Journal of Southeast University (English Edition) Vol. 27, No. 3, pp. 270 - 274 Sept. 2011 ISSN 1003—7985

An algorithm supporting fast scheduling and viewing of remote sensing images in embedded devices

Li Ling Wang Qing Zhang Xiaoguo Wang Huiqing

(School of Instrument Science and Engineering, Southeast University, Nanjing 210096, China)

Abstract: In order to resolve the conflict between the limited resources of embedded devices and the growing amount of massive image data to be shown, a solution for fast images rendering in embedded devices is proposed and implemented. First, an improved algorithm of a multi-resolution tile-pyramid construction which is used for the organization of massive image data is presented. Then, a strategy, adopting technologies such as view-dependent levels of detail, target-tiles quick search and tiles seamless connection, is presented for fast scheduling and viewing of images. The results show that compared with the solution of multi-scale image representations based on wavelet, the proposed solution can improve the rendering speed, and the rendering speed does not depend on the image size, though it increases some data storage space. And the proposed solution is suitable for embedded devices because it can provide real-time and friendly interaction as well as more efficient user experience.

Key words: embedded devices; pyramid; levels of detail; image rendering

doi: 10.3969/j.issn.1003 - 7985.2011.03.009

Remote sensing images are characterized by wide coverage, wide perspective and multi-resolution (spectrum and space)[1]. Recently, they have become one of the main sources of spatial information, and are more and more widely used for land survey and resource management[2]. With the development of surveying and mapping technology, a large amount of geographical data is exponentially growing[3]. However, when analyzing, processing and viewing data, it is usually not suitable to load the whole image into memory. The reason is that the memory size both in computer and embedded devices is limited. Hence, it is of significance to study techniques to use the limited resources of embedded devices to process, query and view images[4].

So far, many research works have been reported on fast real-time visualization of massive images[2-3,5-6]. However, most existing solutions have the following limitations: 1) They are mainly used for personal computers (PCs), and are not specially designed for embedded devices; 2) Traditional wavelet transform techniques include zerotrees of wavelet coefficients, discrete wavelet transform, integer wavelet transform, line-based wavelet transform and so on; some solutions based on these wavelet transform need to construct complex wavelet bases and wavelet functions, so they are not widely used[6]. Enhanced compressed wavelet (ECW), a patented technology, avoids these disadvantages[6] and can be applied to fast viewing for images in embedded devices; but if the image size is greater than 500 MB, a license is needed to buy to compress this image. Moreover, when roaming large-scale images, the image display may overlap or become a dark strip, and sometimes an obvious stop may appear [2].

Still, several technologies for rapid rendering of large-scale images on personal digital assistants (PDA) were proposed in Refs. [7 -9]. In Ref. [7], the file-based approach was used for data organization and management, and all the tiles data were stored into a series of files. A new image data model was proposed to combine the tile-pyramid with the quad-tree index in Ref. [8]. Zhang et al. [9] used the GDAL open source library to create a pyramid file, which increased data storage space.

In this paper, a practical solution to realize fast scheduling and viewing of RS images in embedded devices is proposed. Compared with the previously proposed technologies, it has some key differences. First, all the tile data are stored into a multi-file. Secondly, our scheduling strategy is simple, and does not use the complex quad-tree index. Finally, the additional data storage space size in this paper is smaller than that in Ref. [9]. Compared with the ECW solution, the proposed solution can improve image rendering speed, even though it increases some data storage space. The image rendering speed does not depend on the original image size. So it is suitable for embedded devices.

1 Tile-Pyramid Model

1.1 Tile-pyramid model and tile parameters

The tile-pyramid model, also called the hierarchy model[10], supporting multi-resolution functions[11], has been widely used to organize RS images[12]. Fig. 1 shows a tile-pyramid model of four levels. Each level in the tile-pyramid corresponds to a different resolution. From the bottom to the top level, the image resolution becomes lower and the quantity of the data becomes smaller. Visually, tiles of one specific resolution level are equal-sized (see Fig. 1).

Generally, the tile size is chosen as $2^n \times 2^n$, where n can be any integer greater than 1. And the tile size should also be chosen appropriately[13]. If it is too large, loading redundant data may reduce image rendering speed; while if it is too small, too many I/O operations may also reduce image rendering efficiency. As long as the tile size is larger than the size of the display area, even in the worst case, it usually only needs to load four pieces of tiles. That means all the required data can be obtained using four times the file pointer positioning and data reading operations. The screen reso-

Received 2011-04-22.

Biographies: Li Ling (1986—), female, graduate; Wang Qing (corresponding author), male, doctor, professor, w3398a@263.net.

Foundation item: The National Public Benefit Research Foundation of China (No. 201111013-02).

Citation: Li Ling, Wang Qing, Zhang Xiaoguo, et al. An algorithm supporting fast scheduling and viewing of remote sensing images in embedded devices[J]. Journal of Southeast University (English Edition), 2011, 27 (3): 270 - 274. [doi: 10.3969/j.issn.1003 - 7985.2011.03.009]

lution of the PDA we use is 240 × 320 pixel, and the size of its display area is 217 × 249 pixel. Therefore, the tile size we define is 256 × 256 pixel. The tile-splitting procedure starts from the left-top corner of the image, then from left to right and top to bottom.

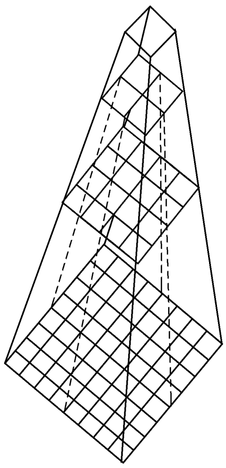

Fig. 1 Four-level tile-pyramid

1.2 Improved algorithm of tile-pyramid construction

The traditional tile-pyramid construction algorithm includes two main steps[14]: 1) Create temporary image files for image data of every level by re-sampling; 2) Perform the tile-splitting algorithm on temporary image files and save all the tile data into a series of files. Its main disadvantage is that it carries out too many I/O operations, which can reduce the efficiency of the algorithm.

Unlike the traditional tile-pyramid construction algorithm, the main advantages of our algorithm are: 1) Only perform the required I/O operations, including loading the original image and saving all the tile data into a multi-file; 2) Fully make use of computer memory from the perspective of software processing. The latter means that the memory occupied by the previous level should be released after obtaining data of the current level by re-sampling; then we perform the tile-splitting algorithm on the current level. Thus, the operating efficiency of the program and its ability to process original image data can be improved. As shown in Fig. 2, the procedure of tile-pyramid construction is described as follows:

1) Define the total number of levels of the tile-pyramid as N according to the original image size;

2) Assume that k is the current level, and initialize k with -1;

3) Load the original image data as the 0th level data, and $k=0$;

4) Perform the tile-splitting algorithm on the k-th level, and save the tile data;

5) Perform the resampling algorithm on the k-th level to obtain the upper-level data block, and then $k=k+1$;

6) Perform the tile-splitting algorithm on the new data block, and save the tile data;

7) If k is not equal to $(N-1)$, go to step 5), until tile-pyramid construction is accomplished.

The implementation process of the tile-splitting algorithm mentioned above is described as follows:

1) Input parameters include poDataBlock, GT[6], W, H and k; while poDataBlock is the data block of the k-th level; GT[6] represents the affine geo-referenced conversion parameters of the k-th level, which are used for coordinate transformation between pixel coordinates and geographic coordinates; W and H are the width and height of poDataBlock, respectively (unit: pixel); k represents the level of the pyramid.

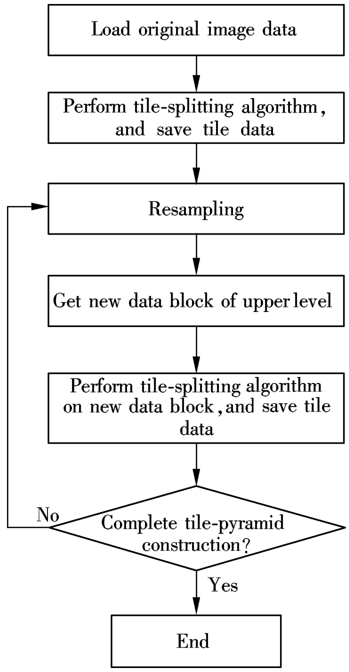

Fig. 2 Flowchart of constructing tile-pyramid

2) Assume that the tile size is $W_T \times H_T$;

3) According to W and H, the size $M_k \times N_k$ of the k-th tile matrix can be calculated as follows:

$$M_k = \left\lfloor \frac{H-1}{H_T} \right\rfloor + 1, \quad N_k = \left\lfloor \frac{W-1}{W_T} \right\rfloor + 1 \tag{1}$$

where $\lfloor \ \rfloor$ represents round down.

4) If W is not a multiple of W_T, the right border of the image should be padded with a default value. Similarly, if H is not a multiple of H_T, the bottom border of the image should also be padded with a default value.

5) For each tile, initialize the size of the tile data block with W_T and H_T;

6) Extract the contents of the tile data block from the poDataBlock according to the tile size and the position of the tile data block in its tile matrix;

7) Utilize GT[6] to calculate the affine geo-referenced conversion parameters of the tile data block;

8) Save the tile data;

9) If the row number of this tile is not equal to (M_k-1), or the column number of this tile is not equal to (N_k-1), go to step 5), until the tile-splitting process of the current level is accomplished.

2 Fast Scheduling and Viewing Strategy of RS Images

2.1 View-dependent LOD strategy

The principle of the view-dependent LOD strategy can be briefly described as follows: 1) Set a viewpoint and a specific visual scene on data expression; 2) Render the image with an appropriate resolution data which should be within the visual scope. The latter means that we render the image with low resolution data when it is far from the viewpoint, and with high resolution data when it is near the viewpoint.

2.2 Fast scheduling and viewing strategy

After accomplishing the organization of the image data based on the tile-pyramid, a reasonable strategy for fast image rendering needs to be studied. In this paper, the required data for rendering can be obtained in terms of a query operation which is related to two parameters, namely, a given query area and resolution. The query area and resolution can be expressed as (x_1, y_1, x_2, y_2, r), where (x_1, y_1) and (x_2, y_2) are upper-left and lower-right screen coordinates, respectively (unit: pixel), and r is the view resolution. The fast scheduling and viewing strategy has two main procedures: 1) Search and find the optimal level from tile-pyramid; 2) Search and find the optimal target-tiles. During the latter procedure, we do not compare the query area with the minimum boundary rectangle (MBR) of each tile. Thus, it accelerates the query speed.

Tiles at every level can be used to form an $M \times N$ matrix. (M_r, M_c) represents the tile position in its matrix, where M_r and M_c are the row number and the column number, respectively. All the tile data are stored into a multi-file.

(x, y) is a pixel point of the image at a level, and its tile position (M_r, M_c) in the matrix can be calculated as follows:

$$M_r = \left\lfloor \frac{y}{H_T} \right\rfloor, \quad M_c = \left\lfloor \frac{x}{W_T} \right\rfloor \tag{2}$$

The pixel coordinate (x, y) of its upper-left point can be calculated as

$$x = M_c \times W_T, \quad y = M_r \times H_T \tag{3}$$

Assume that $(x_{\min}, y_{\max})$ is the upper-left coordinate of the original image under the geographic coordinate system, and the procedure of the fast scheduling and viewing strategy is described as follows:

1) According to the current view resolution r, the modified dichotomy algorithm is utilized to quickly locate the optimal view level of the tile-pyramid, and the optimal level is assumed to be i.

2) Calculate the affine geo-referenced conversion parameters of the i-th level.

3) Assume that (x_s, y_s) is a point on the screen and calculate its geographic location (x_g, y_g). Then the position (M_r, M_c) of its tile in the matrix can be calculated as follows:

$$M_r = \left\lfloor \frac{y_{\max} - y_g}{\text{fabs}(\text{GT}[5])\, H_T} \right\rfloor, \quad M_c = \left\lfloor \frac{x_g - x_{\min}}{\text{fabs}(\text{GT}[1])\, W_T} \right\rfloor \tag{4}$$

And $T_1(M_{r1}, M_{c1})$ of the upper-left point $P_1(x_{s1}, y_{s1})$ and $T_2(M_{r2}, M_{c2})$ of the lower-right point $P_2(x_{s2}, y_{s2})$ on the screen can be obtained.

4) Based on i, M_{r1}, M_{c1}, M_{r2} and M_{c2}, the target-tiles which intersect with the screen scope can be determined.

5) In order to realize tile seamless connection, each target-tile's MBR is used to determine its display area on the screen through coordinate conversion; then the data of each target-tile are read and the double buffering mechanism of off-screen bitmap technology is used to quickly render images. Fig. 3 and Fig. 4 are the interface effects of the image display at two different levels.

Fig. 3 The 4th level image display

Fig. 4 The 0th level image display

3 Experimental Results and Discussion

In our experiments, commercial PDA G738L is selected as the hardware platform, and its basic configuration is described as follows: Windows Mobile 5.0 operation system; 240 × 320 pixel resolution; Intel Xscale processors 520 MHz; 64 MB SDRAM, 320 MB Flash, and built-in SD card slot. Visual Studio 2005 is chosen as the embedded developing tool. The version of ECW SDK used is 3.3, and its paging mechanism is enabled.

Considering the limited performance of PDA, the tile-pyramid is constructed on PC and then the preprocessed data is copied to the SD card.

To evaluate the performance of the proposed solution, two sets of experiments are done to test memory consumed by the program and average time cost of rendering images when roaming images. When the program is running without importing image data, the memory consumed is 1.01 MB.

Experiment 1 Comparing our solution with the ECW solution of the ERDAS company, the memory consumed by the program is tested under different sizes of images when roaming images. Before testing, original images (GeoTIFF format) need to be preprocessed as follows:

1) Perform our tile-pyramid construction algorithm on the original images.

2) Use SuperMap Deskpro software to compress original images into ECW files. The preprocessed image size is shown in Tab. 1. And the experimental results are shown in Fig. 5, where M represents memory consumed.

Tab. 1 Image size

Original image size/MB	Preprocessed image size/MB	
	ECW solution	Our solution
41.8	1.54	65.1
107.0	3.48	153.0
178.0	5.50	253.0
249.0	6.09	349.0
273.0	6.33	376.0
439.0	7.06	510.0

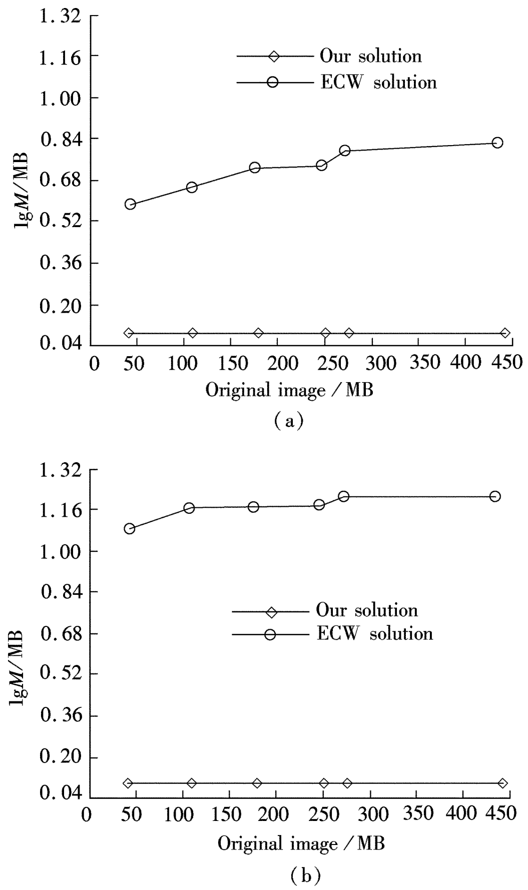

Fig. 5 Memory consumed. (a) Minimum memory consumed; (b) Maximum memory consumed

Experiment 2 Comparing our solution with the ECW solution, the average time cost of rendering images is tested under different sizes of images when roaming images. And the experimental results are shown in Fig. 6.

From the test results in Fig. 5, we can see that memory consumed in our solution is smaller than that in the ECW solution; and the memory consumed in our solution is nearly the same under different sizes of images. The reason is that the image is tiled, and no more than four tiles will be loaded and rendered. From the test results in Fig. 6, we can see that the rendering speed is improved in our solution.

Compared with the ECW, our solution has the following advantages:

1) The ECW solution based on the wavelet technology requires smaller data storage space, which can improve the efficiency of data access from the external memory. But it needs to decompress data, which makes its rendering speed slower. An obvious stop even appears when rendering large-scale image.

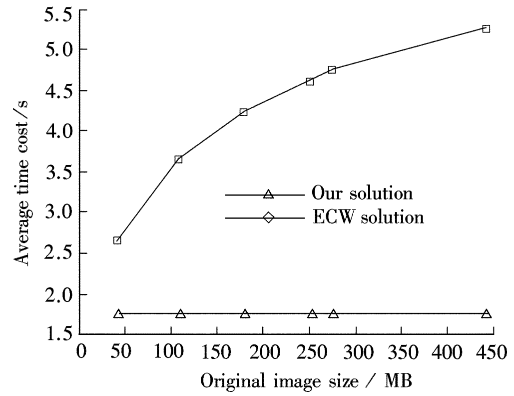

Fig. 6 Average time cost of rendering image

2) The image data is not stored in a compressed format in our solution, so our solution increases some data storage space (see Tab. 1). But, currently, the PDA can support the SD card with at least a 2 GB storage capacity, so it can be used to store the data of one district, which meets the requirements of the land survey.

3) Our solution reduces the quantity of data access and does not need to decompress data; therefore, it improves the rendering speed. Furthermore, the rendering speed does not depend on the image size (see Fig. 6).

4) The image can be viewed smoothly when roaming images in different ways, such as switching browsing between adjacent levels, skipping browsing, or continuous browsing within one level.

4 Conclusion

In this paper, a solution for fast scheduling and viewing of RS images in embedded devices is proposed. The experimental results prove that this solution is efficient. Compared with the ECW solution, despite the need for more data storage space, the proposed solution has better results in image rendering and memory consumption, and, furthermore, its rendering speed does not depend on the image size. It can be concluded that this solution is suitable for embedded devices because it can provide real-time and friendly interaction as well as more efficient user experience.

In most cases our solution can meet the demands of the land survey, but the disadvantage of the proposed solution is that the size of the preprocessed image file cannot be more than 4 GB. Because the file system of the SD card is of FAT32 format, the size of one file cannot be more than 4 GB. Further studies will solve this problem.

References

[1] Mei A X, Peng W L, Qin Q M, et al. *An introduction to remote sensing* [M]. Beijing: Higher Education Press, 2001. (in Chinese)

[2] Xi Y W, Xin L, Hong S. Fast view of mass remote sensing images based on image pyramid[C]//*Proc of the First International Conference on Intelligent Networks and Intelligent*

Systems. Wuhan, China, 2008: 461 − 464.

[3] Dou Xipeng, Lu Xiaoping, Lu Yao. The design of visualization and analysis component in the large-scale and multi-source spatial data [J]. *Bulletin of Surveying and Mapping*, 2011(1): 21 − 24. (in Chinese)

[4] Yu X G. Research on web service-based image processing for mobile commerce [J]. *Key Engineering Materials*, 2010, **439/440**: 117 − 122.

[5] Wang Y Z, Zhang L G, Sun J H, et al. Research on real-time visualization of massive data oriented to digital earth [C]//*Proc of International Conference on Information Engineering and Computer Science*. Wuhan, China, 2009: 1 − 4.

[6] Lü Xikui, Yi Sirong, Han Chunhua. Research on the remote sensing image compression based on ECW technology[J]. *Science of Surveying and Mapping*, 2007, **32**(5): 55 − 57. (in Chinese)

[7] Cheng Jinhai, Ke Ranxu, Peng Guojun. Technology of rapid rendering of massive remote sensing image in PDA [J]. *Journal of Jimei University*: *Natural Science*, 2009, **14**(4): 379 − 383. (in Chinese)

[8] Wang Tao, Deng Xueqing, Dai Chenguang, et al. A remote sensing image data model for PDA and its display algorithm [J]. *Science of Surveying and Mapping*, 2009, **34**(2): 184 − 186. (in Chinese)

[9] Zhang Jian, He Yadong, Zhou Lü, et al. Study on the read and display of geo-image based on PDA[J]. *Surveying and Mapping*, 2010, **33**(2): 69 − 71. (in Chinese)

[10] Tu S R, He X F, Li X F, et al. A systematic approach to reduction of user-perceived response time for GIS web services [C]//*Proc of the Ninth ACM International Symposium on Advances in Geographic Information System*. New York, USA, 2001: 47 − 52.

[11] Yang C W, Wong D W, Yang R X, et al. Performance-improving techniques in web-based GIS[J]. *International Journal of Geographical Information Science*, 2005, **19**(3): 319 − 342.

[12] Yang J Y, Ji S, Zhang Y S, et al. Investigation of dynamic reconstruction methods for remote sensing image pyramids based on multi-resolution characteristic of wavelet transformation[C]//*Proc of the* 2008 *Congress on Image and Signal Processing*. Sanya, China, 2008: 656 − 660.

[13] Chi C C, Chung H H. Improved visual tracking using the technique of image pyramid[C]//*Proc of the* 2009 *IEEE International Conference on Robotics and Biomimetics*. Guilin, China, 2009: 659 − 664.

[14] Tan Qingquan, Bi Jiantao, Chi Tianhe. A flexible and efficient algorithm for constructing remote sensing image pyramid [J]. *Computer Systems and Application*, 2008(4): 124 − 127. (in Chinese)

一种适用于嵌入式设备的遥感影像快速调度显示算法

李　玲　王　庆　张小国　王慧青

(东南大学仪器科学与工程学院,南京 210096)

摘要:针对嵌入式终端设备资源有限与日益增长的海量影像数据显示之间的矛盾,提出并实现了一种适用于嵌入式设备的影像快速渲染方案. 首先,研究了多分辨率瓦片金字塔构建改进算法,用以实现海量影像数据的组织管理. 然后,在此基础上利用视点相关层次细节技术、目标瓦片快速搜索技术和瓦片无缝拼接技术实现影像的快速调度和显示. 实验结果表明,和基于小波技术的多尺度影像显示方案对比,尽管所提方案需要增加数据存储空间,但可提供更快的显示速度,且图像显示速度和图像大小无关. 此外,所提方案可实现交互的实时性和友好性,并提供更高效的用户体验,因此适用于嵌入式设备.

关键词:嵌入式设备;金字塔;层次细节;影像渲染

中图分类号:O422.7