A Map Tiling Algorithm and Implementation Based on Mxd Documents

Yonglong Xu, Bin Li, Mingxia Xu, Lichao Zhao
College of Geological Engineering and Geomatics
Chang'an University
Xi'an, 710054, P.R.China

Abstract—***Aiming at the needs for using map tiles and creating map tiles in web map applications, this study puts forward a practical coding algorithm for digital map tiles. Based on the map tile algorithm, a program of creating digital map tiles for ArcGIS mxd map documents was implemented, which can produce seamless map tiles at any scale in any extent.***

Keywords-WebGIS; map tiling algorithm; map tiles

I. INTRODUCTION

With the rapid development of web geographical information system (WebGIS), more and more web map applications appear on the Internet, such as Google maps, Baidu map, Map World and other online maps. All these applications use map pictures as the carrier of visual geographic information, therefore, how fast the applications' client-side can get access to the map pictures on the server-side mainly determines the applications' overall efficiency [1]. In the early stage of web map applications, all map pictures displayed on the client-side are generated by real-time dynamic rendering pictures on the server-side, which inevitably increase the server burden and also reduced the overall efficiency. Since Google put forward the conception of map tiles, the map tile technique has been widely used to solve the web map efficiency problem and achieved great success. By caching map tiles on the server-side in advance instead of real-time dynamic rendering so that the client-side can get quick access to map tiles and splice them, this technique effectively solves the low efficiency problem in early web map applications.

At present, most WebGIS applications use map tile technique in their map display, but the generation and the web publishing of map tiles are contained in the wholesale solutions of mainstream commercial GIS software such as ArcGIS Server, with high cost and low flexibility, rarely meeting specific WebGIS map application's actual demands. How to use existing digital maps to generate map tiles independently is an urgent problem faced by current self-developed WebGIS applications. Based on the analysis on the technical principles of map tiles, this study designed a practical digital map tiling algorithm for general use, and implemented it in a computer program which can generate map tiles using ArcGIS map documents (namely *.mxd files).

II. TECHNICAL PRINCIPLE OF MAP TILES

Map tiling is a process that cuts a digital geographical map into several rows and columns of square raster images which are also known as map tiles, with fixed length and width in a certain scale level.

The map is cut in a quadtree way, which means the map in each tile's extent is cut again into 2 rows and two columns till the maximum scale [2, 3], so each tile's map scale in current scale level is twice as in previous level and the number of tiles in current scale level is 4 times as in previous level. For the tiles in each level, we can code them with their level, row number and column number from left to right, top to bottom, e.g. coded tiles in level 1 and level 2 is shown in Fig.1.

The server usually consumes large amounts of storage space to store all the map tiles. When viewing or zooming in the client map, the client dynamically determines rows and columns of the map tiles which are needed to display within the current geographical extent, and then gets tiles directly from the server-side and seamlessly displays them, so as to achieve the purpose of rapid response [4].

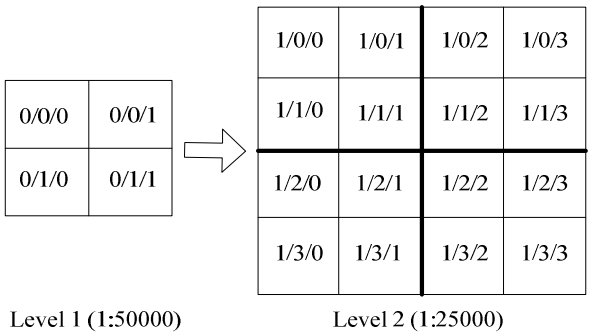

Figure 1 Technical principle of map tiles

978-1-4673-0964-6/12/$31.00 ©2012 IEEE

III. A DIGITAL MAP TILE ALGORITHM

The technical principle of map tiles above shows that the key point in map tiling is to figure out the calculation formulas of the relationships between row number(or row count), column number(or column count) and the map's extent, tiles' scale, size and resolution, so as to correctly code all the tiles. The only difference between the two ways of tiling a map mentioned in [5], i.e. globe-map-based and local-map-based, lies in the map's geographic extent. Through the study on these two tiling methods, this paper put forward a general calculation method for tiling a map.

A. The Map Tile Algorithm

For a given digital map, its geographic extent is constant, so we use X_{max}, X_{min}, Y_{max}, Y_{min} in meters to describe the maximum and the minimum value of x and y coordinates of the map's extent. If both the length and the width of a map tile in pixels is T (here and after we consider the tile's length and width are equal), and its resolution is D pixels per inch, the length of the tile l in meters can be calculated in Formula (1).

$$l=(T/D)\times 0.0254 \quad (1)$$

If the current map scale is K, the actual distance d represented by map distance l can be calculated in Formula (2).

$$d=l/K=(0.0254\times T)/(D\times K) \quad (2)$$

Here we provide that $M_r=(Y_{max}-Y_{min})/d$. If M_r is an integer, the row number of map tiles R in current scale K can be calculated in Formula (3), else in Formula (4).

$$R=M_r=(Y_{max}-Y_{min})/d \quad (3)$$

$$R=\text{int}(M_r)+1=\text{int}((Y_{max}-Y_{min})/d)+1 \quad (4)$$

In Formula (4), *int* is an operation that keeps only the integer part of a numeric value.

Similarly, we provide that $M_c=(X_{max}-X_{min})/d$. If M_c is an integer, the column number of map tiles C in current scale K can be calculated in Formula (5), else in Formula (6).

$$C=M_c=(X_{max}-X_{min})/d \quad (5)$$

$$C=\text{int}(M_c)+1=\text{int}((X_{max}-X_{min})/d)+1 \quad (6)$$

In Formula (6), *int* is an operation that keeps only the integer part of a numeric value.

In Formula (3), (4), (5) and (6), if we replace X_{max} and Y_{min} respectively with x and y, we can calculate the row number and the column number of the map tile in which the point (x, y) is located. Also, for given row number and column number of a map tile in a certain scale, we can get the tile's geographic extent by back calculation.

Therefore, as long as the map scale and the size and the resolution of the tile is specified, we can calculated the tiles' row count and column count, and then code them according to the rule shown in Fig. 1.

B. Algorithm Verification

As for a globe map, its extent is from 180°W to 180°E in the longitude direction and from 90°S to 90°N in the latitude direction. After projected to the plain by WGS_1984_Web_Mercator projection, the globe map's length is just twice as long as its width, and the length of x direction (or X_{max}-X_{min}) is $2\pi\times$ 6378137 meters and that of y direction (or Y_{max}-Y_{min}) is $\pi\times$6378137 meters. If the first level of the globe map is regarded as two tiles splicing together in one row, as is shown in Fig. 2, the row count of globe map tiles in level n is 2^{n-1} and the column count is 2^n. Therefore, we can use that to verify calculation formulas of the map tile algorithm.

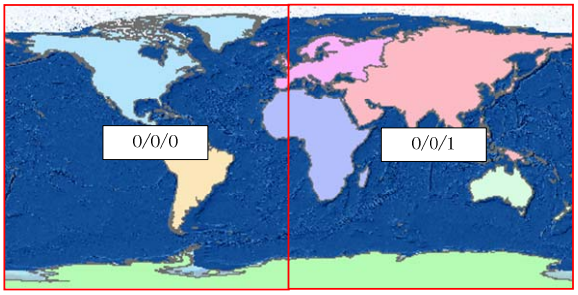

Figure 2 The first level tiles of globe map

If the length and the width of map tiles are 256 pixels and the resolution is 96 pixels per inch, the scale K_0 of the first level of globe map can be calculated in Formula (7).

$$K_0 = l/(\pi\times 6378137) = ((T/D)\times 0.0254)/(\pi\times 6378137) \approx 1/295829355.5 \quad (7)$$

From the technical principle of map tiles shown in Fig. 1 we see that the scale of globe map in level n is $2^{n-1}K_0$. Then we calculated the row count and the column count of globe map tiles in the first 5 levels according to Formula (3), (4), (5) and (6), as is listed in Table 1.

TABLE I. CALCULATED ROW COUNT AND COLUMN COUNT OF GLOBE MAP TILES

Tile level (n)	Map scale (K)	Calculated row count (R)	Calculated column count (C)
1	1/295829355.5	1	2
2	1/147914677.9	2	4
3	1/73957338.9	4	8
4	1/36978669.4	8	16
5	1/18489344.7	16	32

Calculated results in Table 1 show that the row count of globe map tiles in level n is 2^{n-1} and the column count is 2^n statistically. The results are the same as the inference drawn from Fig. 2. So the map tile algorithm we put forward is verified to be correct and practical.

IV. ALGORITHM IMPLEMENTATION BASE ON MXD DOCUMENT

A. Why Choose Mxd Document

Mxd document is the map document file supported by ESRI ArcGIS software and the file extension is *.mxd. This kind of file contains digital maps and can be viewed and edited in ArcGIS. This paper chooses mxd file to conduct map tiling because the following advantages.

1) ArcGIS software supports various kinds of data source such as shapefile, geodatabase, coverage, and online data. All the supported data can be easily made into mxd map document files.

2) It's easy to configure the map layers in mxd documents so that we only need to set the layers' visibility to carry out single-layer or multiple-layer tiling.

3) The process of tiling a mxd document is much simpler than caching a map service in ArcGIS server [6, 7], because it doesn't need to publish the digital map on the server-side.

B. Map Tiling Work Flow

In this paper, the core idea of tiling a digital map in mxd documents is exporting the map in a specified geographic extent to a raster image whose length or width and resolution are also specified. For each level, the following steps are followed to cut a digital map into map tiles.

1) Calculate row count R and column count C of map tiles in current level;

2) Calculate geographic extent of the tile in Row 1 and Column 1, and then export the map in the calculated extent to an image with fixed size and resolution;

3) Repeat Step (2) until the tile images from Column 2 to Column C in Row 1 are exported;

4) Repeat Step (3) until tile images in all rows are exported.

As the number of map tiles is very large, tiles in the same level are stored in the same directory and tiles in different rows are stores in different subdirectories, and all tile image files are name with the tiles' column number, so as to manage and publish them easily.

C. Programmatic Implementation

In this paper, the authors implemented above map tiling work flow by computer programming based on C#.Net and ArcObjects. The interfaces and methods of ArcObjects mainly used in this program are listed in Table 2.

TABLE II. INTERFACES AND METHODS OF ARCOBJECTS MAINLY USED IN THE PROGRAM

Interface	Property/Method	Description
IEnvelope	PutCoords()	Constructs an envelope from the coordinate values of lower, left and upper, right corners.
IActiveView	Extent	The property used to get or set the visible extent rectangle
	Output()	This method is used to draw to a non-app window hDC
IExport	StartExporting()	Initializes the Exporter.
	FinishExporting();	Shuts down the Exporter.
	Cleanup()	Cleanup should clean all temporary files, free used memory, etc...

In the program, to achieve fast tiling, the user only needs to specify the mxd file path, the tiles' storage directory path and the map extent to tile, and set the size, resolution and file type of the tile image. After the mxd map is tiled, all tiles will be stored in different directories according to their levels, row numbers and column numbers, as is shown in Fig. 3. The map tiles can be published on the web server for online or offline web map applications to use.

Based on the technical principle of map tiles, the paper studied on the map tiling algorithm and put forward a practical calculation method for coding and cutting digital map tiles, which can be used to tile digital maps in any geographic extent at any scale. The map tiling program implemented with C# and ArcObjects based on the algorithm can carry out fast tiling for mxd map documents, and the map tiles generated by the program can be published for online and offline web map applications to use. The map tiling algorithm and its implementation technique proposed in this paper provides a significant reference for building high efficient web map applications.

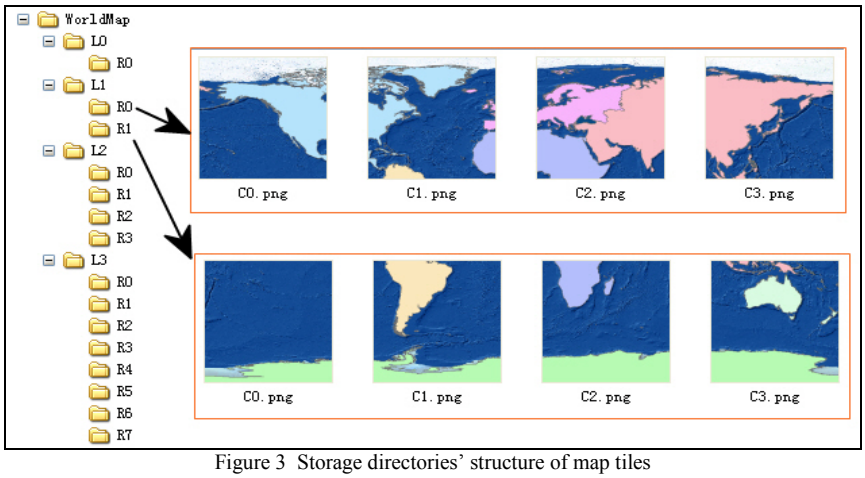

Figure 3 Storage directories' structure of map tiles

V. CONCLUSION

ACKNOWLEDGMENT

This paper is supported by the Fundamental Research Funds for the Central Universities (CHD2011TD019). The authors thank Prof. Peng Jianbing at Chang'an University for his kind help in preparing the manuscript.

REFERENCE

[1] ZHANG Jun-li, LI Yan-ming, LI Xin, "Research of map tile data buffer policy based on WebGIS," Microcomputer Information, vol. 25(10-1), pp. 141-143, 2009.

[2] ZHAO Da-long, SUN Heng-yu, "The simply analyzing and implement of map tile technology," Geomatics & Spatial Information Technology, vol. 33(1), pp. 116-118, Feb. 2010.

[3] NIE Yun-feng, LIU Hai-ling, XU Hu, "Research on GeoWebCache tile map service middleware," Science of Surveying and Mapping, vol. 36(6), pp. 208-209, 2011.

[4] CHEN Xiaoning, MA Yafei, "Web map service system application based on NewMap API," Bulletin of Surveying and Mapping, vol. 3, pp. 77-80, 2012

[5] WANG Yan-li, YIN Ke, ZHANG Lian-tang, "Research of the technique for map cache based on ArcGIS Server," Journal of Henan University (Natural Science), vol. 39(6), pp. 637-640, Nov. 2009.

[6] LI Aixia, GONG Jianya, JIA Wenjue, CHEN Yumin, "WebGIS based on WMS," Journal of Geomatics, vol. 29(6), pp. 1-2, Dec. 2004.

[7] Tile Map Service Specification (Online). http://wiki.osgeo.org/wiki/Tile_Map_Service_Specification