of map symbol library are discussed, then the total structure of the symbol library system and the different graphic elements are proposed with emphasis. Secondly, some functions and characteristic of the symbol library system are described, such as: Special data structure of the line symbol is to improve the drawing speed and have a good drawing effect. The feasibility of the symbol library system by applying the symbol library to symbolize the map is proved. Finally the author gives some conclusion and comments on the prospects of the application and development of the symbol library.

INTRODUCTION

The map symbol is the language of map, the major representation form, and the communication channel between the cartographer and users (Bertin, 1967). The map symbol library is the collection of map symbol, and taken for the visualized module of the electronic map or GIS system, and provides the functions of symbolizing spatial data (Koussoulakou and Kraak, 1992). Therefore, the design of map symbol library can play an important role in a digital map system.

The map is a kind of highly symbolized graphics. The map symbolization is the key step in the process of map making (Gong, 2001). Up to now, the research in map symbol library
almost lies in the drawing algorithm and structure design of symbol library, where there have been successful theory and practice for the drawing algorithm of symbol. There are two structures of symbol library: one is to use different functions to draw every different symbols, another is to use one function to draw all symbols based on symbol library (Cheng, 2001). The traditional means of designing map symbol is that parameters and functions are used to describe the symbols, where each certain symbol data is input and drawn by a function, which will form the symbolization software package finally. There is considerably limit in this method, for the symbolization software is bounded with certain system. With the changes of system, the symbolization software must take changes, which makes the design and drawing of new symbol more difficult. What’s more, the users still need to understand the usage and requirements of the functions, which is not very convenient.

At present, object-oriented method is often used to design symbol and manage the symbols in the database. The symbols are classified and saved in the database, and the management functions of symbol library are realized, such as querying, modifying, defining and deleting etc. In the symbol library, the symbol is made up by different kind of graphic elements, which is propitious to extend and modify symbols. Besides, it uses uniform ways to manage symbols and provides the uniform output interface, which is very convenient to the users (Yang et al., 2002).

GDI+ is the next-generation graphics device since the Windows 2000. Its major functions are to manage the display and printer, like the primary GDI. However, GDI+ is upgraded on the base of GDI. There are many functions such as the output of vector map, the output of raster map, the typesetting of text and so on. Comparing the traditional GDI, GDI+ has some new traits: gradual changing brush, sole path objects, matrix object, scalable regions and multi-format image support. Using GDI+ to draw map makes the graphics and texts much smoother, and the images will be sharper when changing their sizes.

In this paper, the authors introduce a high precision cartographic symbol library designed and implemented by oriented-object method. The software is developed using Visual C++ 6.0 and has friendly interface, convenient operation and great function; it uses the parametric data structure and high precision geometry data to organize the symbols and uses GDI+ to draw the symbols. The users can design the relevant symbols easily by using the software. And it meets the needs of the national topographic map specification.

**DESIGN OF HIGH PRECISION SYMBOL LIBRARY SYSTEM**

When designing the symbol library system, some factors should be taken into consideration such as demand analysis, application field, expandability and update of system and so on, which makes the data structure very important in the design of system. We must follow the rules of integrity, agility, precision, usability and opening (Zhao and Yin, 2002).

**Data structure of symbol library**

Designing a good symbol library must consider not only the display speed and output quality but also the maintenance of symbol library (Wang and Zhang, 2002), and the convenience in connecting with other GIS systems. Besides, we also need to consider the representation style and storing of the symbols. In order to query certain symbol in the symbol library rapidly, the symbols should be assigned with ID and indexed in the system. Use the only symbol ID to label all kinds of symbols, and the symbol ID must represent the
type and meanings of symbols. Building the index system has two ways: one is that the index and symbol data are stored in one file; another is that the index and symbol data are divided into different files (Cai and Li, 1999). The former has limit in symbol numbers. The latter has not such shortage, and it may use same file name and different file extension to structure the data file and index file, but it has difficulty in maintenance for there are often too many files.

The system uses the database structure that the index and data are stored in the same file. The shortage that the symbol number is limited exists certainly. But by updating symbol library and resetting the head index of symbol library, we can make up the shortage and meet users’ need. The data structure of symbol library is as in Figure 1.

![Figure 1: Data structure of symbol library.](image)

**Classification and structure of graphic elements**

In fact, though point, line and area symbol have their own characteristics, they have commonness. The difference is in the graphic elements, attribute and layout. Their drawing parameter and operating way are the same. Therefore, the map symbols can be abstracted and decomposed into some graphic elements: such as point, poly-line, Bezier line, ellipse, rectangle, polygon, text, image etc. These elements can be further grouped into three types: image, vector object and text. In order to reduce the redundancy of programming code, the base class graphic elements is constructed based on the image, vector object and text. This is very appropriate for the management and operation of symbol library and single symbol and the design the symbol. The structure of graphic elements is shown in Figure 2.

![Figure 2: Graphic classify and structure frame graph.](image)
High Precision of symbol library
We use parametric data structure to store the graphic objects in the symbol library. The parametric data includes length, width, interval, radius, angle, direction of the object and so on. The geometrical unit is millimeter, and the precise is as low as 0.01 millimeter in order to assure the output quality of the symbol. When saving the symbols, the non-dimensional unit is used to define the symbols to 1.0 * 1.0 unit in order to symbolize the different scale map conveniently. As to the different scale map, scale the symbol to adapt to the different map. It can assure the precision of map symbol and the reuse of symbol library.

In addition, the text is stored with Windows standard font and matrix, coordinates of which are transformed to realize the text transfiguration during the symbolization. The transfiguration and alignment of text can thus be achieved with high efficiency and precision (Wu and Xu, 2003).

IMPLEMENTATION OF HIGH PRECISION SYMBOL LIBRARY SYSTEM

Edit of symbol
Symbol is composed of a series of graphic elements according to certain rules. Therefore editing symbol is to edit the graphic elements of symbol, which includes modifying attribute data and geometrical data of graphic elements. The system provides the functions of editing graphic elements, such as: select, delete, copy, clear, move, undo, redo and so on. The most operations are based on the selected object. Here the implementation of undo and redo will be described in detail.

In any software, the function of undo and redo is always important, which makes the operation convenient and worryless when editing the symbols. The general idea of undo and redo is to make the operation of graphic element store in a list. When undoing or redoing, the data are extracted from the list, reset the graphic data. As for undo and redo, there are two methods.

1. Undo and redo irrelated to operation
The basic idea is that there are two lists: undo list and redo list. When the graphic data take changes, put the changed data into the undo list. When undoing, extract the second last data from the undo list, reset graphic data, and take out the last data from the undo list; put this data into the redo list. The advantage of this method is that the operation is simple and convenient. But the shortage is that more memory is needed for multiple storages of all changed data.

We propose the improvement of the method by adding a tag for deletion of the graphic data. Every time when the data change, only the IDs of all data and the tag for deletion are stored. Thus the graphic objects are saved only once in the memory. For example, when deleting a object, set the tag of the object to TRUE, and still leave it in the memory in order to undo or redo the operation. When moving the graphic object, add the new moved object and set its tag to FALSE, and set the tag of the original object TRUE, then save all IDs of objects into the undo list.

2. Undo and redo related to operation
Here undo and redo refer only to the currently operated objects. Therefore, save only the IDs of the operated objects into the undo list. When undoing, take out the last data from the undo list, switch the tag of deletion of objects in the data. At the same time, put the data in
the redo list, and then realize the operation of undo and redo. The idea is only related with the operation, and the record of the any operation in the lists is not needed. For example, when moving the object, make the tag of original graphic object TRUE, and set the tag of moved object to FALSE, and put the original object ID and moved object ID into the undo list. When undoing, take out the last objects from the undo list, switch the tags of the objects.

**Drawing of symbol**

Symbol is classified as point, line and area symbol. Point symbol can be drawn rapidly by the parameters: coordinates, angle and scale. As to the area symbol, positions of filling symbols are computed by the filling pattern and point symbol is placed at each position. The drawing of line symbol is relatively complicated. There are three ways: one is to draw each category of symbols by respective procedure, where it is not very easy to maintain the procedures. Another is to draw graphic elements along the basic line repeatedly to form all the line symbols. The third is to integrate the graphic elements with user-defined line type to form the line symbols, where the graphic element are drawn along the position line repeatedly or draw different lines to form the line symbol. The later two will cause problem at abrupt bend of line, which is often used in some complicated symbol (Zhao and Yin, 2002).

Our symbol library uses the drawing of user-defined line type and graphic elements to optimize the drawing of line symbol. For example, for the traditional drawing of railway symbol is to use the graphic elements of black rectangle and white rectangle to align repeatedly (as in Figure 3).

![Figure 3: Railway symbol and its element.](image)

However our symbol library introduces the user-defined line-type of OS, and uses the line-type to form part of the symbols as in Figure 4. When drawing railway, firstly one black solid, and then one white dashed line are drawn. The width of line and the interval of dashed can be reset. Compared with the traditional drawing way, the improved way draws symbols more rapidly and smoothly.

![Figure 4: Railway symbol and its line components.](image)

The former way stores the attribute and geometrical coordinates of graphic elements of one symbol, while the latter stores the parameters that describe the linear symbol, such as
width, color, linetype etc. Storing the parametric structure instead of coordinates can keep the symbol library much smaller.

When symbolizing the line by the former way, drawing the graphic elements along the line repeatedly need many transforms, which will cause the distortion of the graphic elements at the bend line. In the latter case, drawing linear symbol with user-defined line-type is fast and efficient. What’s more, If the symbolized line is Bezier line, the line need to be broken into poly-line and draw the symbol along the poly-line if using the former way, which will worsen the smoothness of line to a great extent.

Therefore, many line symbols such as road, railway can be drawn by the user-defined line. Of course, some complicated symbol which can’t be represented by the line-type can still be made up by drawing graphic elements along the line repeatedly.

Management and persistence of symbol library
The management of symbol library includes some operations such as save, delete, replace, modify, update, and copy between the symbol libraries. Considering the management and persistence of symbol library, we store the point symbol and line symbol in a symbol library, and set up the head indexes of point symbol and line symbol separately in order to query, delete and modify the symbol more conveniently.

When deleting a symbol, the symbol will not be deleted from the symbol library immediately, just set the ID of symbol negative, which is a tag for deleting the symbol logically. They will not be actually deleted until the symbol library is updated or rebuilt. The advantage is to avoid rebuilding the symbol library frequently and make the undo and redo operation easily. See Figure 5 below.

![Figure 5: Dialog of managing symbol base.](image)

Application Instance of Symbol Library System
As follows, cistern symbol and park symbol are designed by using the symbol library system, which shows the main interface and functions of system (See Figure 6). Furthermore, utilizing the symbol library to symbolize the map objects will improve the drawing efficiency and represent the abundant information.
CONCLUSION

In this paper, the object-oriented method used to build a high precision map symbol library is proposed. By the design of data structure, the abstraction and classification of graphic element and usage GDI+, the prototype software can meet the need of high precision output and print in GIS and digital cartographic system.

REFERENCES