GIS Data Compression Using Spatial Compression Approach

Phyo Phyo Wai
University of Computer Studies Mandalay
Phyophyowai84@gmail.com

ABSTRACT

Geographical Information System (GIS) is rapidly developing and then embedded technology, wireless communication technology, mobile GIS are becomes an active research area in the field of GIS. The purpose of this paper is to get best solution of GIS data compression for information collection. This paper mentions the spatial data compression approach. Compression techniques are very important for GIS. Spatial data contains points, lines, polygons, latitude and longitude. In Global Positioning System (GPS), spatial data are getting from satellite. And satellite images are very large size. So, it needs to compress these images for easy to use by handle devices like mobile phone.

Keywords: Compression techniques, Embedded GIS, GPS.

1. INTRODUCTION

Computer has moving fast into the mobile times. With the support of hardware, operation systems, GPS application and network technology, the access real-time location information, visit information and seamless combine the mobile object with the large geographic spatial database by wireless network have become the reality. Embedded GIS is the combination of embedded technology and GIS, which is the extension, complementation and development of the former GIS [3].

The embedded system is a typical kind of environment of pervasive computing. Embedded GIS is the combination of embedded technology and GIS. It's an ideal solution for navigation, position, map query and spatial data integration which can be used in various areas. Under the embedded environment, the storage resource and the computing resource are finite, the system computing ability is very low. The on-vehicle EGIS has a large amount of map data and is demanded for high real time performance. GSM/GPRS network is available anytime and almost anywhere. It can support the control center and the navigation systems on vehicles. GPS (Global Positioning System) calculates the current position by the free satellite data received passively [4].

This paper presents the system architecture of the mobile GIS for information collection and proposes the detail design idea of some most important modules and data compression. The rest of this paper is organized as follows. Section 2 shows the related work for this paper. In section 3, we briefly describe some background theories about system architecture, spatial data management, compression techniques, spatial data visualization, and spatial data transmission. The paper is concluded in section 4.

2. RELATED WORK

GIS data is different from audio, visual, of text/binary data. First, GIS data has a layer structure. It often represents lines, areas or both. These GIS data consist of multiple geographic points or locations. Each point data consists of latitude, longitude, and optionally altitude, time, or other parameters. Second, any geographic point and the next point in GIS data are probably 'close' to each other and stay in the local region, if data represents routes or areas. However, data scale, latitude and longitude, is available globally. Thus, sets of data cause large redundancy that can be compressed easily. Even single usage of differential-based compression method may work fine. Third, GIS data require 'Lossless' compression and 'Lossy' compression, depending on the situation and purposes. In other words, what you need is to pinpoint an exact location, route, or shape of path/area. 'Lossy' compression may fit when the 'route' is more important than each of the points it consists of [2].

S.Ching Chen, X. Wang, N. Rishe, M. Allen Weiss were designed the system architecture utilized the existing resources to achieve maximum performance by using the “Internally Distributed Multithreading (IDMT)” technique. The spatial access method, semantic R-trees, was used to search for an object, based on both spatial and semantic information. System performance results were presented and analyzed. Reducing network traffic to achieve faster response to users was also discussed [1].

3. THEORITICAL BACKGROUND

3.1. Spatial Data Management

In this paper, it use of spatial data management. The core of spatial data management is to manage the map layer and its spatial entity's data (such as points, lines, and polygons). An object-oriented spatial dataset can be composed of different map layers, including point layers, line layers, polygon layers and image layers. And these layers store the corresponding spatial entities respectively. The actions of map layers include appending, inserting, deleting, editing, opening, closing etc., and the actions of spatial entity include appending, updating, deleting, querying etc. Figure 1 shows the object-oriented management of spatial data.

MapLayer is the most important basic class of map layer management, which has the whole basic information of the map layer, such as type (point, line, polygon, and image), state (opening, closing, editing, deleting), name, digital map's range, maximal and minimal display scale. And it also has the common actions of all kinds of layers, such as opening, closing, saving, deleting and appending layers, fetching the attribute database, updating the basic information etc. ArcLayer inherits from MapLayer, at the same time it is the basic class of LinLayer, RegLayer. PntLayer, LinLayer, RegLayer and ImgLayer control various kinds of actions of the point, line, polygon and image data respectively.

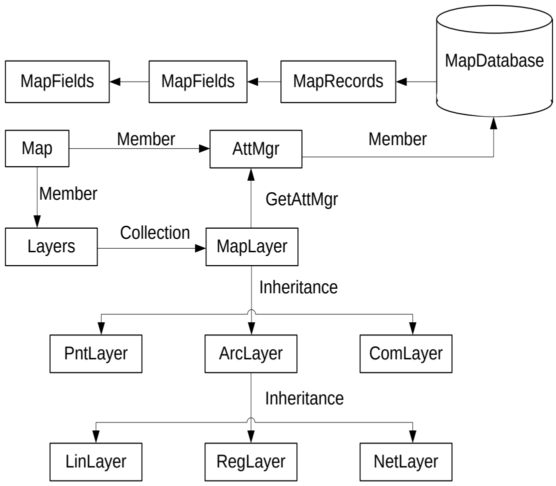

Figure 1. Spatial data management chart

In mobile terminal devices, spatial data is more desirable because of their small size. In general, the storage capacity and memory of the mobile terminal are all very small, and the bandwidth of data transmission of wireless network is limited, but the spatial data are larger, so must choose the suitable storage structure and storage tactics, and carry on rational compression to the spatial data. Choosing proper data structure, trying one's best to save the memory space on the premise of keeping the exactness and efficiency. Removing the surplus subsidiary data, selecting suitable index way for spatial data, and adopting compression algorithm to compress the spatial data and save the memory space, which is an effective way to reduce the memory space [5].

3.2. Compression Technique

Compression techniques for spatial data can allow mobile terminal devices to carry larger subsets of vector maps or free up memory for other datasets [6]. Compression techniques for spatial data can also reduce the communication cost of downloading new maps to the mobile terminal devices over low-bandwidth wireless network. The goal of spatial data compression is a compact representation of map data.

The goal of spatial data compression is a compact representation of map data. Spatial data compression is divided into loss compression and lossless compression. Lossy compression also can be divided to the compression based on statistical model and the compression based on dictionary. The classical Douglas-Peucker algorithm is recognized as the one that delivers the best perceptual representations of the original lines. It is used extensively for both computer graphics and GIS. There are two variants of this algorithm, the original O(nm) method, where n denotes the number of input vertices and m the number of output segments, that works in any dimension, and the O(n log n) one, which only works for simple 2D planar poly lines.

In order to handle spatial data efficiently, a database system needs an index mechanism that will help it retrieve data items quickly according to their spatial locations. Spatial data objects often cover areas in multidimensional space and are not well represented by point locations. It is important to be able to retrieve objects efficiently according to their spatial location. An index based on objects' spatial locations is desirable, but classical one-dimensional database indexing structures are not appropriate to multi-dimensional spatial searching. Structures based on exact matching of values, such as hash tables, are not useful because a range search is required [5].

Compression methods are divided into two classes, lossless or lossy.

3.2.1. Loseless compression

Lossless data compression makes use of data compression algorithms that allows the exact original data to be reconstructed from the compressed data. Lossless data compression is used in many applications

Lossless compression is used when it is important that the original and the decompressed data be identical, or when no assumption can be made on whether certain deviation is uncritical.

Lossless compression methods may be categorized according to the type of data they are designed to compress. Some main types of targets for compression algorithms are text, executables, images, and sound.

Most lossless compression programs use two different kinds of algorithms: one which generates a statistical model for the input data, and another which maps the input data to bit strings using this model in such a way that "probable" (e.g. frequently encountered) data will produce shorter output than "improbable" data.

Statistical modelling algorithms for text (or text-like binary data such as executables) include:

- Burrows-Wheeler transform (BWT; block sorting preprocessing that makes compression more efficient)
- LZ77 (used by Deflate)
- LZW
- PPM

Encoding algorithms to produce bit sequences are:

- Huffman coding (also used by Deflate)
- Arithmetic coding [8].

3.2.2. Lossy compression

A lossy compression represents an irreversible operation. You never can get back an uncompressed image exactly identical to the original one, because some kind of information suppression will be introduced anyway during the compression process. Typically, lossy compression algorithms can squeeze your images more much better than loseless algorithms does, but at the cost of some irreversible information suppression (and quality degradation).

There are two basic lossy compression schemes:

- In lossy transform codecs, samples of picture or sound are taken, chopped into small segments, transformed into a new basis space, and quantized. The resulting quantized values are then entropy coded.

- In lossy predictive codecs, previous and/or subsequent decoded data is used to predict the current sound sample or image frame. The error between the predicted data and the real data, together with any extra information needed to reproduce the prediction, is then quantized and coded.

In the systems the two techniques are combined, with transform codecs being used to compress the error signals generated by the predictive stage and use Douglas-Peucker Algorithm.

3.2.2.1 Douglas-Peucker Algorithm

The Douglas-Peucker algorithm for curve simplification is illustrated on Figure 2. In this method, given a polygonal curve, the Douglas - Peucker algorithm computes a subset of DPs standing for the simplified curve. The selected DPs have the important property of being perceptually attractive. In this algorithm, a DP is computed as the furthest point, say 1(Figure 2), from the given curve endpoints, say M and N. Further DPs are computed by recursively applying the DP definition for the two resulting sub-segments: [M . . . 1] and [1 . . . N]. In this algorithm, the end of process criterion is a threshold, say *td*, on the Euclidian distance *di*. This curve simplification step delivers a curve skeleton with fewer points to process [7].

Algorithm 1: CurveSimp($C_{A;B}; J; G;D$)

Input: *J*, *G* and *D* were the tolerance band of James, Light Column and Douglas - Peucker Algorithm, $C_{A;B}$ is the curve needed to be simplified.
Output: $\acute{P}$ is the set of dominant points after compression, M is the number of the points.

$\acute{P} \leftarrow Ø \cup P_1$, $i \longleftarrow 1$, $k \longleftarrow 2$, $flag \longleftarrow false$;
while $P_k \in C_{A;B}$ **do**
  if $\alpha_k \geq J$ **then**
    $\not{\acute{P}} \longleftarrow \acute{P} [P_k$;
  else
    LCA(P_i, P_k, G);
    if $flag \neq false$ **then**
    $a_1 \longleftarrow b_1$, $a_2 \leftarrow b_2$;
    end
    if P_{k+1} 2 $LCA(P_i; P_k;G)$ **then**
      $P_k \leftarrow P_i, i \leftarrow i + 1, flag \leftarrow true$;
    else
      $\acute{P} \leftarrow \acute{P} [P_k$, $flag \leftarrow false$;
    end
  end
  $k \leftarrow k + 1$;
end
if $P_1 == P_N$ **then**
  CurveSimp($C_{A;C}; J;G;D$);
  CurveSimp($C_{C;B}; J; G;D$);
end
$M \leftarrow |\acute{P}|$;
DP($C_{A;B}$, D);
return ($\acute{P}$, M);

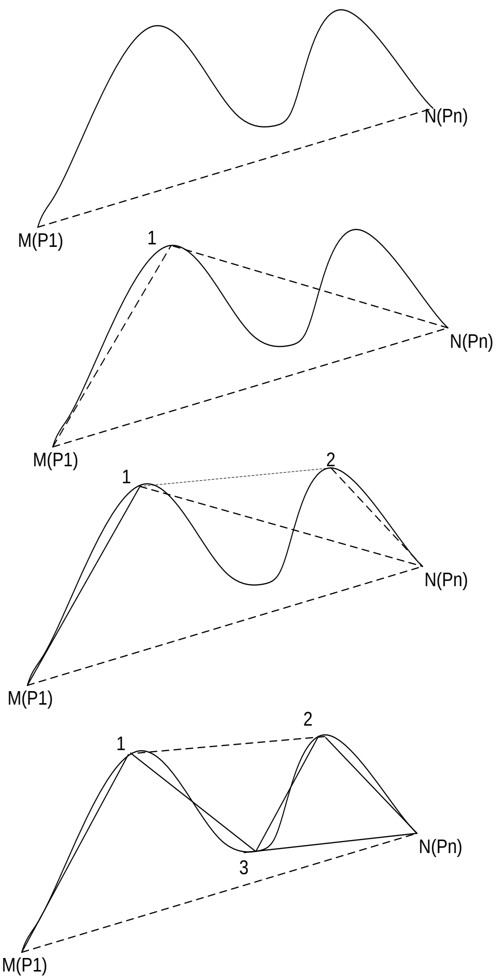

Figure 2. Douglas – Peucker Algorithm for Curve Simplification

The advantage of lossy methods over lossless methods is that in some cases a lossy method can produce a much smaller compressed file than any known lossless method, while still meeting the requirements of the application.

The compression ratio (that is, the size of the compressed file compared to that of the uncompressed file) of lossy video codecs are nearly always far superior to those of the audio and still-image equivalents.

Lossless compression algorithms usually exploit statistical redundancy in such a way as to represent the sender's data more concisely, but nevertheless perfectly. Lossless compression is possible because most real-world data has statistical redundancy.

Another kind of compression, called lossy data compression, is possible if some loss of fidelity is acceptable.

Lossless compression schemes are reversible so that the original data can be reconstructed, while lossy schemes accept some loss of data in order to achieve higher compression. However, lossless data compression algorithms will always fail to compress some files; indeed, any compression algorithm will necessarily fail to compress any data containing no discernible patterns. Attempts to compress data that has been compressed already will therefore usually result in an expansion, as will attempts to compress encrypted data.

In practice, lossy data compression will also come to a point where compressing again does not work, although an extremely lossy algorithm, which for example always removes the last byte of a file, will always compress a file up to the point where it is empty [10].

3.3. Spatial Data Transmission

Wireless network communication is the foundation of the wireless transmission for spatial data. The data that should be transmitted include two kinds: simple data and complicated data. Simple spatial data transmission means transmit a small amount of positioning data or brief text information. Complicated spatial data transmission means transmit the vector or raster data of the spatial information in real time [9].

4. CONCLUSION

With the development of GIS technology, wireless positioning technology, such as GPS, of mobile terminals, wireless transmission technology of spatial information and mobile representation technology of spatial information, there is a new opportunity for the development of mobile spatial information services.

Mobile spatial information services facilitate users with different resolutions, different spatial resources, real-time dynamically change, and large amount of spatial information.

This paper presents the system architecture of the mobile GIS for information collection and compression algorithm for spatial data management.

REFERENCES

[1] S. Ching Chen, X. Wang, N. Rishe, M. Allen Weiss, *"A web-based spatial data access system using semantic R-trees"*, Information Sciences. 2003, 7

[9] C. HO Lee, L. Chen, J. Dong Lee, and H. Young Bae, "Content Adaptation and Transmission Strategy of Spatial Information for WWW and Mobile Applications", HSI 2003, LNCS 2713, pp.12-22

[3] Zhanya XU, Zhong XIE, *"Research on Key Technology of General Embedded GIS"*,

[4] P. Yu, L. Yuye, *"An Embedded GIS Integrating GSM and GPS under VxWorks Platform"*, 2010, First International Conference on Pervasive Computing, Signal Processing and Applications

[5] C. Feixiang, Z. Ming, Q. Jiandong, J. Xiaoru, *"Research on Mobile GIS Based for Information Collection",* 2009 International Forum on Information Technology and Applications

[6] Alexander. Kolesnikov, Pasi Franti, "Data reduction of large vector graphics", Pattern Recognition, 2005, 38.pp.381-394.

[7] Xin Xu, Jinguang Gu, Hongbing Zhu, Yimin Qiu, “Design and Realization of the Enhanced Compression Method aimed at the vector map”, International Journal of Intelligent Engineering and Systems 2 (2008) 10-17

[8]http://www.maximumcompression.com/lossless_vs_lossy.php

[9]http://www.geospatialworld.net/index.php?option=com_content&view=article&id=19277&Itemid=408