Navigation Map Data Representation and Parallel Display Algorithm in an Embedded Environment

ZHANG Dong  WANG Ling  HUI Hongchao  LI Heyuan

Abstract  Map data display is the basic information representation mode under embedded real-time navigation. After a navigation display data set (NDIS_SET) with several dimensions and corresponding mathematical description formula are designed, a series of rules and algorithms are advanced to optimize embedded navigation data and promote data index and input efficiency. A new parallel display algorithm with navigation data named N_PDIS is then presented to adapt to limited embedded resources of computation and memory after a normal navigation data display algorithm named NDIS and related problems are analyzed. N_PDIS can synchronously create two preparative bitmaps by two parallel threads and switch one of them to screen automatically. Compared with NDIS, the results show that N_PDIS is more effective in improving display efficiency.

Keywords  embedded system; map representation; parallel display; vehicle navigation

CLC number  P208

Introduction

A mobile map, especially a navigation map, is mainly employed in embedded geographic information systems (GIS), which hold restrictive resources of computation, storage, screen display, communication and interface. Most cases of navigation application require high-demand real-time characteristics[1] that include data pre-process and access, screen bitmap producing and updating, guidance picture with zoom-in, location and navigation services, and re-programming after distraction from the selected route. Furthermore, navigation map display updating is always within 1 s, making time a pivotal metric to avoid possible computational trap and fluent display effects. Navigation map data display is the basic way to show information. The map presentation influences the user to perceive the route directly. Reasonable collocation, organization and effective display of elements are the basic requirements of navigation visualization. A whole display flow involves several aspects, i.e., huge amount of data, data organization, parallel access algorithm, symbolism configuration and integration, feature reduction, adaptive LOD model, and multi-types of data, such as index data, background data and topological data, which are multi-scaled to meet LOD application[2]. The map data display is the main information presentation mode in a navigation system, and the display effects directly influence the driver’s judgment for road situations. In this study, a navigation data parallel display, N_PDIS, is advanced based on data representation technology, which delivers an excellent impact on display speed and integrated efficiency.

Received on November 7, 2007.

ZHANG Dong, Xi’an Research Institute of Surveying and Mapping, 1 Middle Yanta Road, Xi’an 710054, China.
E-mail: navgrid@163.com
1 Navigation data representation

1.1 Navigation data set

There are major differences between the navigation map and the common map, in terms of the element, content, data organization and other factors[3]. Navigation map data includes a high number of features, and deciding to accept or reject features is difficult. A display data set, being a multi-dimension data set, is defined as follows.

Definition 1 \( \forall t \), there is a multi-dimension data set BitMap \((P, T, S)\): 

\[
\text{ScreenMap} (T, S) \subseteq \text{BitMap}(P, T, S) \quad (1)
\]

where \( P \) is the pointer to BitMap in memory; \( T \) denotes time; and \( S \) denotes display scales.

Definition 2 Bitmap contents:

\[
\text{BitMap}(P, T, S) = F_1 \ (\text{NDIS_SET}, \text{Route}) \quad (2)
\]

where the BitMap \((P, T, S)\) is a multi-dimension set, which changes according to \( T \) and \( S \); NDIS_SET denotes the map data contents of the bitmap in memory at the time \( t \); Route denotes the selected route; and \( F_1 \) denotes the rules to produce bitmap.

Definition 3 Display content set on screen:

\[
\text{ScreenMap} (T, S) = F_2 \ (\text{NDIS_SET}, P, C) \quad (3)
\]

where the ScreenMap denotes display content set on screen at the time \( t \), which is a subclass of BitMap \((P, T, S)\), and changes in terms of \( T \) and \( S \); \( F_2 \) is the symbolism expression rule; and \( C \) denotes color setting model of all map features.

Definition 4 Feature contents:

\[
\text{NDIS_SET} (T, S) = \{ \text{Road, Background, Assis-Info, R, V} \} \quad (4)
\]

where the set items express the Road net, display Background, and Assistant information, respectively. However, \( R \) means current selected navigation route, while \( V \) is current vehicle.

1.2 Mathematical representation of display contents

NDIS_SET is a data set of real-time display contents, which needs to be transferred as mathematical description and representation[4].

Definition 5 \( \forall L \), Road \((T, S)\) = 

\[
N \{ \text{Node, Arc, Attributes, } \cdots \} \quad (5)
\]

where \( N \) denotes the road net; Node expresses the node or associated cross with the road; and Arc expresses the road segment and is provided with direction and connection between nodes mostly. In an urban area, crosses have more complicated attributes – they generally include the simple cross, complicated cross and crossroad. Attributes express items such as the road grade, traffic restriction conditions, road length and surface, and instances of traffic jam.

Definition 6 Background \((T, S) = \{ \text{drainage, vegetation, annotation, residential area, borderline of regionalism, train, interest-points, etc.} \} \), is changed along with scales and expressed as a region or point. Interest-points are used to choose a target for navigation, shown as: interest-points \((S) = \{ \text{public establishment, restaurant, hotel, gas station, emporium, cultural places, traffic establishment, etc.} \} \), is expressed as attribute records.

Definition 7 Assis-Info \((T) = \{ \text{time, scale, north point, turn direction, distance, left time, speed} \} \), every item is defined as an integer, double or float.

Definition 8 Bool ModeSet \((T) = \{ D - N, N - V \} \), denotes the display pattern, where \( D - N \) is the mode corresponding to period of time, 1 for daytime, and 0 for night; \( N - V \) is the display mode, 1 for north to go up on screen, and 0 for vehicle forward to go up.

2 Data preprocessing

Given the demands of real-time and dynamic characteristics, the navigation display data must be pre-processed according to some special rules[5], which can modify the display efficiency, promote the read data speed and reduce redundancy data operations.

1) Map tiling. In an embedded navigator with limited resources, display data must be organized into blocks[6]. There are two methods for partitioning the map into a series of blocks: ① the user-defined block, where an area is partitioned into 16 or more blocks and corresponding display data blocks are constructed to reduce data quantity in one access. However, it is always difficult to determine the optimal block size in actual application. ② According to the scales of the map, for example, we can organize the display data according to 1:10 000 navigation map, with less work in map data preprocessing. This allows more data
blocks convenient for data access, although data management efficiency is lower.

2) Classification. Classification is another necessary approach in data organization. All display data need to be classified to reduce data quantity and increase speed. For simplifying data organization, road display classification should correspond to real road classes. For example, the roads are always classified as six levels, i.e., freeway, national highway, provincial way, county way, country road, and town road according to geography conditions, area, and means of transportation.

3) Index construction. Since data access must be implemented with real-time, dynamic and multi-time modes based on little blocks, establishment and application of a two-level dynamic index is an effective approach. ① Taking a 1:25 000 sheet as a unit to index and dispatch. ② Taking a region as a unit to set up a file, every record of the file is a sheet. ③ The first-level index file can be used to search the second-level. ④ The second-level index can be used to get the position of map, with each record having 12 bytes, including map ID, start position of the record, record length, and whether it has logical blocks or not.

4) Object cut algorithm. Compare all map data and parts in a display buffer, and then remove extra map data. The method is: ① looking for a dynamic balance point between data read frequency and data capacity, and avoiding excessively large data quantity; ② use Layer to organize and store data, and eliminate redundant data.

5) Content configuration algorithm. To solve the problem of loading size caused by putting fixed data into memory, classify and accept or reject the spatial data according to their different scales, and organize and display them according to layer model. The bigger the map scale is, the more specific the spatial data are.

3 Navigation data display algorithm-NDIS

The basic idea of NDIS: according to Definition 1, the main task of display is to establish a BitMap at time t and ScreenSet (P, T, S), then display the contents in terms of current vehicle position P(X, Y) and the screen center (SX, SY). This means that the NDIS is a serial concept, not a parallel one.

The NDIS algorithm is a method that can be used from calculation, reading map data, and creating a bitmap to switching to screen according to the current position P, and repeating while the position changed. This is easy to understand and realize, but the problem is clear too. It will cost extra time to read repetitive map data in memory. The optimized algorithm should use current data in memory repeatedly and input only the new data. The optimized algorithm is shown as Fig.1.

Suppose the cache and the map scope are of the same size. When the map data wanders, the background process will calculate the wandering direction in advance vaguely; splice, organize, and update the map contents; then reset the cache. When the wandering direction is right and the screen center exceeds the cache center, the background process will eliminate the contents of cache 1 and cache 2 at first, and copy cache 3 to cache 1, cache 4 to cache 2, and then draw the two adjacent scopes on the right to cache 3 and cache 4. All these steps build a new bitmap and complete the map wandering smoothly.

4 Navigation data parallel display algorithm-N_PDIS

NDIS has optimized read data mechanism, but it is also single-thread. The N_PDIS is constructed under parallel idea and has the following main approaches.

1) There are always two bitmaps in memory at the same time, with each sized as 4×4 map sheets at 1:25 000 scale, *pp1 and *pp2 point to them respectively. BitmapInUse represents the current bitmap being used, and the other one being updated.

2) The object cut algorithm and content configuration algorithm are used for building bitmap.

3) The bitmap dispatch is controlled by trigger and dispatch feature. \( W_i, W_j \) is the center region control
ler of the two bitmaps, when the screen center exceeds one region, the thread will dispatch another bitmap. When blocks of map data need to be read, the thread will compare the IDs with the sheets in memory and only read the new maps.

4) The operation, eliminating map data from memory, is processed according to the strategy of the farthest distance. The bitmaps switch is controlled by trigger feature \( W_2 \). When the screen exceeds the scope of the current bitmap, \( W_2 \) will inspire the other bitmap read for use by another thread, and switch it to screen.

The relative parameters are defined in Table 1.

| No. | Variable name                | Meanings                                      |
|-----|-----------------------------|----------------------------------------------|
| 1   | ScreenWidth                 | Screen size                                  |
| 2   | BitMapWidth                 | Size of the memory for display               |
| 3   | p_Mem                       | Spatial pointer (16 sheets)                  |
| 4   | pp1                         | Pointer of bitmap 1                          |
| 5   | pp2                         | Pointer of bitmap 2                          |
| 6   | BitmapInUse                 | Identification of current bitmap             |
| 7   | MapListNumber1 [16]         | Pointer 1 to the used maps                   |
| 8   | MapListNumber2 [16]         | Pointer 2 to the used maps                   |
| 9   | MapIDinDisplay [16]         | Identification of the used maps (16 sheets)  |
| 10  | MapIDforDisplay [16]        | Map ID prepared to enter(16 sheets)          |
| 11  | MapdisLabel                 | Map ID input in real time                    |

5  Experiment and conclusions

In Nav-01, navigation data visualization has been realized according to the data organization and algorithm mentioned in this study. The right of the screen becomes the left picture of the next cross. The assistant status bar displays other information, including turn left after 100 m, route length is 3.2 km, distance to destination is 0.9 km, D-N is 1, and GPS is ready.

The navigation experiment shows: ① the NDIS_SET describes the contents on the screen clearly at time \( t \), and object selection closely combined with practical applications; ② establishment of the index mechanism reduces excessive dependence on memory. Data expansion has little impact on code and memory, map search speed is greatly improved, input time for all data needed at current location to memory is less than 5 s \((S=200 \text{ m})\) when the navigator turns on. ③ Results from comparing the time of establishing a new bitmap between N_PDIS and NDIS, when the screen center point exceeds the bitmap region, are shown as Fig.3. Supported by multi-threads parallel mechanism, N_PDIS is better than NDIS in terms of time cost.

References

[1] Zhang Dong, Guo Jianzhong, Liu Ailong, et al.(2005) Multi-topological producing and real time route programming [J]. Cartography Science and Engineer, 25 (2): 13-17 (in Chinese)

[2] Parkinson B W, Jr Spilker J J(1996) Global positioning system theory and applications[M]. Washington D C: American Institute of Aeronautics and Astronautics

[3] Zhang Dong, Liu Ailong, Zhang De(2005) The present situation of vehicle navigation techniques in China and the multi-topological formation and route computing of national navigation map[C]. The 22td International Cartographic Conference, Spain

[4] May A J, Ross T, Bayer S H(2003) Driver’s information requirements when navigation in an urban environment[J]. Journal of Navigation, 56: 89-100

[5] Yan Hanbing, Liu Yingchun(2002) A new algorithm for finding shortcut in a city’s road net based on GIS technology[J]. Chinese Journal of Computers, 23(2): 210-225 (in Chinese)

[6] Guo Jianzhong (2005) Mass data display on multi-scales [J]. Journal of Cartography Institute, 22(2): 136-138 (in Chinese)