Research on Offline Electronic Map Application Framework of Mobile Terminal

Li Feng, Ming Jing, Wang Changhan (Chongqing Survey Institute, Chongqing 400020, China)

Abstract: With the widespread use of mobile devices such as iOS and Android, the use of electronic maps for mobile devices is becoming more and more popular. For the current situation of offline map-based electronic map application, this paper designs an ArcGIS-based mobile terminal offline electronic map application framework, constructs an offline electronic map API, and implements tile data encryption packaging and synchronization tools. Experiments have proved that this paper has achieved tile data security organization and efficient access, and has solved the problems such as rapid construction, data integration, security, and update of mobile terminal offline electronic map applications.

**Keywords :**mobile terminal; offline caching; electronic map

[CLC number] P208 [Document code] A [Article ID] 1009-2307 (2013) 06-0129-03 1

### 1 Introduction

With the development of mobile terminal devices, mobile communication technologies, wireless Internet technologies, and the widespread application of GIS in social life, electronic map applications for mobile terminals are becoming more and more popular [1-2]. The electronic maps of mobile terminals are divided into two modes based on data acquisition and deployment. One is online data mode and the other is offline data mode [3]. The former uses the wireless network to transmit map data. Due to the limitations of signal coverage and transmission speed, it is generally suitable for real-time data acquisition and real-time updating; the latter deploys map data to mobile terminals along with the program, and can adapt to no network or network stability. In the case of poor performance, it has the advantages of fast data access and timely system response. According to the hardware and software of the mobile terminal, the implementation of the electronic map of the mobile terminal also has its own characteristics. Tang Yingxue and others achieved the reading and drawing of the electronic map of the PDA platform[4]; Xu Hualong studied the electronic map system based on Windows CE[5]; Wang Xiaojun et al. explored the method of cutting and updating tiles based on ArcGIS[6-8] ]; Yu Ying et al. studied the offline map browsing method based on ArcGIS and Flex [9], and realized the storage of map tile data in a folder hierarchy. In the existing method for implementing an electronic map of a mobile terminal, there are the following problems: 1 There is no set of solutions that cover data integration into data access; 2 The data packaging of mobile terminals is not supported, and a large number of small files cause data deployment to be slow. Poor maintenance; ArcGIS 9 only provides data file decentralization, ArcGIS 10 provides two methods of packaging and decentralization, but these data storage methods can only be applied to the server [10]; 3 does not support tile data encryption, data Poor security. To solve these problems, this paper studies and designs an offline electronic map application framework for mobile terminals.

### 2 Application Framework Design

The design goals of the offline electronic map application framework of the mobile terminal are as follows: 1 Quickly construct a highly reliable map application and support offline map tile data access; 2 It can manage map tile data, including operations such as generation, encryption, packing, and updating.

**2. 1 overall architecture**

The main design idea of ​​offline electronic map application framework for mobile terminals is to expand the design of offline tile layer based on ArcGIS runtime, construct an offline tile map application framework, and focus on the core issues of generating, encrypting, packaging, and updating tile data. The tile data management API is implemented. The overall architecture of the mobile terminal offline map application is shown in Figure 1. Figure 1 Overall Architecture The mobile terminal hardware platform refers to various smartphone and tablet platforms running iOS, Android, and Windows Phone systems. Based on the hardware platform, ArcGIS publishes the corresponding SDK interface. This article designs the mobile terminal electronic map API on the mobile terminal's ArcGIS SDK, and then implements the tile data management API and the tile map application framework, and binds the data access layer, the core logic layer, and the view presentation layer provided by ArcGIS. This provides a quick and efficient solution for offline electronic maps.

**2. 2 Structural Design**

According to the view controller mode, the offline electronic map application framework of the mobile terminal is divided into three parts: view layer, core logic layer, and data access layer. The application framework design is shown in Figure 2. Figure 2 Structure Design Diagram When the system is initialized, the UI view is initialized and the offline tile layer is added. The offline tile layer invokes the corresponding configuration file parsing interface to complete the parsing of the tile metadata. After the offline tile layer is initialized, ArcGIS automatically performs asynchronous tile request and loading according to the current map browsing level and scope. The implementation mechanism is to create an offline tile request operation, then add it to the operation queue, and then asynchronously by ArcGIS. carried out. During data acquisition, the display front-end requests multi-threaded scheduling of corresponding tile data according to the current level and position. The core logic of the dispatch thread is to parse the tile URL, obtain the tile set name and 3 fields of the level, row, and column, and then call the tile cache read API to access the encrypted and packaged tile data, and the result returns the corresponding The memory data block is then parsed into a drawable tile image through the image read interface and submitted to the drawing front-end electronic map view layer for use.

### **3 Tile Cache Database Integration and Access**

**3.1 tile cache generation method**

The e-map tile cache contains cache map pictures of different levels of collection, as well as the cache configuration file conf.xml. The tile cache directory structure is organized in three levels: level, row, and column. There are two methods for generating it. The first one uses ArcGIS Server's map cache function, and the second is a custom tile-cutting interface and tools. After the map service is published, ArcGIS Server can convert the map data in the database into static images of different levels according to the specified cache progression and store it on the Web server. The client gets static tiles from the cache instead of dynamically rendered map services.

The main steps to generate a map cache are:

1 Use ArcMap to edit the map document;

2 Use ArcGIS Server to publish map services;

3 Use ArcCatalog to access the map service and generate a tile cache.

Using custom tile cutting interface and tool steps include:

1 grid data and related parameter preparation;

2 generate tile metadata description information;

3 Use third-party raster image library to achieve automatic tile cutting.

**3.2 tile cache database schema design**

After the electronic map tile cache is generated, the default is organized according to the level scale-row-column three-level structure. Each tile is 256 × 256 size, the file format is JPG/PNG, and the length is generally less than 16KB. There are two problems with massive tile data: 1 The files are stored in plaintext, and the data security is poor; 2 The files are numerous and fragmented; the deployment of tile data packages is complex and maintainability is poor; ArcGIS 10 supports packaged tile caching but Insufficient support for tile cache encryption. Therefore, this paper designs a tile cache encryption and packing mechanism based on Sqilte embedded database and RC4 encryption algorithm to solve the problem of tile cache encryption and packetization. The tile database schema design is shown in Figure 3. Figure 3 Database schema design diagram Mobile terminal offline electronic map packaging data supports the storage of tile cache metadata. The main metadata fields include: 1 tile cache name; 2 corresponding tile data table name; 3 tile cache range; 4 Tile raster image information; 5 tile cache space reference information; 6 tile cache levels LOD configuration. The packaging scheme supports multiple sets of map tiles, each corresponding to a tile data table. The tile data table uses the BLOB type to store the corresponding tile data, and the tile data is jointly indexed by 3 fields: level, row, and column. This article selects Sqlite as a binary tile data container. The reasons are: 1Sqlite is a database that the mobile terminal comes with, and deployment is simple; 2Sqlite provides a standard SQL access interface, simple and efficient; 3PC and mobile terminals support Sqlite, for data deployment and update The problem, Sqlite storage program has prominent advantages. For the management of electronic map tile data, the application framework provides management tools to provide data tile generation, binary data encryption, data package integration, data update and other operations in a graphical manner, and supports ArcGIS's hierarchical folder storage. Distribute tile data integration and updates.

**3.3 Tile Cache Database Management and Access**

The API designs a tile cache database access API based on the offline electronic map tile data pattern of the mobile terminal. The API encapsulates the functions of tile cache metadata information, read interface and implementation, management interface and implementation, supports the opening and closing of data packages, metadata acquisition, and tile data acquisition; supports obtaining a list of cache names, creating a cache, Delete cache, update cache metadata and tile data at all levels. In terms of implementation mechanisms, the same API interface is implemented using different platforms and languages. Under the iOS platform, based on the C + + language implementation, mixed compilation with Objective-C supported by iOS itself; based on the Java language implementation under the Android platform. In the deployment mechanism, the dll and jar files of the access interface are deployed with the electronic map application; however, the dll files of the map tile management interface are not released together with the electronic map application, and are mainly used in the data production integration. Taken together, electronic map tile generation, packaged encryption integration, data deployment, and data offline access are combined to provide an offline electronic map solution for mobile terminals in the form of an application framework.

### **4 Experiments**

After designing and implementing an offline electronic map application framework for mobile terminals, this article has successfully implemented an electronic map application based on iPad and Samsung Android mobile terminals. For this application framework, relevant performance tests were conducted in this paper. (hardware environment: 1 data integrator PC: Intel dual-core E4500, 2. 2GHz, memory 2GB; 2iOS mobile terminal: iPad2 platform; 3Android mobile terminal: Samsung Galaxy Tab P6800, 16GB, dual-core 1. 4GHz)

**4. 1 Data Encryption Packing Experiment**

The data encryption and packing experiment mainly tests the scatter files and disk occupation and deployment time after encryption and packaging. Using a scale of 1:40 maps in Chongqing, a total of 39, a total of three levels, the total file size of 278MB, the test results are shown in Table 1. Table 1 Data Encryption Packing Experiment Results ArcGIS Scatter Tiles Method Data Improvement File Number 32975 1 Encrypted Package Improves Data Security Disk Usage 341MB 289 MB 15% Reduction Deployment Time 15 Min. 17 Seconds 1 Min. 4 Seconds Lower 93% from Table 1 It can be seen that the Sqlite-based encryption and packaging mode stores the cached images in a cache database, which significantly reduces the number of small files, which is less than the disk space occupied by the traditional loose cache, and reduces the waste of disk; at the same time, the data packet can be Quickly copy/move/delete caches between computers and mobile devices, dramatically saving data deployment time and facilitating the replication of caches between transitional environments and production environments. At the same time, the data is encrypted with the RC4 algorithm and stored in the BLOB field of Sqlite. Compared with the scattered tile files stored in the original plaintext, the security is improved.

**4. 2 Data Access Experiments**

The data access experiment mainly tests the access control of scattered files and encrypted packaged data. Using a scale of 1:4000 on a certain scale in Chongqing, there are 39 maps with a total of 3 levels. The total file size is 278MB. The test results are shown in Table 2. Table 2 Data Access Experiment Results ArcGIS Scattered Tiles (ms) This method (ms) Performance improvement accesses 1 tile 1. 24 0. 77 Time reduction 38% Access 6 tiles (1 row) 5. 67 1. 96 65% reduction in time access to 24 tiles (1 level) 21. 48 6. 65 Time reduction by 69% Access to 955 tiles (1 cache set) 718. 33 161. 44 Time reduction by 77% Seen from Table 2 The Sqlite-based encryption and packaging model stores cached images in a cache database. Based on the memory page cache, the number of disk file accesses is significantly reduced. Compared with the traditional loose cache files, the number of accessed tiles increases the number of data. The more significant the increase in access performance. The main reasons are as follows: 1 Cache package files save the number of disk accesses and reduce the overhead of a large number of trivial file accesses. 2 The Sqlite database supports data access caching and improves data access performance.

### **5 Conclusion**

For the current status of offline map-based electronic map applications, this paper designed an ArcGIS-based application framework, constructed an offline electronic map API, and implemented a tile data encryption package and synchronization tool. This article has been applied to the actual off-line cached electronic map project, which shows the following advantages: 1 The application framework can quickly develop offline tile cache electronic map applications, support e-map browsing, POI query and other common functions; 2 custom tile data Encrypted packing format improves data security. Compared with distributed massive tile files, it improves disk utilization and reduces data deployment time. 3 Make full use of the embedded SQLite database of mobile terminals to improve map tile data. Access performance. The further development direction of the application framework is to support multiple encryption algorithms, further support and improve the offline electronic map related functions of mobile terminals, and reduce the complexity of application development and data integration.