International Journal of Signal Processing, Image Processing and Pattern Recognition
Vol.9, No.9, (2016), pp.123-138
http://dx.doi.org/10.14257/ijsip.2016.9.9.12

A Pipelined-Based Multi-Thread Acceleration Method for Remote Sensing Image Progressive Transmission

Haicheng Qu[1,2], Junping Zhang[1*] and Yu Meng[3]

[1]*Dept. of Information Engineering,Harbin Institute of Technology, Harbin, China*
[2]*Dept. of Software Engineering,Software College of Liaoning Technical University, Huludao,China*
[3]*Dept.of Computer Application Technology,College of Information Science and Engineering of Northeastern University,Shenyang, China*
haichengqu@hit.edu.cn, zhangjp@hit.edu.cn, perfectmeng@gmail.com

Abstract

In order to meet the different data browsing requests for different users to the visual quality of remote sensing images in heterogeneous network especially under narrow bandwidth environments, an online remote sensing image progressive transmission model is constructed in which remote sensing image compression and decompression are synchronized with transmission. At the same time, a pipeline-based multi-threaded acceleration method has been proposed through solving the asynchronous problem between compression decompression and transmission to improve the efficiency of remote sensing progressive transmission. At last, an idea of retry broken downloads transmission interruption has been implemented to improve end-user interactive experience. Experimental results show that the whole processing speed has been improved nearly twice without reducing image transmission quality and the amount of data transmission was reduced evidently by using the proposed progressive transmission and real-time compression model.

Keywords: *Progressive transmission; Real-time processing; Socket protocol; Retry broken downloads*

1. Introduction

The demand for remote sensing image information has increased dominantly with the rapid development of Internet and photographic remote sensing construction, against this backdrop, the remote sensing image processing technology has become a hot spot in the image processing research [1]. Especially when used in the area of location-based service (LBS), remote sensing images could present many good features, such as abundant contents, intuitive to the users, short cycle of updating, *etc*. However, problems arise when handling large-sized remote sensing images of 10, 50, 100 or more Megabytes, due to the amount of time required for transmitting and displaying, time of transmission being even worse when a narrow bandwidth transmission medium is involved (*i.e.*, dial-up or mobile wireless network), because the receiver must wait until the entire image has arrived. To fill the gap that the compression and decompression are not synchronous, progressive transmission schemes are used. The progressive image transmission (PIT) is a technique to progressively approximate an image on the receiving device over the remote transmission. Instead of waiting a long time for a high resolution image, the remote user can grasp a rough image quickly, and then, continuously, finer and finer image will be constructed. Therefore, it is important and practical in managing large amount of images for remote users especially over low speed network channels.

*Corresponding Author

ISSN: 2005-4254 IJSIP

There are several PIT methods that have been developed. Previously, Tzou [2] presented a thorough review and comparison of some PIT methods, namely, the spatial domain, the transform domain, and the pyramid-structure approaches [3]. Among these approaches, there are trade-offs between system complexity and performance. Subsequently, many images compression methods which use scalable coding techniques have been proposed to implement progressive transmission such as bit planes, TSVQ [4], DPCM, and, more recently, matrix polynomial interpolation, Discrete Cosine Transform (DCT, used in JPEG) and wavelets (used in JPEG 2000), set partitioning in hierarchical trees (SPIHT), JPEG-2000 and *etc.* SPIHT algorithm is a quality progressive method which codes the most important wavelet transform coefficients first and transmits the bits so that an increasingly refined copy of the original image can be obtained progressively [5]. In addition, many improved versions have been proposed to enhance compress efficiency and visual quality [6-8]. JPEG-2000 [9] is an image compression standard and coding system which provides efficient code-stream organizations which are progressive by pixel accuracy and by image resolution (or by image size). On the other hand, many improved method are also put forward to reduce computation complexity or improve image quality. Deng [10] has proposed a novel progressive coding method based on ROI to improve transmission efficiency. Wang [11] proposed an improved STC image compression method by using the no overlapping non-symmetry and anti-packing model (NNAM) and the extended shading approach to reduce the decoding time. In order to reach higher processing speed, Song [12] adopts the pipelined CPU and GPU heterogeneous to accelerate inverse DWT in decoding system and achieves remarkable acceleration effect. However, the asynchronous problem between compression decompression and transmission need to be considered as a whole in order to get higher processing efficiency without special hardware acceleration [13] such as DSP, GPU, FPGA *etc.* In addition, favorable user experience is another problem which seems increasingly important in mobile applications. The main purpose of technological innovation is to enrich overall user experience and make data interaction much easier. So, in this paper, an online remote sensing image progressive transmission model is constructed in which remote sensing image compression and decompression are synchronized with transmission. At the same time, a pipeline-based multi-threads acceleration method has been proposed through solving the asynchronous problem between compression decompression and transmission to improve the efficiency of remote sensing progressive transmission. And an idea of retry broken downloads transmission interruption has been implemented to improve end-user interactive experience. The rest of this paper will be arranged as follows: Section 2 describes the progressive computing model and associative compression and decompression algorithm; Section 3 explains the system architecture and its accelerated implementation using multi-thread pipelined techniques; Section 4 shows the experimental results and Section 5 makes a summary.

2. Progressive Computing Model and Compression Algorithm

2.1. A Model of Top-down Progressive Computing

Top-down progressive computing is guided by and explores a multilevel granular structure in structured thinking and structured information processing [14]. In some situations, one can first explicitly construct a multilevel granular structure and then process the structure from top levels with smaller granularity to bottom levels with larger granularity. A basic top-down progressive computing algorithm is formalized as detailed in Table 1.

In this paper, we construct a multilevel granular structure based on SPIHT and JPEG2000 algorithms. Figure 1, shows the top-down multilevel code streams structure.

First, specific image was compressed into serial code stream segments which are defined as R_L_i (i=1, 2, 3, ⋯, n; R: Resolution; L: Level). R_L_1 stands for the core code stream or rather the low-frequency component of image. And R_L_i (i=2, 3, ⋯, n) stands for incremental code stream. The current i-th resolution image $\mathbf{f_i}$(i=2,3,…,n) is added with the previous incremental code streams R_L_i, resulting in the finer resolution images .The equations above can be expressed as:

$$f_2 = D(f_1, R\_L\_1) \tag{1}$$

$$f_3 = D(f_2, R\_L\_2) \tag{2}$$

$$f_n = D(f_{n-1}, R\_L\_(m-1))(m = 2,3,\cdots,n) \tag{3}$$

Where D(·) is image restoration operation.

Table 1. Basic Progressive Computing Algorithm

1. begin
2. $\boldsymbol{k} \leftarrow 0$;
3. Initial $\boldsymbol{G}_0$ and $\boldsymbol{R}_0$;
4. do $\boldsymbol{k \leftarrow k+1}$
5. $T_{cru} = T_{c_1} + T_{r_1} + T_{u_n} + \sum_{k=1}^{n-1} \max(T_{c_{k+1}}, T_{r_{k+1}}, T_{u_k})$;
6. $R_k = result\_refinement(R_{k-1}, G_k)$;
7. Calculate fitness value $\boldsymbol{F}_k$:
8. $F_k = fitness(R_k)$;
9. until $\boldsymbol{F}_k$ satisfies a certain condition ;
10. end

Note: granulation_refinement(·) and result_refinement(·) represent granulation and result refinement operations respectively, and fitness(·) means computing threshold operation.

2.2. Referred Compression Algorithm

The implementation of progressive transmission model needs efficient algorithm support for compression and decompression of remote sensing images. SPIHT is a method of coding and decoding the wavelet transform of an image. By coding and transmitting information about the wavelet coefficients, it is possible for a decoder to perform an inverse transformation on the wavelet and reconstruct the original image. The entire wavelet transform does not need to be transmitted in order to recover the image. Instead, as the decoder receives more information about the original wavelet transform, the inverse-transformation will yield a better quality reconstruction (*i.e.*, higher peak signal to noise ratio) of the original image. SPIHT generates excellent image quality and performance due to several properties of the coding algorithm.

JPEG-2000 is a new image compression standard. The compression algorithm is mainly based on DWT. This image format has many good features, and some of them are very beneficial for mobile obtained remote sensing image compression, including:

(1) High compression ratio, at the same image quality, its compression rate can be 30% higher than that of JPEG. Image can keep smooth when the bit rate is low (without blocking artifacts);

(2) Progressive recovery of an image by fidelity or resolution, instead of waiting for downloading the whole image file, users can quickly browse the image in a lower quality, and the quality will be enhanced with more data transferred;

(3) Region of interest region, different parts of an image can be coded with differing fidelity, so we can set higher bit rates for important features;

(4) Good error resilience, some redundant information is retained after encoding, in order to resist bit errors and frame dropping during transmission, even if the code stream is incomplete, the whole image could be represented in a lower quality.

At present, development platforms support JPEG-2000 far less than JPEG. Because of its complex algorithm, the development would be difficult, and the decoding would give much pressure to terminal devices. In fact, the JPEG2000 developer toolkit named Kakadu[1] was adopted to reduce workload in this paper.

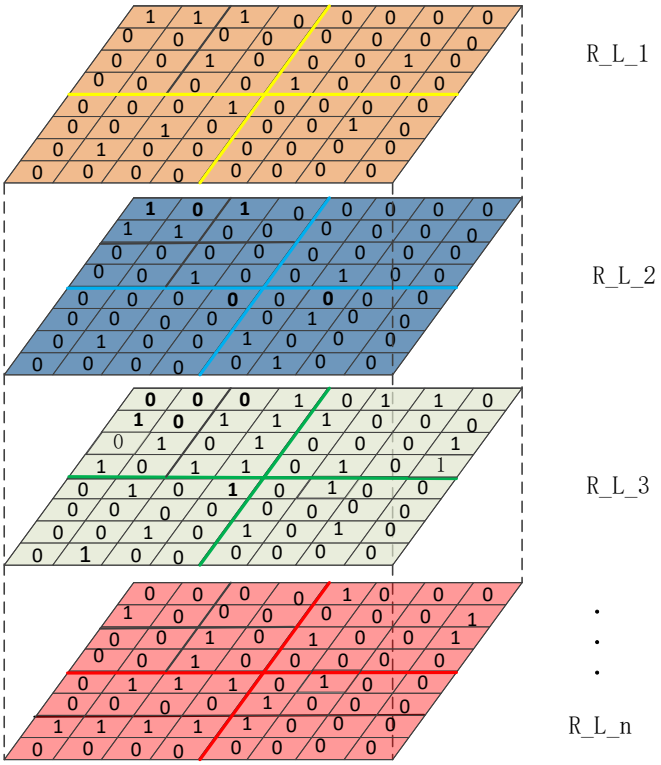

Figure 1. Multilevel Code Streams Data Structure

3. Implementation of Progressive Transmission

3.1. Progressive Transmission System Architecture

In order to implement the above-mentioned model, appropriate progressive transmission system architecture should be designed to satisfy the user's different data browsing requests. Especially when used in LBS, the remote sensing information is mainly obtained by mobile devices combining with users' interactive operation with the server. So a remote sensing mobile service should have some good features, mainly as follows [15]:

[1] http://www.kakadusoftware.com/

(1) Mobility, mobile device is the carrier of the client; it needs to interact with the server through wireless communication network which is narrow bandwidth generically;

(2) Real-time performance, the server must process requests and transfer images in time, so that users can browse and operate real-time;

(3) Scalability, the carriers of the client may be different, and the demands of various user groups would also be different, so system must adapt to the differences.

According to these features, the architecture of a remote sensing information mobile service is roughly designed as follows (as showed in Figure 2). It is mainly constructed with Client, Web server, application server and data sources. Client: running in PDA, smart phone, or other mobile devices. Usually, they should be able to access wireless network (*e.g.*, Wi-Fi signal). For the restriction of the terminal equipment, the client would not undertake complex computing, and should reduce the amount of local data. Its main functions would be interacting with the server, sending requests, downloading and displaying remote sensing images. It requires concise and friendly operating interface. Management server: would deal with http requests from clients, parse and forward them to compression server; on the other hand, receive the processing results from compression server, and inform the client to download remote sensing images and other data. Compression server: deal with users' requests sent by management server, process and compute, compress image data to code streams from the image data sources, and deal with load-balance problem. Image data source: include remote sensing databases, metadata databases, *etc*,. Always be ready for upper servers to get data real-time. Data may be distributed in several servers, depending on the scale of the system. In addition, it should deal with data security and backup.

In order to simulate the whole progressive transmission, a simplified model was designed as shown in Figure 3. This simplified model consists of three parts: images reading and compression part, data transmission part, images decompression and visualization part. The main work is embodied in remote sensing image data progressive transmission part which was implemented in Section 3.

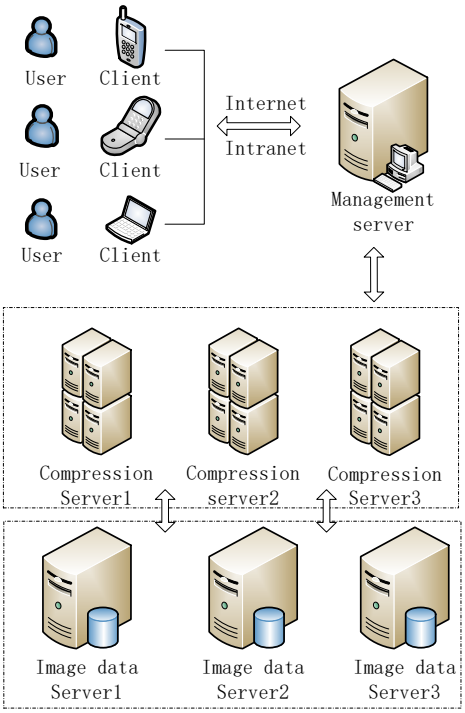

Figure 2. Architecture of Remote Sensing Mobile Service

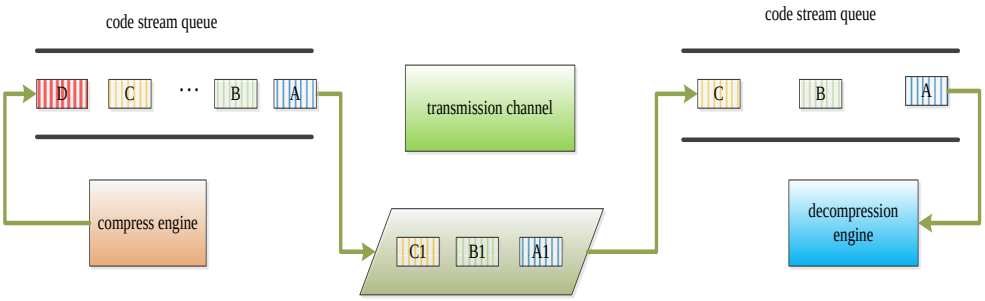

Figure 3. Real-Time Compression Progressive Transmission Model

3.2. Parallel Acceleration Using Multithreads

For single-thread mode, the progressive processing scheme execution time is described in Figure 4. Each rectangle represents one packet (a code stream segment which size depends on specific compression algorithm) at a time corresponding to the state, *i.e.*, T_c: represents a time compression time for a packet, T_r: transmission time for a packet, T_u: decompression time for a packet, n is the total number of packets. So for a specific image, the total processing time T_s is defined as follow:

$$T_s = \sum_{k=1}^{n} (T_{c_k} + T_{r_k} + T_{u_k}) \tag{4}$$

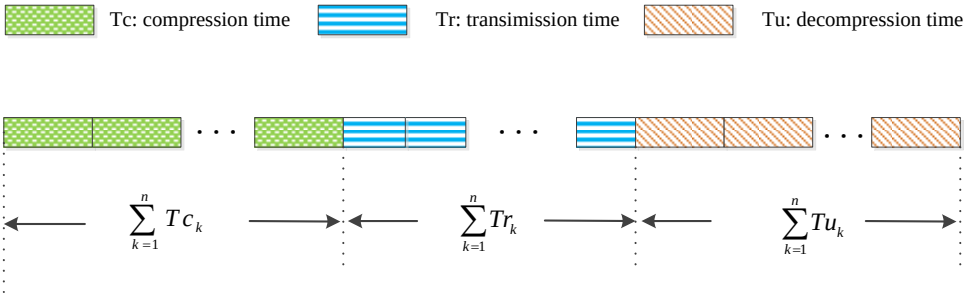

Figure 4. Single-Thread Image Processing Time

For multithreaded pipeline parallel mode, the progressive processing scheme execution time is described in Figure 5. Due to image compression and decompression are synchronized with transmission, the total execution time can be reduced through pipeline hidden each other. Since compression, decompression, and network transmission are independent during normal processing, the processing time is entirely different for each packet. Figure 5. shows one case of $T_c > T_r > T_u$. In general, the parallel execution time between compression and transmission can be formulized as follows:

$$T_{cr} = T_{c_1} + T_{r_n} + \sum_{k=1}^{n-1} \max(T_{c_{k-1}}, T_{r_k}) \tag{5}$$

And the parallel execution time between transmission and decompression can be formulized as follows:

$$T_{ru} = T_{r_1} + T_{u_n} + \sum_{k=1}^{n-1} \max(T_{c_{k+1}}, T_{r_{k+1}} T_{u_k}) \tag{6}$$

So the total executing time can be formulized as follows:

$$T_{cru} = T_{cr} + T_{ru} - T_h \tag{7}$$

where T_h means pipeline hidden time.

$$T_h = T_{r_n} + \sum_{k=1}^{n-1} \max(T_{c_{k+1}}, T_{r_k}) \tag{8}$$

$$T_{cru} = T_{c_1} + T_{r_1} + T_{u_n} + \sum_{k=1}^{n-1} \max(T_{c_{k+1}}, T_{r_{k+1}}, T_{u_k}) \tag{9}$$

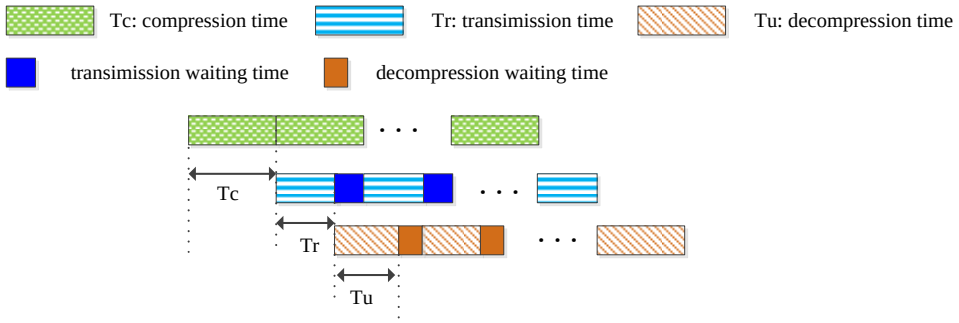

Figure 5. Multithread Pipeline Processing Time

3.3. Real-Time Compression and Progressive Transmission

The real-time compression and progressive transmission model is composed of four parts: transmission part, compression and decompression part, database part and visual show part. The whole interactive procedure about compression, transmission to decompression and visualization is shown in Figure 6. First, it is initialization of model system and configuration of parameters for client and server. Once succeed in creating connection, the client query requests for interested image will be submitted to the Web server. And then, Web server will process the query requests with the database engines immediately. After a moment, the query results which include attributes of remote sensing images and their accessing addresses will be returned to the client. Next step, client browses the specific image of interest from database server. The multilevel code streams which are generated on the database server by a compression algorithm such as SPIHT or JPEG2000 are transmitted to the client with a pipeline-based multi-threaded acceleration. When a complete code stream packet has been received, the decompression thread will be startup to renew the current image in the client until arriving at the end mark. Once an intermediate image satisfies the user's request, the client can stop the transmission request immediately without transmitting all the multilevel code streams. The detailed description about data transmission will be shown in Section 3.4. At last, the visual engine will show two formats of progressive effect both resolution and quality according to the user's preference.

3.4. Online Compression and Decompression with Retry Broken Downloads

Presently, retry broken downloads (RBD) is widely used in large file transfer protocol (FTP).When a client downloads data from server, the downloading tasks (a file or a compressed package) are artificially divided into several parts and each part is processed by one thread. A case of network failure or other breakdowns, the client could record the interruption positions. When recovered, the client can continue to download data from interruption positions. In this paper, this idea was introduced to progressive transmission system. That is to say that, the client can stop transmit image data at any time according to the users' preferences. If we want to acquire more distinct image, the client can continue transmit data from the server without starting afresh through retry broken downloads techniques. The whole procedure is designed in Figure 7.

The online remote sensing image progressive transmission model with RBD is constructed as shown in Figure 8. Remote sensing image compression and decompression are synchronized with transmission. At the same time, the pipeline-based multi-threads acceleration scheme was designed to improve the efficiency of remote sensing progressive transmission.

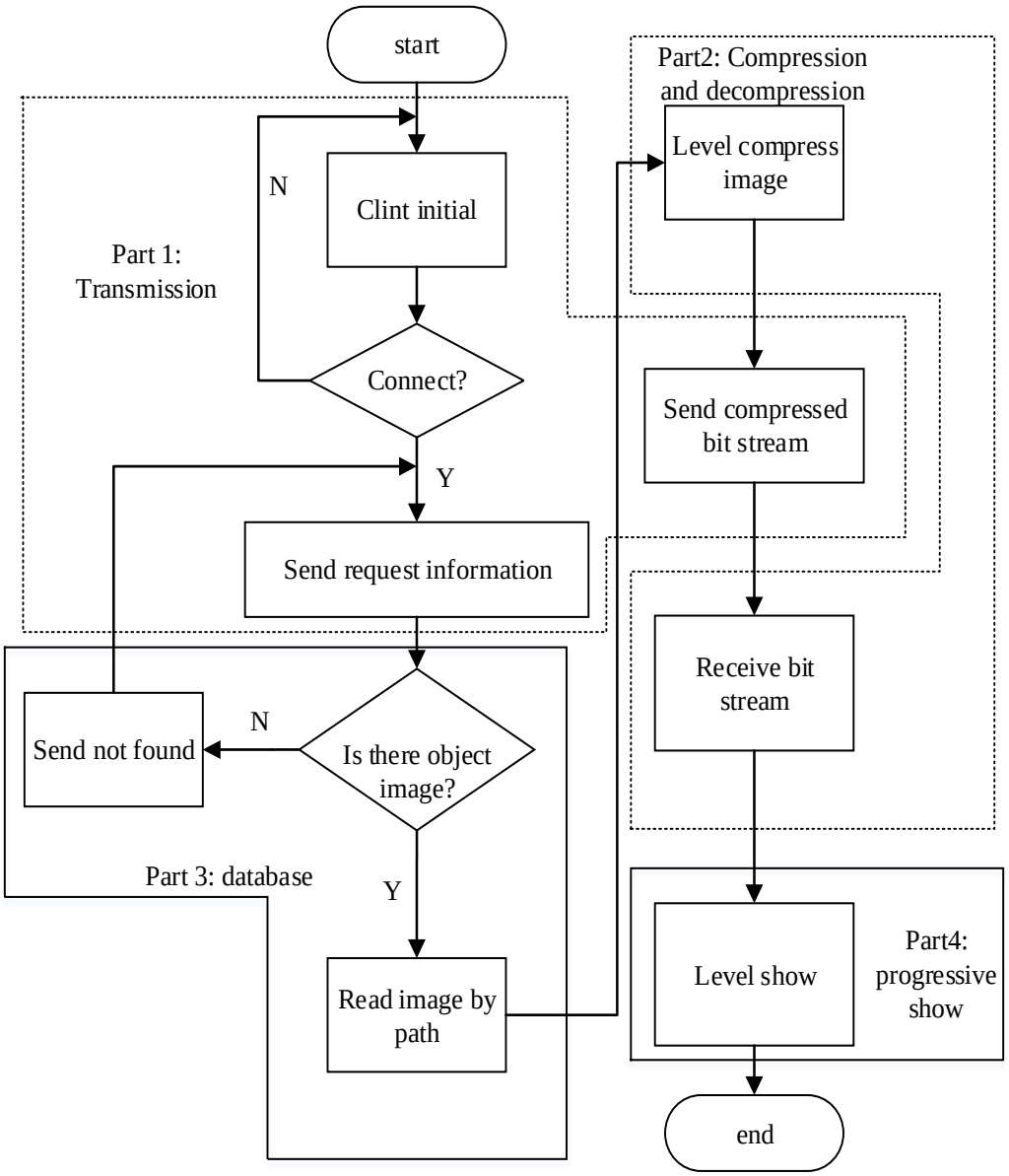

Figure 6. Global Image Data Interactive Procedure

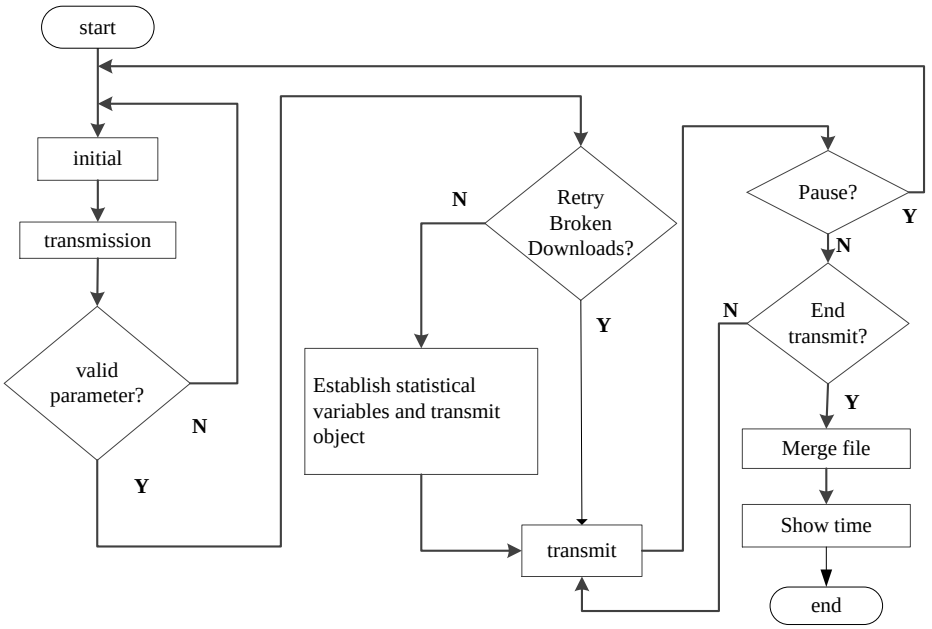

Figure 7. Retry Broken Downloads Flow Chart

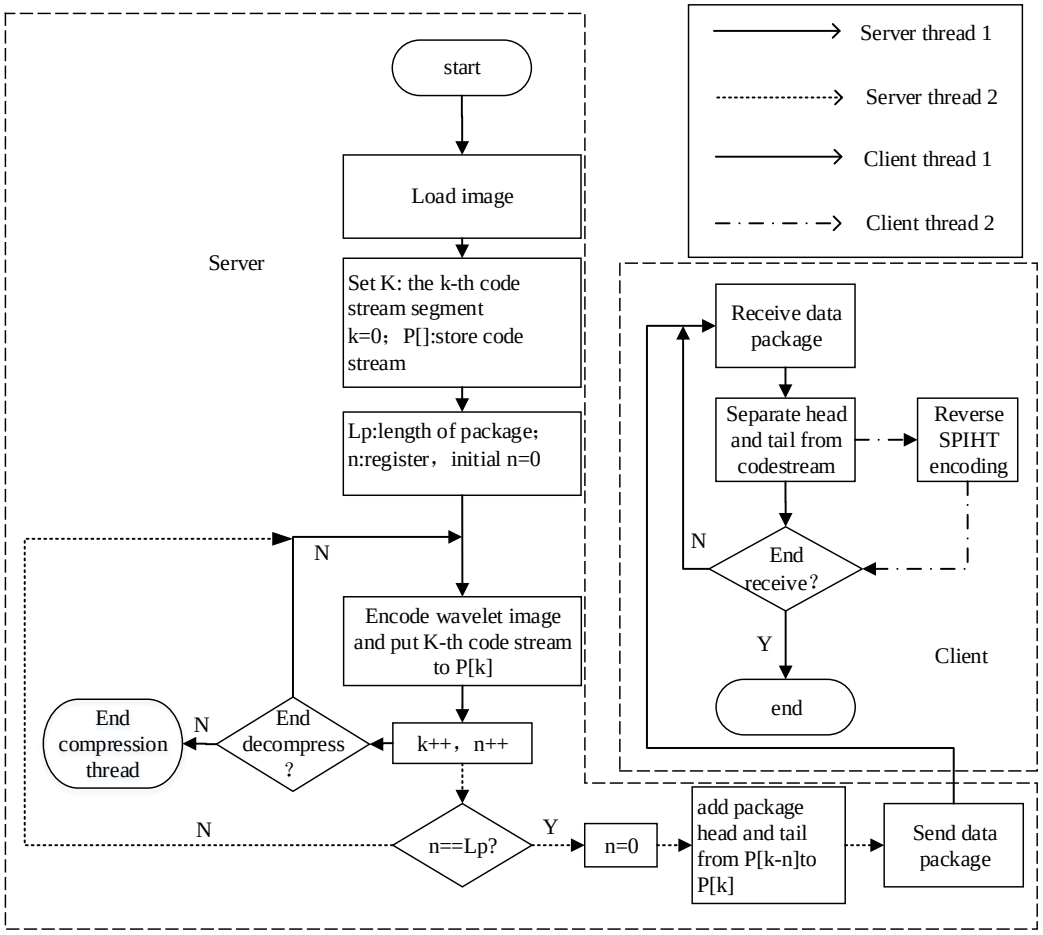

Figure 8. Online Compression and Decompression Flow Chart

4. Experimental Result and Analysis

4.1. Experimental Environment and Data

We use two computers to simulate the Client and the Server separately in intranet environment.
Experimental data: There are several sets of remote sensing images which come from Google Earth and WorldView-II. In our experiment, a piece of remote sensing grey-scale image A with size of 2048x2048 was pitched on in order to simplify procedure. Image B and C have the same size as A, but different scene contents for contrast experiment.
Software environment:
Operating System: Windows 7
Development platform: Visual studio platform 2010, C++ language with MFC
Database: Microsoft SQL Server 2008
Hardware environment
CPU: Intel Core i7-3820QM CPU @ 2.70GHz
Memory: DDR-3 8 GB Memories

4.2. Experimental Results

There are some experimental results to show transmission effect using proposed method. The experimental results conclude four parts as follow:

(1) Quality progressive transmission based on SPIHT algorithm

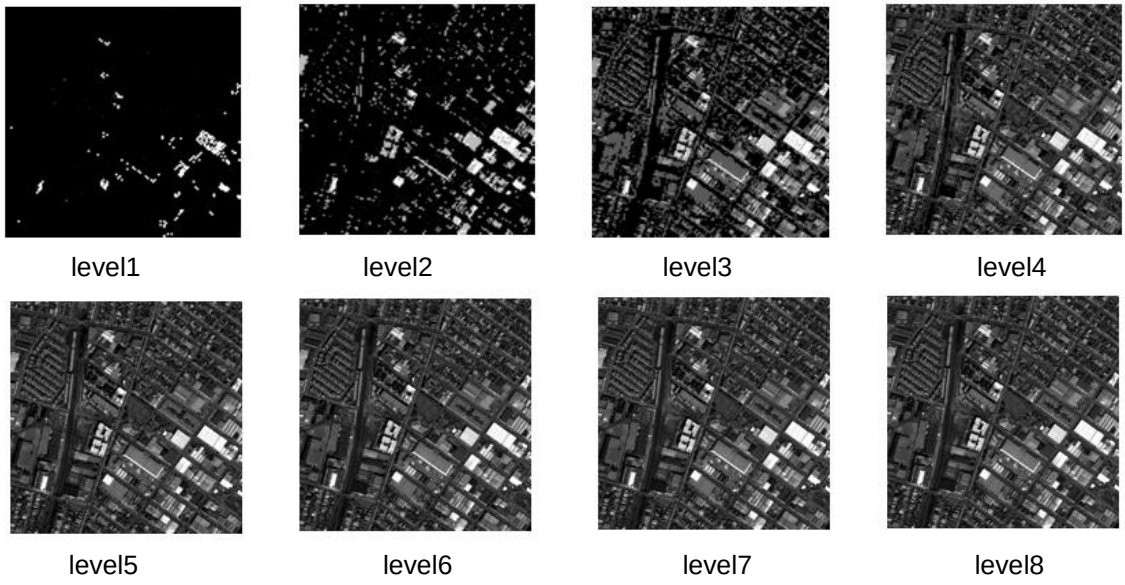

Figure 9. Eight Levels' Progressive Show for the Remote Sensing Image of Diaoyu Island based on Quality

From level1 to level 8 in Figure 9, we can see that the constructed image became finer and finer. The transmission can be terminated at any level according to the user's requirements. If user wants to view finer image, the client can continue to transmit code stream from termination without from start.

(2) Resolution progressive transmission based on JPEG2000 algorithm

In Figure 9, the visualization is based on quality progressive transmission. Another visual format based on resolution progressive transmission is shown in Figure 10.The size of level8 is 2048x2048, the size of level7 is 1024x1024, and then the size of level1 is

16x16. We can see that, the image became finer and finer with the increasing of resolution from 16x16 to 2048x2048.

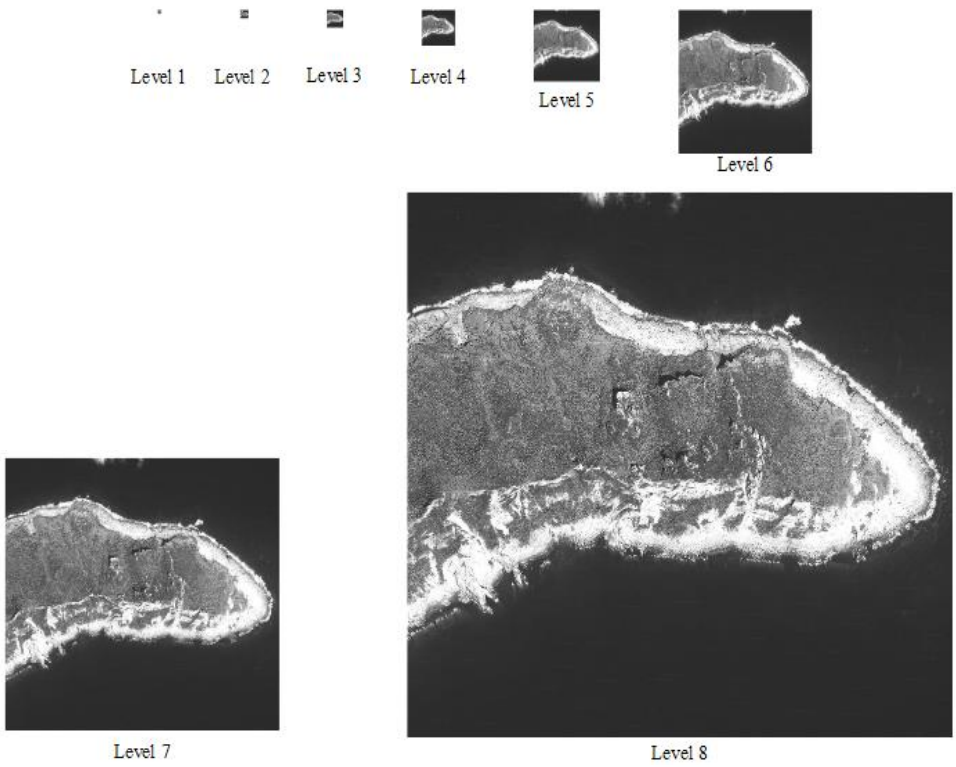

Figure 10. Eight Levels' Progressive Show for the Remote Sensing Image of Diaoyu Island based on Resolution

(3) The main purpose of progressive transmission is to reduce the workload of data transmission without affecting the display effect. First we use SSIM (structural similarity index measurement) which is a method for measuring the similarity between two images to measure the similarity between different level images. The SSIM index can be viewed as a quality measure of one of the images being compared and provided the other image is regarded as of perfect quality. The comparative result can be seen in Figure 11.

From Figure 11, we can see that, it is not possible to differentiate from level 4 to level 8. If the client users only want to browse the image, four levels of code streams are enough. Hence much image data can be reduced to transmit. The length of every level code stream is shown in Figure 12. The horizontal axis shows levels in type of Integer. And the vertical axis shows the length of each level code stream in Bytes. From level1 to level4, the total lengths of code streams are 320370 Bytes, 187192 Bytes and 254024 Bytes, occupying 12.4%, 14.14% and 10.76% of the total length of code streams separately using SPIHT algorithm.

Figure 11. SSIM Index for Different Images

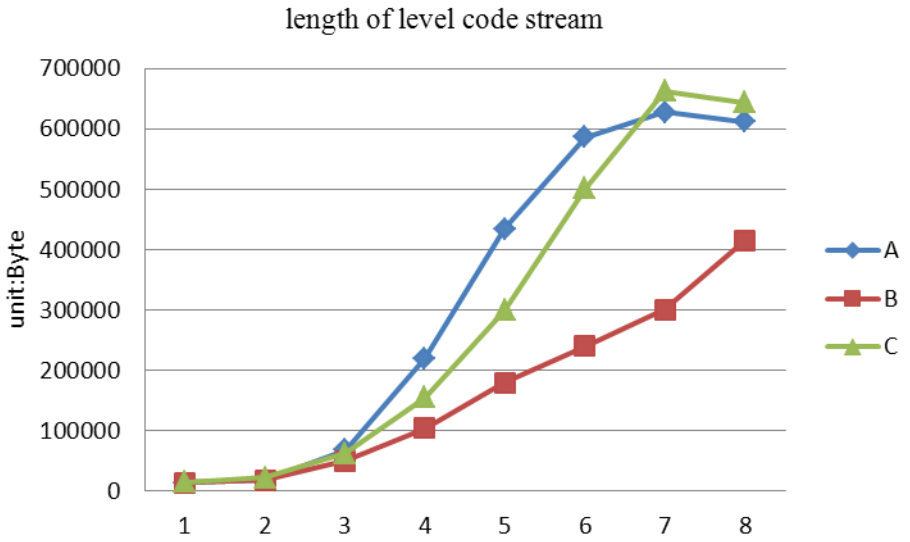

Figure 12. Length of Level Code Stream

(4) In order to obtain comparable data, several group experiments have been implemented. The acceleration results and reduction of image data transmission are given in Table 2.

From Table 2, it can be seen that the whole processing speed has been improved nearly twice without reducing image transmission quality by using the proposed progressive transmission and real-time compression model. The transmission time was reduced dramatically through synchronizing compression, decompression with transmission. With the increasing size of remote sensing image, the acceleration is more and more obvious especially in heterogeneous network with low bandwidth. The comparison of data size with retry broken downloads and data size without retry broken downloads for each compression level shown in Figure 13.

Table 2. Time of Real-Time Progressive Transmission Based on Multi-Threads and Single-Thread Progressive Transmission

Data \ Bit-plane		1	2	3	4	5	6	7	8	Total number
Data packages		15	15	22	40	79	166	318	548	1203
Data size with retry broken downloads (KB)		15	15	22	40	79	166	318	548	1203
Data size without retry broken downloads (KB)		15	30	52	92	171	337	655	1203	1203
Single thread imple ment	Compression time(ms)	218.5	78	133	250	414	906.5	1648	2273.5	5921.5
	Transmission time(ms)	22.2	24.8	22	36.2	68.2	159.8	332.6	674.8	1340.6
	Decompression time(ms)	601.6	514.3	561.3	633	792.3	1155.3	1718	2174.3	8150.3
	Total time (ms)	842.3	617.1	716.3	919.2	1274.5	2211.6	3689.6	5122.6	15412.4
Multithread implement time(ms)		609.5	500	531.5	625	781.5	1171.5	1922	2702.5	8843.5
Speedup		1.4	1.2	1.3	1.4	1.6	1.9	1.9	1.9	1.7

Annotate: Length of data package L_p is defined as this: $Lp = L_h + L_d + L_t$ where L_h length of package head with size of 16Bytes; L_d: length of code stream data package; L_t: length of package tail with size of 16Bytes. Package head and package tail include control information.

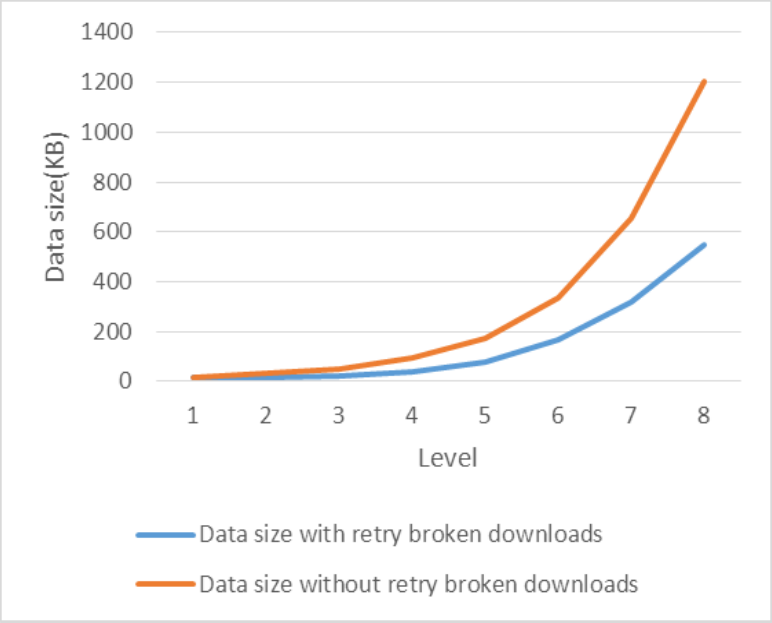

Figure 13. Comparison of Data Size with Retry Broken Downloads and Data Size without Retry Broken Downloads for Each Compression Level

5. Conclusions

In this paper, an online remote sensing image progressive transmission model using pipeline-based multi-threads acceleration was constructed in which remote sensing image compression and decompression are synchronized with transmission. A new scheme aim at overcoming long time waiting for a high resolution image and improving user interactive experience was proposed with retry broken downloads in this model. The whole processing speed has been improved nearly twice based on SPIHT algorithm without reducing image transmission quality by using the proposed progressive transmission and real-time compression model. The whole transmission data is reduced nearly by half using the proposed method.

In future, we will construct a self-adapting online remote sensing image progressive transmission system in which the reduction ratio parameter can be generated automatically based on SSIM index. On the other hand, in order to accelerate the compression and decompression processing, we are currently working on efficient parallel implementations using specialized hardware devices such as commodity graphics processing units (GPUs) [12-16] and field-programmable gate array (FPGA) [17].

Acknowledgments

This work was supported in part by the general project of scientific research of the Education Department of Liaoning Province under grants L2015216 and National Natural Science Foundation of China (No. 61172144).

References

[1] W. Zhang, L. Wang, D. Liu, W. Song, Y. Ma, P. Liu and D. Chen, “Towards building a multi-datacenter infrastructure for massive remote sensing image processing”, Concurrency and Computation: Practice and Experience, vol. 25, no. 12, **(2013)**, pp. 1798-1812.
[2] K. H. Tzou, “Progressive image transmission: a review and comparison of techniques”,Optical Engineering, vol. 26, no.7, **(1986)**, pp. 581-589.
[3] B. C. Dhara and B. Chanda, “A fast progressive image transmission scheme using block truncation coding by pattern fitting”, Journal of Visual Communication and Image Representation, vol. 23, no. 2, **(2012)**, pp. 313-322.
[4] C. P. Chu, S. H. Hwang, S. C. Chang and C. Y. Yeh, “An Improvement to Tree-Structured Vector Quantization”, Applied Mechanics and Materials, vol. 284-287, **(2013)**, pp. 2926-2929.
[5] A. C. Miguel, A. E. Mohr and E. A. Riskin, “SPIHT for generalized multiple descriptions coding”, Proc.1999 International Conference on. IEEE, Kobe, Japan, **(1999)** October 24-28.
[6] B. Mohammed, H. Yassine, M. L. Abdelmouneim, A. Bassou and T. A. Abdelmalik, “The Performance of Discret Bandelet Transform Coupled by SPIHT Coder to Improve the Visuel Quality of Biomedical”, Color Image Compression, vol. 6, no. 5, **(2014)**, pp. 64-72.
[7] G. Madhuri and V. V. M. Krishna, “An SPIHT Algorithm with Huffman Encoder for Image Compression and Quality Improvement”, vol. 2, no. 9, **(2013)**, pp. 361-365.
[8] S. H. abhole, V. A. Gundale and J. Potgieter, “An efficient modified structure of CDF 9/7 Wavelet based on adaptive lifting with SPIHT for lossy to lossless image compression”, Signal Processing Image Processing & Pattern Recognition (ICSIPR), 2013 International Conference on. IEEE, Coimbatore, India, **(2013)** February 7-8.
[9] M. W. Marcellin, M. J. Gormish, A. Bilgin and M. P. Boliek, “An overview of JPEG-2000”, Data Compression Conference, Proc. DCC 2000. IEEE, Snowbird, UT, **(2000)** March 28-30.
[10] S. Deng, H. Qian and B. Sun, “Research on Progressive Coding of Remote Image Based on Region of Interest”, Proc.Remote Sensing, Environment and Transportation Engineering (RSETE), 2012 2nd International Conference on, Nanjing, China, **(2012)** June 1-3.
[11] Y. P. Zheng, Z. J. Li and M. Sarem, “An Improved STC Image Compression Method”, Applied Mechanics and Materials,Papers , vol. 143-144, **(2012)**, pp. 742-745.
[12] C. Song, Y. Li and B. Huang, “A GPU-accelerated wavelet decompression system with SPIHT and Reed-Solomon decoding for satellite images”, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 4, no. 3, **(2011)**, pp. 683-690.
[13] L. Santos, E. Magli, R. Vitulli, J. F. López and R. Sarmiento, “Highly-parallel GPU architecture for lossy hyperspectral image compression”, vol. 6, no. 2, **(2013)**, pp. 670-681.

[14] Y. Yao and J. Luo, "Top-down progressive computing", Rough Sets and Knowledge Technology. Springer Berlin Heidelberg, vol. 6594, **(2011)**, pp. 734-742.
[15] Y. Liu, M. Lu, T. Liu and H. Xiang, "Research on the key techniques of remote sensing information mobile services", The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Services, vol. 37, **(2011)**, pp. 917-921.
[16] H. Qu, J. Zhang, Z. Lin and H. Chen, "Parallel Acceleration of SAM Algorithm and Performance Analysis", IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 6, no. 3, **(2013)**, pp. 1172-1178.
[17] I. V. Anjaneyulu and P. R.Krishna, "FPGA Implementation of DWT-SPIHT Algorithm for Image Compression", vol. 2, no. 3, **(2014)**, pp. 20-24.

Authors

Haicheng Qu, received the B.S. degree in computer science and technology from Qingdao Technological University, China, in 2005, and the M.S. degree in computer application technology from Liaoning Technical University, Huludao, China in 2008. Currently, he is pursuing the Ph.D. degree in signal and information system from Harbin institute of technology.His research interests include hyperspectral image processing, software architecture technology and high performance computing.

Junping Zhang, received the B.S. degree in biomedical engineering and instrument from Harbin Engineering University and Harbin Medical University, Harbin, China, in 1993, and the M.S. and Ph.D. degrees in signal and information processing from the Harbin Institute of Technology (HIT), in 1998 and 2002, respectively. She is currently a professor with the Department of Information Engineering, School of Electronics and Information Engineering, HIT. Her research interests include hyperspectral data analysis and image processing, multisource information fusion, pattern recognition and classification.

Yu Meng, received the B.S. degree in software engineering and the M.S. degree in computer technology from Liaoning Technical University, Huludao, China.Currently, he is pursuing the Ph.D degree in computer application technology from Northeastern University.His research interests include Data mining,cloud computing,remote sensing transmission, real-time vision tracking and high performance computing.