A review of parallel computing for large-scale remote sensing image mosaicking

Noname manuscript No.
(will be inserted by the editor)

A Review of Parallel Computing for Large-scale Remote Sensing Image Mosaicking

Lajiao Chen · Yan Ma · Peng Liu · Jingbo Wei · Wei Jie · Jijun He

Received: date / Accepted: date

Abstract Interest in image mosaicking has been spurred by a wide variety of research and management needs. However, for large-scale applications, remote sensing image mosaicking usually requires significant computational capabilities. Several studies have attempted to apply parallel computing to improve image mosaicking algorithms and to speed up the calculation process. The state of art of this field has not yet been summarized, which is, however, essential for a better understanding and for further research of image mosaicking parallelism on a large scale. This paper provides a perspective on the current state of image mosaicking parallelization for large scale application. We firstly introduce the motivation of image mosaicking parallel for large scale application, and analysis the difficulty and problem of parallel image mosaicking at large scale such as scheduling with huge number of dependent tasks, programming with multiple-step procedure, dealing with frequent I/O operation.. Then the we summarize the current state of parallel computing in image mosaicking for large scale applications with respect to problem decomposition and parallel strategy, parallel architecture, task schedule strategy and implementation of image mosaicking parallelization. Finally, the key problems and future potential research directions for image mosaics are addressed.

Keywords Image mosaicking · parallel computing · review

Lajiao Chen, Yan Ma, Peng Liu, and Lingjun Zhao
Institute of Remote Sensing and Digital Earth,Chinese Academy of Sciences,Beijing 10094, China
Tel.: +861082178973
E-mail: chenlajiao@ceode.ac.cn

1 Introduction

Image mosaicking is the process of combining multiple images with overlapping regions into a single seamless composite image [1]. It is an essential task in remote sensing and has been widely used in many fields since a single scene usually cannot cover large spatial extents of interest. In recent years, with our environment undergoing rapid changes, there has been a correspondingly urgent demand for accurate large area information for environmental monitoring, disaster assessment, biodiversity conservation, etc [2-5]. This has highly promoted image mosaicking for large scale applications.

Interest in image mosaicking has been spurred by a wide variety of research and management needs. Since the 1980s, organizations, such as the National Oceanic and Atmospheric Administration (NOAA), the United States Geological Survey (USGS), and international programs such as the International Geosphere Biosphere Programme (IGBP), and the Global Observation of Forest Cover (GOFC), have started activities on large scale image mosaics [6-9]. For example, in 1985, a joint project between the NOAA, the USGS, and the National Remote Sensing Centre (NRSC) of UK completed a NOAA-AVHRR mosaic covering Antarctica [10-11]. The Global Rain Forest Mapping (GRFM), an international endeavor led by the National Space Development Agency of Japan (NASDA), aimed to produce semi-continental, 100m resolution, image mosaic over the tropical belt on the Earth [12]. In addition, the need for real-time information for environmental management is an essential motivation for large-scale image mosaicking. In some circumstance such as large-scale flood prediction and management, how to provide real time and accurate information is urgently needed.

However, for large-scale applications, remote sensing image mosaicking usually requires significant computational capabilities. This is due to the fact that for large scale applications, remote sensing image mosaicking is not only a data-intensive task but also a computation-intensive task. The computation problem of sequential image mosaicking for large scale application has merged as the one of the urgent problem. Several studies have sated the huge computational requirements of large-scale image mosaics. Roseqvist et al [12] stated that, in the GRFM project, the image mosaicking of Africa which involed 3600 senses of images required some 20 hours of computation time. Shusun et al [13] find out that the correction of a 100-m SAR image took 30 minutes using a SPARC1k workstation at the ASF Interactive Image Analysis System (IIAS) Laboratory. Such a processing speed is too slow to generate a mosaic of the state of Alaska dealing with about 800 images. In this regard, the traditional sequential computation techniques can hardly meet the requirements of large-scale image mosaicking applications. Therefore, it is essential to apply the HPC techniques such as parallel computing to assist speed up computation of image mosaicking.

Table 1 Typical large-scale applications of image mosaicking

Product	Geographic scale	Dataset	Resolution (m)	Count of images	Project
North America	Regional	TM and ETM	30	2100	LEDAPS
Alaska	Regional	ERS-1 SAR	100	800	-
Canada	National	AVHRR	1000	800	CCRS
China	National	Beijing-1	32	1000	-
China	National	CEBERS-2	19.5	1300	-
Antarctica	Continental	TM	30	1100	USGS
Equatorial Africa	Continental	JERS-1 SAR	100	3600	GRFM
South and Central America	Continental	JERS-1 SAR	100	5000	GRFM
South-East Asia and North Australia	Continental	JERS-1 SAR	100	4300	GRFM

To address the problems of huge computation demand with its complicated algorithms and massive date amounts, High Performance Computing (HPC) such as parallel computing has been considered to be an effective solution. Several studies have attempted to apply parallel computing to improve image mosaicking algorithms and to speed up the calculation process. Despite these research, there still remain a number of challenges with respect to how to handle huge number of images, how to implement the processing tasks with complicate denpendency, how to conduct the parallel programming with easy logic, and so on. To date, the state-of-art of this field has not yet been summarized, which is, however, essential for a better understanding and for further research of image mosaics on a large scale.

The purpose of this paper is to present the current sate of parallel computing for large-scale remote sensing image mosaicking. We firstly present the multiple-stage procedure of image mosaicking and then illustrate the motivation of image mosaicking for large scale application. Then the we summarize the current state of parallel computing in image mosaicking for large scale applications. Finally, the key problems and future potential research directions and visions for image mosaics are addressed.

2 Motivation of image mosaicking parallelism

Motivated by global change, organizations, such as NOAA and USGS, and international programs such as IGBP and GOFC, have launched many activities with large scale image mosaicking. A growing number of projects and studies are focusing on large scale image mosaics. The image mosaicking for large scale in recent years are summarized in Table 1.

2.1 Regional scale applications

Regional image mosaics are increasingly being developed recently to meet national monitoring and reporting needs. Now that Landsat is available, it has

been widely used to construct regional scale images [14-16]. Large volumes of image mosaics were constructed using Landsat Data [17-19]. For example, Landsat Ecosystem Disturbance Adaptive Processing System (LEDAPS), processed over 2 100 TM and ETM+ acquisitions to provide wall-to-wall surface reflectance coverage for North America for the 1990s and 2000s [19]. Recently, SAR data have been widely used to generate medium-resolution radar mosaics. For example, Shusun [13] generated a terrain-corrected SAR mosaic of Alaska using 800 ERS-1 SAR images from 1992 to 1993.

2.2 National scale applications

On national scale, merging images is keenly needed for national environmental policy design. In United States, attentions have intensively been paid to the national land cover image mosaicking using Landsat data [20-22]. In Canada, a Canada-wide mosaic was conducted using using 800 NOAA/AVHRR daily mosaics [23]. They are the first composite AVHRR scenes acquired over Canada on a given day. In China, attempts have been made to conduct national images based on satellite data, such as CBERS-1 (China-Brazil Earth Resources Satellite) and Beijing-1 [24-25]. For example, Wang et al. [25] composited a China-wide mosaic map based on a Beijing-1 small satellite, which is in current operation, and which will be used in the many fields of national environment conservation.

2.3 Continental or global scale applications

On Continental-scale, the Antarctic has been a hot spot for image mosaicking applications, fueled by the desire to reveal its unknown features [26-27]. In 1985, a joint project of NOAA, USGS, and National Remote Sensing Centre of UK (NRSC, UK) completed a mosaic covering Antarctica [10, 29]. A total of 28 three-band AVHRR scenes were used in this project, and a provisional mosaic image of Antarctica was produced with a resolution of 1 km and a scale of 1: 5 000 000. In 1997, Radarsat-1 SAR data was used to create the first high-resolution (25 m) radar image mosaic of the continent [30]. More recently, Scambos et al. [31] presented digital image mosaics for the Antarctic continent and its surrounding islands, assembled from 260 Moderate Resolution Imaging Spectroradiometer (MODIS) images. Bindschadler et al. [32] generated a Landsat mosaic of Antarctica from nearly 1100 Landsat ETM+ austral summer acquisitions, which provides the first true-color, high-spatial-resolution images of the continent.

Despite Antarctic, image mosaics for other continents are also promoted in recent years. In 2001, a cloud-free mosaic of Australian continent was assembled using images from the Along Track Scanning Radiometer (ATSR-2) onboard the European Remote Sensing satellite (ERS-2) with a spatial resolution of 1 km [33]. The image is 5001 pixels 4001 pixels in size. The GRFM

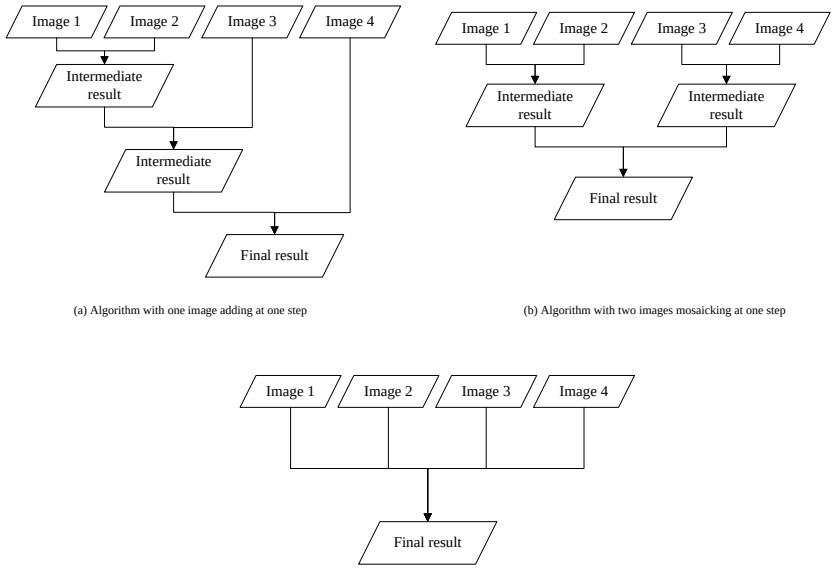
African mosaic was assembled at the JRC [34] in the framework of the JAXA GRFM project. It covers the central part of the African continent. Since global scale maps are important information sources for global change research, they have received particular attention in recent years [12, 36-37]. For example, the Global Land Survey project is now makingefforts to create a global-scale DEM (GLSDEM) [38]. Once the problems of data acquisition and computational efficiency have been overcome, there will be an increasing number of global-scale mosaickingimages developed for planetaryenvironmental change research.

The computation problem of sequential image mosaicking for large scale application has merged as the one of the urgent problem. Several studies have sated the huge computational requirements of large-scale image mosaics. Shusun et al [13] find out that the correction of a 100-m SAR image took 30 minutes using a SPARC1k workstation at the ASF Interactive Image Analysis System (IIAS) Laboratory. Such a processing speed is too slow to generate a mosaic of the state of Alaska dealing with about 800 images. Roseqvist et al [12] stated that, in the GRFM project, the image mosaicking of Africa which involed 3600 senses of images required some 20 hours of computation time. In this regard, the traditional sequential computation techniques can hardly meet the requirements of large-scale image mosaicking applications. Therefore, it is essential to apply the HPC techniques such as parallel computing to assist speed up computation of image mosaicking.

3 Difficulty of image mosaicking parallelism for large scale applications

3.1 Difficulty in dealing with huge number of dependent tasks

For large scale application, image mosaicking has to deal with extremely huge number of images, especially on continental or global scale. With the resolution of the image increasing, the number of images to cover the whole area increase dramatically. For example, the GRFM project conducted many image mosaicking in Continental scale, such as north America, Africa. For the case of Africa (excluding Madagascar which is treated separately), the image mosaicking involves some 3600 scenes, resulting in a normal equation matrix larger than 10 000 lines by 10 000 columns [12], which requires some 20 hours of computation time. Several studies have sated the huge computational requirements of large-scale image mosaics. Shusun et al [13] find out that the correction of a 100-m SAR image took 30 minutes using a SPARC1k workstation at the ASF Interactive Image Analysis System (IIAS) Laboratory. Such a processing speed is too slow to generate a mosaic of the state of Alaska dealing with about 800 images. Roseqvist et al [12] stated that, in the GRFM project, the image mosaicking of Africa which involed 3600 senses of images required some 20 hours of computation time. Such large numbers of RS images make



1.pdf

Fig. 1 Image dependency of mosaicking with different algorithm

the traditional mosaic on basis of scene-by-scene no longer inapplicable on parallel system due to the intolerable time consumption and inevitable poor scalability with increasing processors.

Image mosaicking has to deal with georeferenced RS images that have overlapping regions. One image have to operated with the adjacent images with respect to image registration, seamline detection, image blending. Fig.2 shows the different image dependent relationship and processing execution order by different mosaicking algorithm. For example, in Fig.1(a), Image 3 can be processed only Image 1 and Image 2 have finished while Image 4 can only be processed when Image 3 has been processed.

Such interdependency among huge number of RS images give rise to the complexities of image mosaicking parallelism. Firstly, due to the adjacent relationships among images and their determined processing order, the task partition need be conducted recursively according to the intricate adjacent relationships among images. As a result, the task partition becomes a problem of properly representing large scale mosaicking in the form of data-driven task graph (DAG) which consists of a large collection of interdependent tasks. Secondly, the interdependency among huge number of RS images give rise to the complexities of image mosaicking parallelism. How to arrange such a huge number of tasks into an efficient processing order so as to gain low completion time becomes a challenge for image mosaicking parallelism.

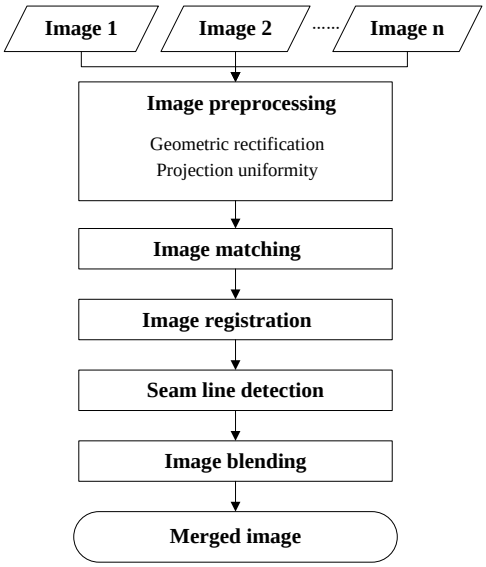
3.2 Difficulty in programming with multiple-step procedure

Image mosaicking is a complex multi-stage processing procedure, which mainly consists of five computational processing performed in order (Fig.2): 1) Image preprocessing: Image preprocessing commonly includes geometric rectification and projection uniformity, assuring input images to a common spatial scale, coordinate system, and projection. 2) Image registration: Registration is applied for automatic control point abstraction and geometric transformation between a pair of images with overlapping region. 3)Image matching: Image matching is applied to reduce the radiometric differences among images that are shot at the same time with different equipment, or at a different time with the same equipment. 4) Seam line detection and image blending: Seam line detection together with blending is employed for eliminating the artificial edges in the overlapping region introduced by the radiometric discontinuous between images.

With the complex and intensive interaction among images, it seriously increase the difficulty of parallel implementation. The tasks need to be executed with an order constraint. In such circumstance, some certain tasks have to wait for the up-stage tasks to be available. For example, the task of image blending have to wait for the task of seam line detection to be finished. This would subsequently lead to frequent and trivial process synchronization and data communication. Thus, it gives rise to high complexity for parallel programme based on traditional parallel computational mode such as open multi-processing (OpenMP) or message-passing programming model (MPI). The programming has to concern frequent and trivial communication among processes which makes the program instable and low efficient. Therefore, further studies should be undertaken to develop a parallel computing approach, which can effectively organise huge amounts of images with easy logic control and parallel programming.

3.3 Difficulty in dealing with frequent I/O operation

Image mosaicking for large scale area has to deal with frequent image loading and exporting massive dataset, which will introduce intensive data I/O operations and also undesirable I/O overhead. In this case, the data processing has to wait a plenty of CPU cycles for data accessing which will introduce intensive I/O operations and undesirable I/O overhead. The I/O operation may be a considerable constrain of mosaicking parallelism as it is time consuming and may cause system collapse if it has not been properly operated. Despite several studies have considered parallel I/O such as by using parallel filesystem, most of the existing studies have not fully considered parallel I/O. In the future work, more attention should be paid to the parallel I/O operation.



2.pdf

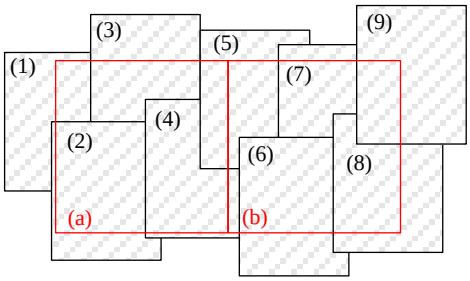
Fig. 2 Multiple-stage procedure of image mosaicking

4 Existing research on parallelization of image mosaicking

Though great efforts have been paid to RS image parallel processing[39-42], parallelization of image mosaicking is rarely pay attention to and only a handful of related research have attempted to apply parallel computing to speed up the processing of image mosaicking [43-49].

4.1 Problem decomposition and parallel strategy

Problem decomposition is the base of image mosaicking parallelization. Data decomposition and function decomposition are two general approaches to problem decomposition for the distribution of tasks between multiple processors. Due to the fact that image mosaicking algorithm involves of huge volume of images and each images have the same calculation procedure, the images for mosaicking can be partitioned into pieces and distributed to a separate processor. According to the independent relationship of images in mosaic processing, the steps of mosaicking processing can summarized into two type: independent procedure such image preprocessing and neighbourhood-dependent procedure such as image registration. For the independent processing, each image are calculated independently without connection with each other, all of these images can calculated simultaneously. For the neighbourhood-dependent procedure, therefore, the images without overlap area can be calculated simultaneously. With the image dependent and procedure constrain, the difficult of neighbourhood-dependent procedure parallelization is how to partition the



3.pdf

Fig. 3 Data decomposition with tracing to original images

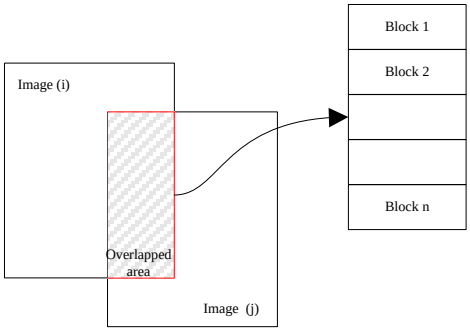
images into pieces with minimum interaction and load balancing.

4.1.1 Data decomposition ignoring image dependency

In this type, images are simply partitioned with given scope and the data blocks are then distributed to different computation node for calculation. The partition grain is usually coarse in size. Communication among the data block is not fairly considered in this data partition scheme which will decrease parallel efficiency [43-46].

For example, An et al [43] proposed a data decomposition scheme which partition the images according to the spatial scope final mosaicking image Fig.3. This method firstly divides the final spatial scope of the mosaic into grids with same size (red square in Fig. 3) and the original images involved in each grid (black rectangles with serial number) are traced according to their geographic coordinate. Those images related to one grid are treated as one data block and each data block can be processed simultaneously. As shown in Fig.3, Image (1), (2),(3), (4),(5) related to Grid (a) are treated as one data chunk while Image (4), (5),(6), (7),(8),(9) related to Grid (b) are treated as another data chunk. Due to the variation of overlapped area among the images, the grid with same size may involved with different amount of images. Take Fig.3 for example, Grid (a) involves 5 images while Grid (b) involves 6 images. Therefore, to avoid load unbalance and reduce calculation waiting time, a appreciated task schedule strategy is needed.

Hu et al [44] proposed a vertical partition strategy for parallel of a PCI image fusion algorithm. In this approach, image are partitioned with a vertical partition strategy, and each data block are in same size. The number of the data block is determined by the number of computation notes so each computation node can handle with one data block. Though the images are partitioned into same size of block, it still will cause load unbalance as the same-size data blocks may demand different calculation time due to the different overlapping areas. In addition, this partition will bring frequent communication among



4.pdf

Fig. 4 Data decomposition with overlapped partition

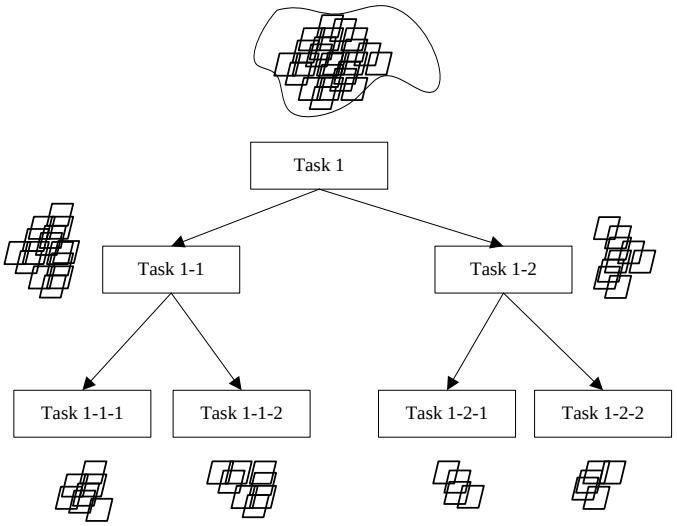
data blocks leading to very complicated control logic, poor stability and extensibility of parallel computing.

4.1.2 Data decomposition considering image dependency

In this kind of data decomposition, images are firstly partition into same blocks and in each block images are further partitioned into groups according to their overlapping relationship. In such a way, tasks are partitioned into sub-image scale which can highly promote the parallel. With respect to different image mosaicking algorithm, different fine-grain data decomposition strategy are proposed.

Wang et al. [48] proposed a fine-grain data decomposition scheme based on the overlapped area of images (Fig.4). This strategy firstly partitions all of image pair with overlapped area and each image pair without dependency can be processed simultaneously. The overlapped area of every image-couple are further subdivided into data blocks with same size, and each chunk is distributed into separate processor for calculation.

Ma et al [49] developed data partition scheme based on the top-down task partition approach which recursively partitions the image mosaicking into independent tasks until the tasks can not be divided (Fig. 5). An adjacent table is used to represent the adjacent relation among images. The overlapped region of each image pair is also subdivided into pieces for calculation similar to the Wang et al's method. Though this kind of partition can gain a fairly fine parallel grain, it may introduce huge communication overhead among data blocks. Therefore, it is a severe challenge to well organise and schedule these tasks.



5.pdf

Fig. 5 Data partition with top-down strategy

4.2 Architecture of image mosaicking parallelization

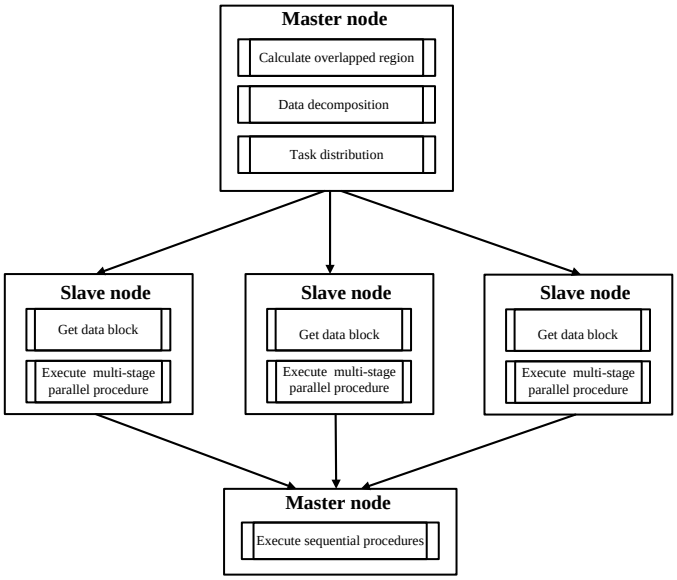
According to the characteristic of the data decomposition scheme, the architecture of image mosaicking parallelization varies. The following text introduce the main architectures that are adopted.

4.2.1 Master-slave parallel structure

Nearly all of the existing studies use the master-slave parallel structure for image mosaicking parallelization [42-49]. This is due to the fact that image mosaicking is a multiple-stage process with some of the procedure are global dependent which can not be processed parallelled. Therefore, a master node is needed to gather intermediate results and proceed to sequential computation. Fig. 6 illustrates the processing of image mosaicking with master-slave structure. A master node is responsible for data partition, data allocation, sequential procedures computation, etc, while the slave nodes follows a multi-stage processing procedure which can be parallel computed. There is no communication and interaction among slave nodes.

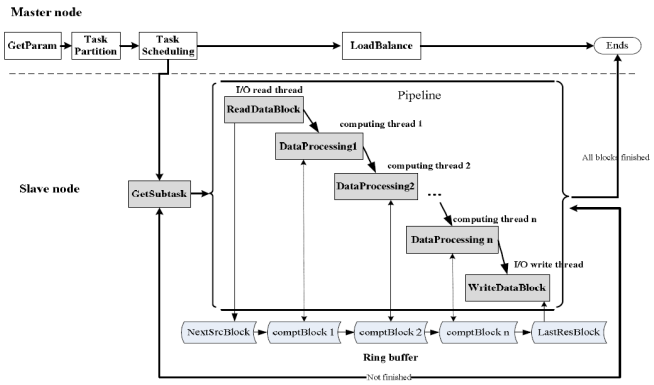
4.2.2 Parallel I/O structure

Though I/O overhead is one of the big problem of parallel image mosaicking, only a very limited studies have tried to solve the problem of intensive I/O operations and undesirable I/O overhead. Wang et al [48] design the pipeline among data reading, data processing, data-stage out. Three threads are responsible for the operation: I/O reading thread which is to get block area in



6.pdf

Fig. 6 Processing of image mosaicking with master-slave structure



7.pdf

Fig. 7 Processing of image mosaicking with master-slave structure

input image, the next block and then read the data in that block area, the block processing thread which performs image mosaicking for current block, I/O write thread which write the result data of previous block. Ma et al [49] adopted a SAN storage system equipped with high performance parallel file system Lustre to handle fast image data staging in and out among tasks. This SAN storage is fully connected with all computation resource.

4.3 Task schedule strategy of image mosaicking parallelization

With respect to the characteristic of task scheduling scheme, the existing parallelization image mosaicking can be summarized into three types: parallelization with prescribed task distribution without scheduling, parallelization with static task scheduling strategy, parallelization with dynamic task scheduling strategy.

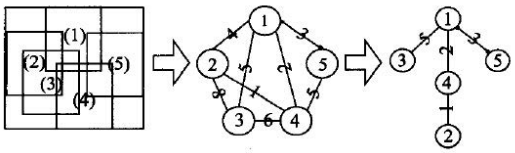
4.3.1 Parallelization with prescribed task distribution without scheduling

In this type, images are partitioned into blocks using coarse-grained data decomposition with prescribed scope stated in 4.1.1. The number of the data block are determined by the number of computation nodes [44][46-47]. For example, in Hu et al.[44]'s work, images are partitioned with a vertical partition strategy and each data blocks are distributed to computation node for calculation. Despite of the simplicity of this kind of strategy, there are two fundamental defect of this strategy: firstly, the parallel efficiency is low because the granularity of parallelization is coarse with only a few of data blocks; secondly, it is prone to load unbalance because the task distribution scheme is static which does not consider the current condition of the computation nodes. The tasks need to be divided up evenly distribute, because the speed of the whole program depend on the time taken by the processor takes the longest time.

4.3.2 Parallelization with static task scheduling

In this kind of strategy, each task is assigned a given (static) optimal priority for scheduling and allocated to processors considering the status of the processor [48]. The optimal task scheduling scheme is calculated in advance with the information of the structure of image mosaicking parallelization, the execution times of individual task and the communication cost between tasks. Such a scheduling scheme is applied for the entire scheduling process until the last task has been executed. Task scheduling is usually be in charged by MPI program instead of task schedular.

The key issue of this strategy is to how to present the priority of tasks so as to gain a minimal execution time for the whole mosaicking. In the work of Wang et al. [48], a minima1 spanning tree is designed to represent the priority of task list for scheduling. Such a tree is transformed from the map which shows the relationship of all the images with the overlapped area as the weight of the line (Fig.8). The tree gives an optimized image mosaicking sequence of these overlapped images. The computation nodes are then divided into groups according to the number of the branches in the tree. Each overlapped area is allocated to one group of computing nodes which could share image overlapped image data. For each node, mosaicking is parallel executed. In spite of the outstanding performance improvement, when applied to large scale with large collections of scenes, the overlapped regions increase sharply which could



8.pdf

Fig. 8 Minimal spanning tree (by Wang et al [64])

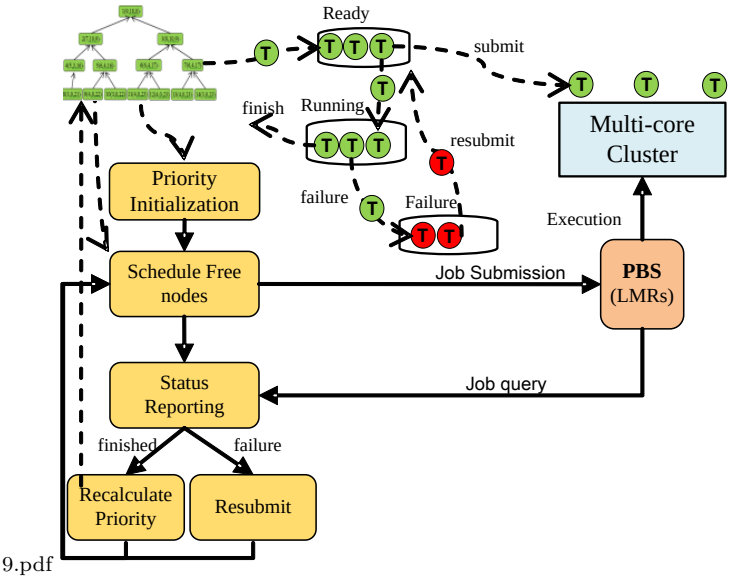
exceed thousands, even millions. To develop the minima1 spanning tree will be impossible as the memory of the computer can not handle. Therefore, such approach are only suitable for generating small regional mosaics, but not yet specialized in large-scale mosaicking.

An et al [43] proposed a task scheduling strategy using a double-buffer queue and a task selection strategy based on the complexity of task. In this approach, as mentioned in 4.1.1, the final spatial scope of the mosaic is divided into grids and the original RS images involved in each grid are treated as one task. The priority of each task is determined by the complexity of the task, which is defined as the pixel numbers of the images involved in this task. The tasks with highest complexity has the highest priority for execution. A greedy strategy is adopted to achieve the optimal task allocation order with the objective of reducing communication overhead and task-waiting time of computation nodes. The problem of this approach is the dependency among tasks are not well present for scheduling, therefore, the frequent communication cost highly influence the parallel efficiency. As showed in the results of the approach, with the increasing of the grid number, the computation time also decreases firstly, and then increases which is due to the communication cost. Therefore, if this approaches is applied for large scale mosaicking, the grid number increased extremely, the communication cost will will introduce huge communication overhead among tasks.

4.3.3 Parallelization with dynamic task scheduling

In dynamic scheduling, the task priority is dynamic according to current status of unexecuted tasks and status of computation nodes[49] [50]. That is, when one task has been executed, priority of the unexecuted tasks will be reassigned. A task schedular is adopted in charge of dynamically construct task queues and distributed the tasks to processors. The goal of dynamic scheduling is not only the minimization of the completion time but also the minimization of the scheduling overhead which constitutes a significant portion of the cost paid for running the scheduler.

Ma et al [49] propose a task-tree based scheduling strategy with dynamic DAG scheduling (Fig.9). The task dependency is presented by a task tree



9.pdf

Fig. 9 The CPDS-SQ DAG dynamic scheduling solution developed by Ma et al [65]

with minimal height. A critical path based dynamical DAG scheduling solution named CPDS-SQ is provided to offer an optimized schedule with minimal completion time. The scheduling starts with the entry nodes in the precedence-constraint DAG, which is also the leaf nodes of task tree. All the entry nodes are packaged into task packages by assigning the amount of computation resources, specifying the input image data and processing arguments. These task packages are then constructed as a list in descending order of priority and insert into a ready queue. The task packages in the ready queue are submitted to PBS a local resource managers of cluster system for concrete computation resource accommodation and job execution. When a task package is finished, then CPDS-SQ will move this task out from running queue and update the of the corresponding nodes with the real runtime of the task. Then the priority of the unscheduled nodes in DAG are recalculated and free the succeeding nodes of this finished node. This strategy can allocate the tasks according to the status of the computation nodes, which can highly reduce the probability of load unbalance. However, when extend to large scale application, the numbers of the images expands, how to construct the task tree with million of images will be a problem.

4.4 Implementation of image mosaicking parallelization

To implement parallelization of image mosaicking, the parallel paradigms such as MPI (Message Passing Interface), MPI + OpenMP hybrid parallel paradigm are applied. In the following text, we will summarize the main parallel paradigms

which have been applied to image mosaicking parallelization.

4.4.1 Image mosaicking parallelization implemented with MPI

MPI is the most widely used parallel paradigm for image mosaicking parallelization[50-52]. MPI has the advantage of high parallel efficiency, open and inter-platform, portable to multiprocessors, and so on. Merzky et al [51] and G.B. Berriman et al [52] illustrated that, the Montage, an astronomical image mosaicking software, using MPI versions of the computational intensive modules, has good performance. MPI parallelization reduces the one processor time of 453 minutes down to 23.5 minutes on 64 processors, for a speedup of 19. A main drawback of this kind of paradigm is that the frequency communication among the nodes makes the programming extremely complicate. However, due to the ordering constraint among tasks, if MPI is using for task scheduling, the programming becomes even more complex and tend to crash with the images increasing.

4.4.2 Image mosaicking implemented with MPI + OpenMP hybrid paradigm

A few studies has demonstrated using MPI + OpenMP hybrid paradigm to implement image mosaicking parallelization. Rabenseifner et al [53] illustrated that a hybrid MPI + OpenMP programming model can reduce communication need, memory consumption and improve load balance. In the task-tree based image mosaicking with dynamic DAG scheduling proposed by Ma et al [49], MPI was used to implement individual mosaic task among multiple processors while OpenMP was used to implement tasks among multiple multi-threading in each computation node. Wang et al [48] comparatively experimented the mosaicking algorithms with these three different parallel paradigms and all led to noticeable performance improvement. This kind of approach can promote the parallel efficiency as it take full advantage of multi-processors and multi-threading. However, it also will encounter the problem of MPI if it was used to schedule the ordering constraint of image mosaicking tasks.

4.4.3 Image mosaicking implemented with GPUs

In recent years, GPUs (Graphics Processing Units) with CUDA(Compute Unified Device Architecture) programming haveevolved into a highly parallel, multithreaded, many-core processors with tremendous computational speed and very high memory bandwidth [55]. Several relevant studies have illustrated GPU-based implementation of RS image processing[56], spatial data analysis [57]. In the field of parallel image mosaicking, Camargo et al [58] presented CUDA can significantly accelerated mosaicking for unmanned aircraft system. Yong et al [59] developed a fast colour balance adjstment of IKONOS

IMAGERY using CUDA. Results showed that, compared with conventional methods, color balancing with CUDA was able to produce images of similar quality in a much shorter time. However, for large scale image mosaicking applications, CUDA will counter problem in handling huge number of images and the complicate control logical of dependent tasks.

5 Conclusion and future work

This paper provides a perspective on the current state of image mosaicking parallelization for large scale application. We firstly introduce the motivation of image mosaicking parallel for large scale application, and analysis the difficulty and problem of parallel image mosaicking at large scale such as scheduling with huge number of dependent tasks, programming with multiple-step procedure, dealing with frequent I/O operation. Then the we summarize the current state of parallel computing in image mosaicking for large scale applications with respect to problem decomposition and parallel strategy, parallel architecture, task schedule strategy and implementation of image mosaicking parallelization.

Parallelization of image mosaicking for large scale application to date is still on its early stage, perspective for future work is stated as follow:

Firstly, a parallel image mosaicking program coded for a specific parallel computing platform often has limited portability to other parallel computing platforms due to not unified parallel programming or parallel hardware. However, the problem of the poor portability of parallel programs has yet been considered in parallel image mosicking. In the field of raster-based geocomputation, several efforts have been paid to overcome such poor portability. For example, Qin et al [60] demonstrate a strategy which is illustrated through the design and implementation of a set of PaRGO compatible with three popular types of parallel computing platforms. PaRGO encapsulates three types of parallel programming details in a form that is transparent to users. In such way, parallel raster-based geocomputation algorithms compatible with three popular parallel computing platforms can be easily and quickly developed. In the future work, such strategy from raster-based geocomputation could be borrowed to promote the portability of parallel programs for parallel image mosaicking

Secondly, data acquisition is one of the big problem for image mosaicking for large scale applications. To address this problem, remote sensing images from different sensors, different data center will be selected to cover the whole area of interest. However the existing mosaic parallelization solutions seldom consider parallel mosaicking with RS images from multiple source. Methodologies to produce image mosaic paralleled from multiple data center could be

an important field to be explored in the future.

Secondly, the lack of realtime images on a regional or national scale has often been cited as a current limitation [61]. However, the need for realtime information is often difficult to satisfy, particularly on national scale or even larger, which posed an awkward problem for large scale image mosaicking. In future research, efforts need to pay to improve the mosaic parallelization methodology concerning the capability of providing real time and accurate information on a large scale.

Acknowledgements This study is supported by National Natural Science Foundation (41301028), Project of State Key Laboratory of Resources and Environmental Information System (LREIS).

References

1. G. Moik, Digital processing of remotely sensed images, NASA (1980)
2. K. Jan, R. Andreas, C. R. Volker, K. Tobias, K. Jacek, H. Patrick, Land cover mapping of large areas using chain classification of neighboring Landsat satellite images, Int. J. Remote. Sens., 113, 957C964 (2009)
3. S. E. Franklin, and M. A. Wulder, Remote sensing methods in medium spatial resolution satellite data land cover classi-cation of large areas, Prog. Phys. Geog., 26, 173-205 (2002)
4. J. Cihlar, Land-Cover mapping of large areas from satellites: Status and research priorities. Int. J. Remote. Sens., 21, 1093-1114 (2000)
5. P. Coppin, I. Jonckheere, K. Nackaerts, B. Muys and E. Lambin , Digital change detection methods in ecosystem monitoring: a review, Int. J. Remote. Sens., 25, 1565-1596 (2004).
6. T. Hame, A. Salli, K. Andersson, and A. Lohi , A new methodology for the estimation of biomass of conifer dominated boreal forest using NOAA AVHRR data, Int. J. Remote. Sens., 18, 3211-3243 (1997)
7. G. De Grandi, J. P. Malingreau, M. Leysen, The ERS-1 Central Africa Mosaic: a new perspective in radar remote sensing for the global monitoring of vegetation, IEEE T. Geosci. Remote., 37, 1730-1746 (1997)
8. W. B. Cohen, T. K. Maiersperger, T. A. Spies, and D. R. Oetter, Modelling forest cover attributes as continuous variables in a regional context with Thematic Mapper data, Int. J. Remote. Sens., 22, 2279-2310 (2001)
9. K. Kim, K. C. Jezek, and H. Liu, Orthorectified image mosaic of Antarctica from 1963 Argon satellite photography: image processing and glaciological applications, Int. J. Remote. Sens., 28, 5357-5373 (2007)
10. R. H. Merson, An AVHRR mosaic image of Antarctic, Int. J. Remote. Sens., 10, 669-674 (1989)
11. K. C. Jezek, Flow variations of the Antarctic Ice Sheet from comparison of modern and historical satellite data, In Geoscience and Remote Sensing Symposium Proceedings, IGARSS, Seattle, WA (New York: IEEE International), 4, 2240-2242 (1998)
12. A. Rosenqvist, M. Shimada, B. Chapman, A. Freeman, G. De Grandi, S. Saatchi and Y. Rauste, The Global Rain Forest Mapping project - A review, Int. J. Remote. Sens., 21, 1375-1387 (2000)
13. L. Shusun, Summer environmental mapping potential of a large-scale ERS-1 SAR mosaic of the state of Alaska, Int. J. Remote. Sens., 20, 387-401 (1999)
14. C. G. Homer, R. D. Ramsey, T. C. Edwards, and A. Falconer, Landscape cover-type modeling using a multi-scene Thematic Mapper mosaic, Photogramm. Eng. Rem. S., 63, 59-67 (1997)
15. M. A. Wulder, J. C. White, S. N. Goward, J. G. Masek, J. R. Irons, M. Herold, W. B. Cohen, T. R. Loveland, C. E. Woodcock, Landsat continuity: Issues and opportunities for land cover monitoring, Remote Sens. Environ., 112, 955-969 (2008)

16. C. H. Matthew, R. L. Thomas, A review of large area monitoring of land cover change using Landsat data, Remote. Sens. Environ., 122, 66-74 (2012)
17. S. V. Muller, A. E. Racoviteanu and D. A. Walker, Landsat MSS-derived land-cover map of northern Alaska: Extrapolation methods and a comparison with photo-interpreted and AVHRR-derived maps, Int. J. Remote. Sens., 20, 2921-2946 (1999)
18. Y. E. Shimabukuro, E. M. Novo, L. K. Merte, Amazon River mainstem flood Landsat TM digital mosaic, Int. J. Remote. Sens., 23, 57-69 (2002)
19. J. G. Masek, E. F. Vermote, N. E. Saleous, R. Wolfe, F. G. Hall, K. F. Huemmrich, F. Gao, J. Kutler, and T. K. Lim, A Landsat surface reflectance dataset for North America, 1990-2000, IEEE T. Geosci. Remote., 3, 68-72 (2006)
20. C. Homer, C. Q. Huang, L. M. Yang, B. Wylie, and M. Coan, Development of a 2001 national land-cover database for the United States, Photogramm. Eng. Rem. S., 70, 829-840 (2004)
21. J. E. Vogelmann, S. M. Howard, L. M. Yang, C. R. Larson, B. K. Wylie, and N. Van Driel, Completion of the 1990s National Land Cover Data set for the conterminous United States from Landsat Thematic Mapper data and ancillary data sources, Photogramm. Eng. Rem. S., 67, 650-662 (2004)
22. P. R. David, J. Junchang, K. Kristi, L. S. Pasquale, K. Valeriy, H. Matthew, R. L. Thomas, V. Eric, Z. Chunsun, Web-enabled Landsat Data (WELD): Landsat ETM+ composited mosaics of the conterminous United States, Remote. Sens. Environ., 114, 35-49 (2010)
23. Z. Li, S. Nadon and J. Cihlar, Satellite-based detection of Canadian boreal forest fires: Development and application of the algorithm, Int. J. Remote. Sens., 21, 3057-3069 (2000)
24. Z. Wei, Z. C. Chen, B. Zhang, CBERS-1 digital images mosaic and mapping of China, Journal of Image and Graphics, 11, 787-791 (2006)
25. A. Wang, Y. Chi, Z. Wang, F. Wu, X. Wang, L. Li, M. Yan, Study on mosaic technology and mapping of China based on the multispectral images of Beijing-1 small satellite, Journal of Remote Sensing, 13, 83-90 (2009)
26. R. Bindschadler, F. Brownworth, S. Stephensonn, Landsat Thematic Mapper imagery of the Siple Coast, Antarctica Antarctic, Journal of the United States, 23, 214-215 (1988)
27. H. Bennat and J. Sievers, Remote sensing and GIS application in Antarctica, EARSel Advances in Remote Sensing, 1, 160-168 (1992)
28. J. G. Ferrigno, J. L. Mullins, J. A. Stapleton, P. S. Chavez, M. G. Velasco, and R. S. Williams, Satellite image map of Antarctica, Miscellaneous Investigations Map Series 1-2560, U. S. Geological Survey (1996)
29. USGS, Satellite Image Map of Antarctica, 1:5, 000, 000, Miscellaneous Map Investigation Series, I-2284 (1991)
30. K. Jezek, Glaciological properties of the Antarctic ice sheet, from Radarsat-1 Synthetic Aperture Radar Imagery, Annals of Glaciology, 29, 286-290 (1999)
31. T. Scambos, T. Haran, M. Fahnestock, T. H. Painter, J. Bohlander, MODIS-based mosaic of Antarctica (MOA) data sets: Continent-wide surface morphology and snow grain size, Remote Sens. Environ., 111, 242-257 (2007)
32. R. Bindschadler, P. Vornberger, A. Fleming, A. Fox, J. Mullins, D. Binnie, S. Paulsen, J. Sara, B. Granneman, D. Gorodetzky, The Landsat image mosaic of Antarctica, Remote Sens. Environ., 112, 4214-4226 (2008)
33. P. J. Turner, A. J. Prata, R. T. Howden, N. R. Houghton and Taylor, An ATSR-2 Mosaic Image of Australia, Int. J. Remote. Sens., 22, 3889-3894 (2001)
34. Y. Rauste, Compilation of Bi-temporal JERS-1 SAR Mosaics over the African Rain Forest Belt in the GRFM Project, Presented at the International Geoscience and Remote Sensing Symposium (IGARSS99), Hamburg, Germany, 1999, 28 June-2 July 1999 (IEEE Publications), 2, 750-752 (1999)
35. L. Khatib, J. Gasch, R. Morris, and S. Covington, Local search for optimal global map generation using mid-decadal Landsat images, Proceedings of AAAI Workshop on Preference Handling for Arti-cial Intelligence, Vancouver, BC (2007)
36. P. Mayaux, F. Achard, and J. Malingreau, Global tropical forest area measurements derived from coarse resolution satellite imagery: A comparison with other approaches., Environmental Conservation, 25, 37-52 (1998)
37. C. J. Tucker, M. G. Denelle, and, D. Dykstm, NASA's Global Orthorectified Landsat Data Set, Photogramm. Eng. Rem. S., 70, 313-322 (2004)

38. C.A. Lee, S.D. Gasster, A. Plaza, et al. Recent Developments in High Performance Computing for Remote Sensing: A Review. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 4(3), 508-527 (2011)
39. J. Dongarra, T. Sterling, H. Simon, et al. High-Performance Computing: Clusters, Constellations, Mpps, and Future Directions. Computing in Science and Engineering, 7(2), 51-59 (2005)
40. B. Hu, H. Zhou, P. Wang, H. Liu, A Parallel Algorithm of PCA Image Fusion in Remote Sensing and Its Implementation, Microelectronics and Compute, 23, 153-155 (2006)
41. A. Plaza, D. Valencia, J. Plaza, and P. Martinez, Commodity cluster based parallel processing of hyperspectral Imagery, Journal of Parallel and Distributed Computing, 66, 345C358 (2006)
42. X. An, X. Wang, Z. Du, D. Liu, G. Li, Fine-grained parallel algorithm for remote sensing image mosaics for cluster system, J. Tsinghua Univ. (Sci. and Tech.), 42, 1389-1392
43. B. Hu, H. Zhou, P. Wang, H. Liu, A parallel algorithm of PCA image fusion in remote sensing and its implementation, microelectronics and Compute, 23, 153-155 (2006)
44. B. Hu, H.Z. Liu, P.F. Wang, et al. Research on parallel image fusion model on pixel scale. Computer Era, 2008, 2: 6-8
45. C. Chen, Y.H. Tan, H.T. Li, H.Y. Gu, A Fast and Automatic Parallel Algorithm of Remote Sensing Image Mosaic. Microelectronics and Computer, 28, 59-62 (2011)
46. C. Chen, High Performance Image Mosaicking for HJ-1 Satellites Images, Huazhong University of Science and Technology, Wuhan, China, 2011
47. Y. Wang, Y. Ma, P. Liu, D. Liu, and, J. Xie, An optimized image mosaic algorithm with parallel i/o and dynamic grouped parallel strategy based on minimal spanning tree, Ninth International Conference on Grid and Cloud Computing, Beijing (2010)
48. Y. Ma, L. Wang, A. Zomaya, D. Chen, R. Ranjan, Task-Tree Based Large-Scale Mosaicking for Massive Remote Sensed Imageries with Dynamic DAG Scheduling, IEEE Transactions on Parallel and Distributed Systems, 25(8), 2126-2137 (2014)
49. Y. Yang, Design and implementation of high performance mosaic system for remote sensing image, M.S. thesis, Huazhong University of Science and Technology, Wuhan,China, 2012
50. A.Merzky,K.Stamou,S.Jha, and D.S. Katz, A Fresh Perspective on Developing and Executing DAG-Based Distributed Applications: A Case-Study of SAGA-Based Montage, in Proc. 5th IEEE Intl Conf. e-Science, 231-238 (2009)
51. G.B. Berriman, J.C. Good, D. Curkendall, J. Jacob, D.S. Katz, T.A. Prince, and R. Williams, Montage: An On-Demand Image Mosaic Service for the NVO, in Proc. ADASS XII, (2002)
52. R.Rabenseifner, G.Hager,and G.Jost, HybridMPI/OpenMP Parallel Programming on Clusters of Multi-Core SMP Nodes, Proc. PDP Netw.-Based, 427-436 (2009)
53. H. Wang, Parallel Algorithms for Image and Video Mosaic Based Applications, M.S. thesis, University of Georgia, Atlanta, GA, USA (2005)
54. J. Nickolls and W. J. Dally, The GPU computing era, IEEE Micro, 30, 56C69 (2010)
55. T. Balz and U. Stilla, Hybrid GPU-based single and double bounce SAR simulation, IEEE Trans. Geosci. Remote Sens., 47, 3519C3529 (2009)
56. C.Z. Qin, L.J. Zhan, Parallelizing flow-accumulation calculations on graphics processing unitsFrom iterative DEM preprocessing algorithm to recursive multiple-flow-direction algorithm, Computers and Geosciences, 43, 7-16 (2012)
57. A. Camargo, R.R. Schultz, Y. Wang, R.A. Fevig, and Q. He, GPU-CPU Implementation for Super-Resolution Mosaicking of Unmanned Aircraft System (UAS) Surveillance Video, in Proc. IEEE SSIAI, 25-28 (2010)
58. K.A.T. Yong , J.T. Wee, K.K. Leong. Fast Colour Balance Adjustment of IKONOS Imagery Using CUDA. Geoscience and Remote Sensing Symposium, 1052-1055 (2008)
59. C.Z. Qin, L.J. Zhan, A.X. Zhu, C.H. Zhou, A strategy for raster-based geocomputation under different parallel computing platforms, International Journal of Geographical Information Science, http://dx.doi.org/10.1080/13658816.2014.911300.
60. P. S. Thenkabail, Characterization of the alternative to slash-and-burn benchmark research area representing the Congolese rainforests of Africa using near-real-time SPOT HRV data, Int. J. Remote. Sens., 20, 839-877 (1999)