The Challenges of Image Segmentation in Big Remotely Sensed Imagery Data

Jin Xing, Renee Sieber, and Margaret Kalacska
Department of Geography, McGill University, Montreal, Canada
Email: jin.xing@mail.mcgill.ca; renee.sieber@mcgill.ca; margaret.kalacska@mcgill.ca

Abstract

With the increase in spatial, spectral and temporal resolutions of earth observing systems, geospatial and remote sensing (RS) image research is shifting towards a big data paradigm. One of the most important challenges in RS big data is image segmentation, which is defined as a process to group pixels together by a pre-defined criteria. Image segmentation allows for the extraction of features such as roads, or habitats or buildings. Image segmentation is rendered more difficult with big data because the computing power on single platforms cannot keep pace with the size and velocity of new data. Big datasets must be decomposed for analysis in distributed and parallel computing platforms. Decomposition through techniques like slicing by spatial extent obscures the geometric and topological information in geospatial data, for example generating fake artifacts. To address these challenges, we propose a geospatial cyberinfrastructure (GCI) that coordinates cloud computing, MapReduce framework, image segmentation algorithms, a spatial extent splitting method and a recomposing technique using moving window. This GCI is evaluated on cloud computing to identify features in a 312.07GB high resolution color aerial photo with Hadoop. K-means based image segmentation is selected as the case study. We deploy the architecture in a private cloud computing and public cloud implementation. The results demonstrate the benefits of the decomposing and recomposing methods in segmenting images, removing fake artifacts, and reducing information distortion.

More general problems in big data are revealed, among them I/O problems, particularly in the amount of pre-processing and post-processing that will be required in any analysis of big imagery data. We conclude with implications for scalability and suggestions to speedup decomposition and recomposition.

Index Terms—Big Data, Image Segmentation, Geospatial CyberInfrastructure, Spatial Feature Extraction, Cloud Computing, MapReduce, Decomposition, Recomposition

1. Introduction

As an important type of raster data often used with GIS (Geographic Information Systems), remote sensing (RS) imagery provides the standard approach to Earth observation and geospatial knowledge (Richards 2013). However, RS technologies are rapidly changing, with increases in the spatial, spectral and temporal resolutions of the imagery. For example, the IKONOS sensor provides 1-m spatial resolution panchromatic images with 3 days revisiting interval (Richards 2013). These enhanced resolutions reveal detailed spatiotemporal information about landscape usage and changes. At the same time, they also result in large volumes of data. This leads to a 'big data' challenge in RS and GIScience research.

The phenomenon of big data does not only pose substantial challenges in data management, but also the corresponding data analysis and the provisioning of computing resource. Because the expanding volume of imagery data exceeds the memory size of most computers, new computing technologies are being investigated as part of GCI (Geospatial CyberInfrastructure) research (Yang *et al.* 2010). These new GCIs can provide parallel computing services for geospatial data analysis with a large body of computing resources, including grid computing (Wang and Liu 2009), CUDA (Compute Unified Device Architecture) (Xia *et al.* 2011) and cloud computing (Yang *et al.* 2011).

We are particularly interested in cloud computing, which has become the standard platform for analyzing big data (Yang *et al.* 2013). Cloud computing is defined as a coordinated remote servers accessible via the Internet. Cloud computing has attracted considerable research interest in GIScience because it provides a very large computing resource with on-demand provisioning; this type of provisioning offers efficiencies in resource allocation (i.e., users purchase hardware time as a service and only as needed); much of the big data already "lives" in the cloud; and numerous server-side tools have been migrated to the cloud. Our GCI coordinates cloud computing with a MapReduce framework to address the challenge of big data in RS research. MapReduce is an approach to manage the distribution of large scale computing tasks (Dean and Ghemawat 2008), which has been studied for geospatial data analysis in RS and GIS (Almeer 2012).

This paper is organized as following. Section 2 introduces the background of big data and related works about image segmentation in RS, and we also present the specific challenges brought by large RS imagery datasets in this section. We propose a four-layered image segmentation GCI in Section 3. We then test this GCI in Section 4 with a high resolution color aerial photo (50cm spatial resolution and 312.07 GB) using a k-means image segmentation algorithm. We choose k-means because it is one of the most popular clustering algorithms and has seen broad application across numerous geospatial domains (Jain 2010). Conclusions and future works are described in Section 5. The contribution of this paper is three-fold: (1) we delineate issues in RS image segmentation specific to big data; (2) we propose GCI that integrates image segmentation algorithms and advanced computing techniques; and (3) we present guidelines for deciding between private/public cloud computing platforms for big RS data analysis.

2. Literature Review

2.1. Spatial Information, Features and Image Segmentation

Spatial information, represented as different spatial features, plays a pivotal role in RS knowledge discovery (Liu and Buhe 2000). Various spatial features in RS images include edge, texture, interest points and shapes. Feature extraction algorithms have been used to extract different spatial objects of interest (Ren and Ma 2010). Among these, image segmentations are among the most widely applied in classification (Mather and Tso 2010), object based image analysis (Blaschke 2010), and change detection studies (Radke *et al.* 2005). Image segmentation is the process of clustering the image into multiple groups of pixels (also called segments) based on similarity criteria (e.g., texture and digital number). We use one of the image segmentation algorithms in this case study (k-means) to illustrate the workflow of our GCI.

Image segmentation algorithms have been studied in RS for decades and have been extended with different computing techniques. For example, Gruia *et al*. (2007) customize Fuzzy c-means clustering algorithm for grid computing to segment MODIS (Moderate Resolution Imaging Spectroradiometer) satellite images. They report speedup and efficiency improvements using grid computing and they point out the importance of joining separate clustering results from each computing node. Due to the small size of their testing data (65MB), their works focuses on the computational intensity of image segmentation.

A number of researchers focus on distributed k-means algorithm for image segmentation. Backer *et al*. (2013) implement parallel k-means image segmentation on a GPU (Graphics Processing Unit). They find the massive parallel processing capacity of GPUs significantly exceeds that of CPUs. They did need to customize the k-means so that it could be parallelized

and their approach assumes all the data is already loaded into GPU memory. Liu and Cheng (2012) present parallel k-means algorithm with cloud computing. They point out that the relation between computational time growth and data volume increasing is not obvious, which further confirms the potential of cloud computing for big RS data. Lv *et al*. (2010) apply the algorithms proposed by Zhao *et al*. (2009) to segment large remote sensing imagery datasets. These works also emphasizes the computational intensity, in this case of k-means image segmentation. Our research begins to shift the focus from, for example, computational intensity to distributing, managing and analyzing big data.

2.2 The Challenge of Big Data in RS Image Segmentation

Handling big data has become increasingly important with the rapid changes in data acquisition approaches, ranging from business transaction records to real-time traffic surveillance datasets (Manyika *et al.* 2011). The quality of data has also been enhanced by new technologies, such as the high resolution satellite sensing systems like Ikonos, QuickBird, WorldView and GeoEye (Richards 2013). It is widely accepted to describe big data by the combination of the "4Vs": volume (large volume size), variety (multiple data types), velocity (data is produced at fast speed) and veracity (accuracy of data becomes more important) (Gupta *et al.* 2012). Researchers already have begun to study big RS data, as evinced by papers on in fields like forestry, land use change, ecology (Hampton *et al*. 2013; Harrison 2013; Michael and Miller 2013).

Big RS imagery datasets offer an excellent example of the 4Vs. The large volume of big RS imagery datasets is caused by two factors: (1) the improvement of spatial and spectral resolutions of sensing instruments, and (2) the emergence of new sensing systems (Richards 2013). The first factor produces images with fine resolution conveying more detailed geospatial

information; whereas, the second factor generates different resolutions and data formats. Therefore, an accurate overview of the big RS imagery is generally unavailable on many existing computing platforms. For us, the variety of big RS imagery not only refers to various image types, but also to the large body of image analysis methods. We are receiving data with far greater frequency, consequently, high velocity can be interpreted as the high temporal resolution in new sensing systems, which enables the study of multi-temporal land use and land cover change detection (Lu *et al.* 2004). The veracity is characterized by the error and noisy information in big RS imagery, including sensing system errors, atmospheric impact, and noisy information introduced by data pre-processing (Lee *et al.* 1990).

Big RS images differ from the other types of big data. As a typical raster data, information is not only contained in the digital values recorded in each band, but also its geospatial position and its position within the file. By contrast, the big data contributed by Twitter (Bughin *et al.* 2010) can be stored in separate files and, with the exception of time, the order of Twitter records has a limited impact on the results of data mining (Ediger *et al.* 2010).

Big RS imagery differs from big Twitter data. Figure 2-1 shows (a) the spectral signature of one sampling point in the parking ground and (c) the spectral signature of one sampling point from the highway. They display quite similar results because both parking ground and highway are cement products. In the spatial context, it is the topological and geometric information of these pixels help us classify “road” and “parking”, not individual pixel values. With big RS imagery, the topological and geometric information becomes more complicated and should be handled carefully, for feature level image analysis.

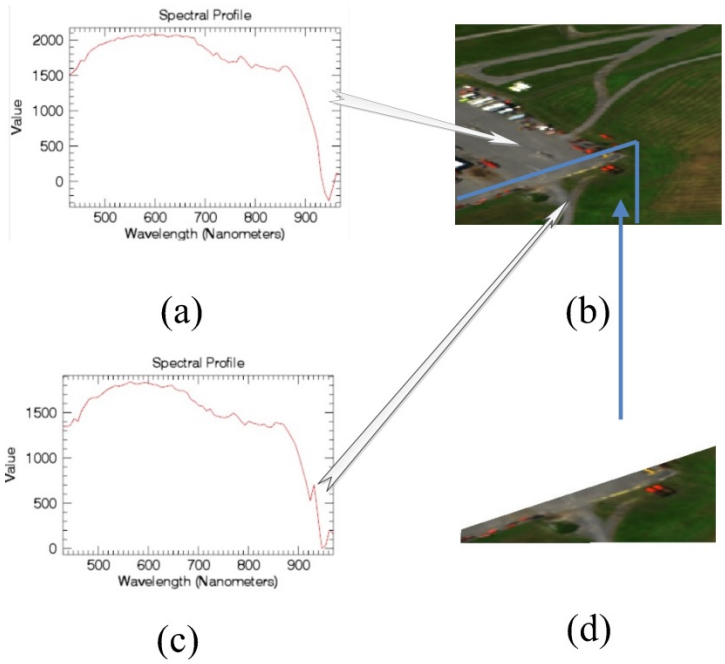

Figure 2-1 (a) Spectral signature of one sampling point on the parking ground; (b) Airborne Visible / Infrared Imaging Spectrometer image; (c) Spectral signature of one sampling point on the highway; (d) A fake "road" generated by image splitting

Because the volume of big data exceeds the memory of most computers, splitting the big data into small chunks or use of sampling methods offers an effective way to handle the data (Cohen *et al.* 2009). While fine for some types of big data (e.g., Twitter), for big RS imagery they may change the original geometric and topological information. Sampling may be highly biased because it breaks the raster cell structure and relationship between pixels is altered.

A significant challenge lies in segmenting images across split image chunks. It is akin to "can't see the forest for the trees." Figure 2.2 shows the separate image segmentation process with image chunks; in which features are extracted locally with significant global information lose. Topological and geometric information in the original big RS image is inevitably altered in

the splitting processing as data is distributed over numerous computing nodes. These challenges will become more important as data grows in size, speed, and variety.

One of the outcomes of splitting the image is the creation of artificial borders. These are borders that do not exist in real life. For example, artificial borders might "cut" a narrow strip from the parking ground in Figure 2-1 (b), and label it as "road". As we show in Figure 2-1 (d), a fake "road" is created by the splitting process. Compared with other types of big data, big RS imagery needs to be processed with the goal of preserving as much original geometric and topological information as possible.

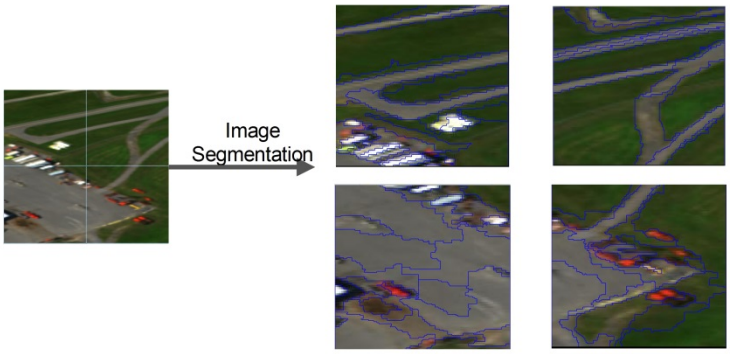

Figure 2-2 Splitting Figure 2-1 (b) into 2×2 chunks and segmenting each chunk, the image segmentation is generated by eCognition®, with scale=50 and color=0.5

In Figure 2-2, most of the highways are segmented into several independent features due to the artificial borders introduced by image splitting. The artificial borders change the geometric information of highways and bring additional errors into the following image analysis process

(e.g., classification). In Figure 2-2, the artificial borders lead to nine additional road segments, because local processing with each image chunk cannot distinguish between the real and artificial borders. We name this type of challenge as artificial border challenge in big RS imagery. In this paper, we remove these false segments caused by the artificial border challenge using a decomposition/recomposition based workflow management framework. The details of artificial border challenge are summarized in Table 2-1. These are a collection of five challenges in which image splitting causes fake features in image segmentation (Figure 2-2). All these challenges grow with big data and will likely see greater attention in GIS and RS research.

Table 2-1 Artificial Border Challenge

Challenge Name	Explanation	Example
Edge Ambiguity	Some edges or features are treated as the image border by mistake	A line of fence at the image chunk border disappears in image segmentation process because it is treated as image border
Feature Bisection	Dividing one feature into two or more features	Cutting a road into two road segments
Fake Feature Creation	Create two or more features from original feature	Parallel cutting of one road into two distinct road segments
Feature Transformation	Change the type of the original feature	Segmenting parts of the parking lot into road segment and smaller parking lot (Figure 2-2)
Feature Distortion	Change the properties of the original feature	Generating a parking lot smaller than its actual size in original RS image

2.3 Addressing Big Data through GCIs

Although there is very little work about using GCI for RS image analysis, GCI has already been proven as an effective solution in big data processing (Wright and Wang 2011). Research in GCIs spans numerous topics. These include the transformation of GCI from a technology-centered to a human-centered paradigm (Díaz *et al. 2011*), workflow optimization in geospatial data analysis (Zhang and Tsou 2009), semantic web with semantic knowledge system (Sieber *et al*. 2011), interfaces for public sciences (Ramamurthy 2006), and interactions among GCIs (Yang and Raskin 2009). Several researchers are adapting GCIs for specific research problems (Yang *et al*. 2011). For example, Liang *et al.* (2010) build a GCI based on social networks and hybrid P2P (Peer-to-Peer) techniques to enable sharing and visualization of big environmental sensing datasets. Díaz *et al.* (2011) present a GCI architecture for large user generated information management and semi-automatic web service built-up using these big data. The emergent computing technologies in current GCI research have been summarized by Yang *et al.* (2010), among which cloud computing and MapReduce are highlighted for managing the exponentially growing geospatial datasets.

Rajasekar *et al.* (2010) highlights the need to utilize GCI in RS research to manage the increasing data volume, and Xue and Diao (2010) confirm the pivotal role GCI plays in analyzing big RS datasets. However, GCIs have not been studied systematically for big RS image segmentation. Big RS imagery datasets requires scalable data management, as the response to the volume and velocity. Like vector-based GCIs, a raster based GCI needs to geometry and topology. Concerning issues in variety, a single image segmentation algorithm may be insufficient to cover different types of data. Therefore, a broad range of image segmentation algorithms should be implementable. Wherever possible, new flexible computing techniques should be utilized.

One flexible technique is utilization of the cloud for GCIs, which already have improved performance in handling big geospatial data. For example, the Google App Engine (Zahariev 2009) is utilized to index and retrieve large spatial image data online (Wang *et al.* 2009). Li *et al.* (2010) build a new GCI based on the Microsoft Azure platform to retrieve and re-project MODIS satellite data. Their cloud computing implementation is able to generate a 90 times speedup over a single desktop implementation. The cloud computing special issue of the *International Journal of Digital Earth* (2013) further reveals the strength of cloud computing in processing big geospatial data and summarizes the wide application of cloud computing based GCI in geospatial research (Yang *et al.* 2013).

MapReduce also has been shown to be valuable for image analysis. Generally, there are two phases in MapReduce: the *map* phase and the *reduce* phase. The *map* phase splits the original datasets into a number of *key/value* pairs and executes data analysis algorithms with the generated *key/value* pairs. The *reduce* phase takes the output from the *map* phase and combines them to form the final results. MapReduce monitors the execution of all tasks; failed tasks are automatically rescheduled on other computing nodes (Dean and Ghemawat 2008). Golpayegani and Halem (2009) test MapReduce with AIRS (Atmospheric Infrared Sounder) images for gridding problem solving, which showed MapReduce is efficient in processing large spaceborne RS images. Zhao *et al.* (2009) develop parallel k-means algorithm with MapReduce. Previous k-means could run only on one computer; they extend it so the analysis could be distributed alongside the data. Lv *et al.* (2010) apply the algorithms proposed by Zhao *et al.* (2009) to segment large RS imagery datasets. This further emphasizes the important role MapReduce plays in RS research. However, these authors have not explored all the implications of MapReduce (e.g., the creation of artificial borders when data is distributed) and they did not explicate

computing resource provisioning needs for big data (e.g., the leasing cost of the virtual machines, input/output issues in moving large data sets). We distinguish between image segmentation to find features and image splitting to divide the image into manageable chunks, although there are interesting similarities between the two.

3. Using GCI as A Solution for Image Segmentation in Big RS Imagery Data

3.1 Architecture of the Image Segmentation GCI

We propose a GCI that combines cloud computing, MapReduce parallel computing framework and RS image segmentation algorithms as a holistic solution for the challenges posed by big RS image data.

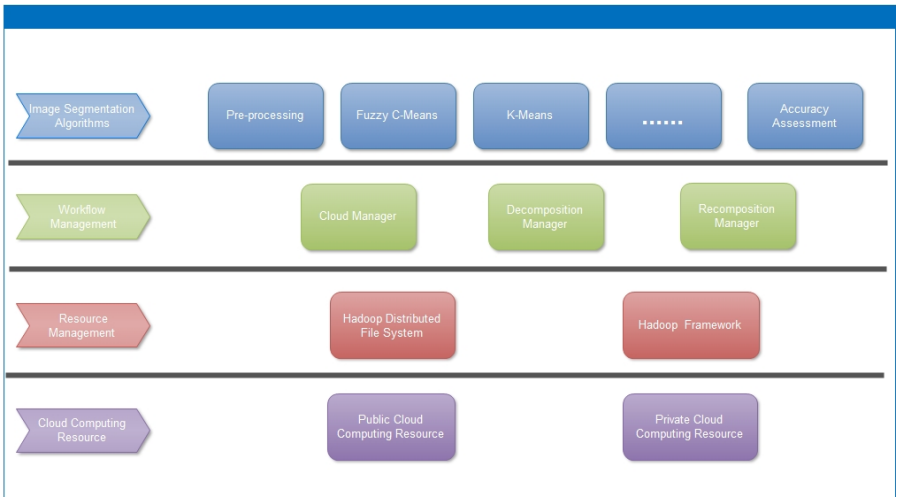

Figure 3-1 GCI Architecture

The architecture of our GCI is shown in Figure 3-1, which is composed of four layers (from bottom to top): cloud computing resource layer, resource management layer, workflow management layer and the image segmentation process layer. The computing resource interface is designed to utilize both computing resources from private and public cloud computing providers, which also can be used to build a hybrid public/private cloud. The resource management layer is developed with Hadoop, which is an open source implementation of

MapReduce (Borthakur 2007). This layer also includes HDFS (Hadoop Distributed File System), which is a scalable distributed storage system compatible with Hadoop computing framework. The workflow management layer is built on top of Eucalyptus open-source cloud computing manager, containing the decomposition and recomposition manager. Since image splitting plays such a large role in our image segmentation, the functionalities of the workflow management layer will be discussed in greater detail in Section 3.2. Finally, different RS image segmentation algorithms, corresponding pre-processing methods, and the accuracy assessment functions compose the image segmentation process layer. This layer will be discussed further in Section 3.3.

3.2 The Workflow Management Layer

The general workflow of segmenting big RS imagery is depicted in Figure 3-2, which consists of decomposition and recomposition steps. The decomposition manages the following functions:

1) Split the big RS imagery into image chunks with spatial extent decomposition method;
2) Schedule image segmentation algorithm in multiple parallel *map* tasks in Hadoop with each image chunk. The generated image segments overlays are cached in the local storage of each computing node, which will be fetched by the *reduce* task.

   The recomposition manager provides functionalities to:
3) Collect the image segments from all *map* tasks;
4) Execute our window based fake segment removing algorithm;
5) Merge all the chunks to generate the holistic results.

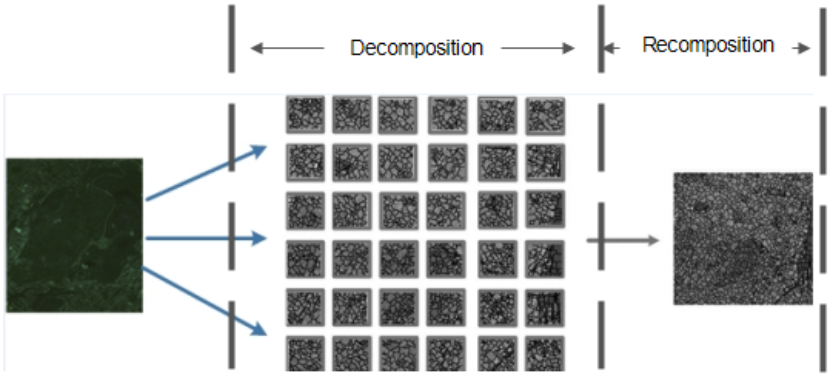

Figure 3-2 Overview of the Decomposition/Recomposition Workflow Management Framework

A detailed description of these steps are given is Figure 3-3. For each big RS imagery, only one *reduce* task is scheduled in Hadoop, which is granted the global view because the fake segments removal needs to access the global information.

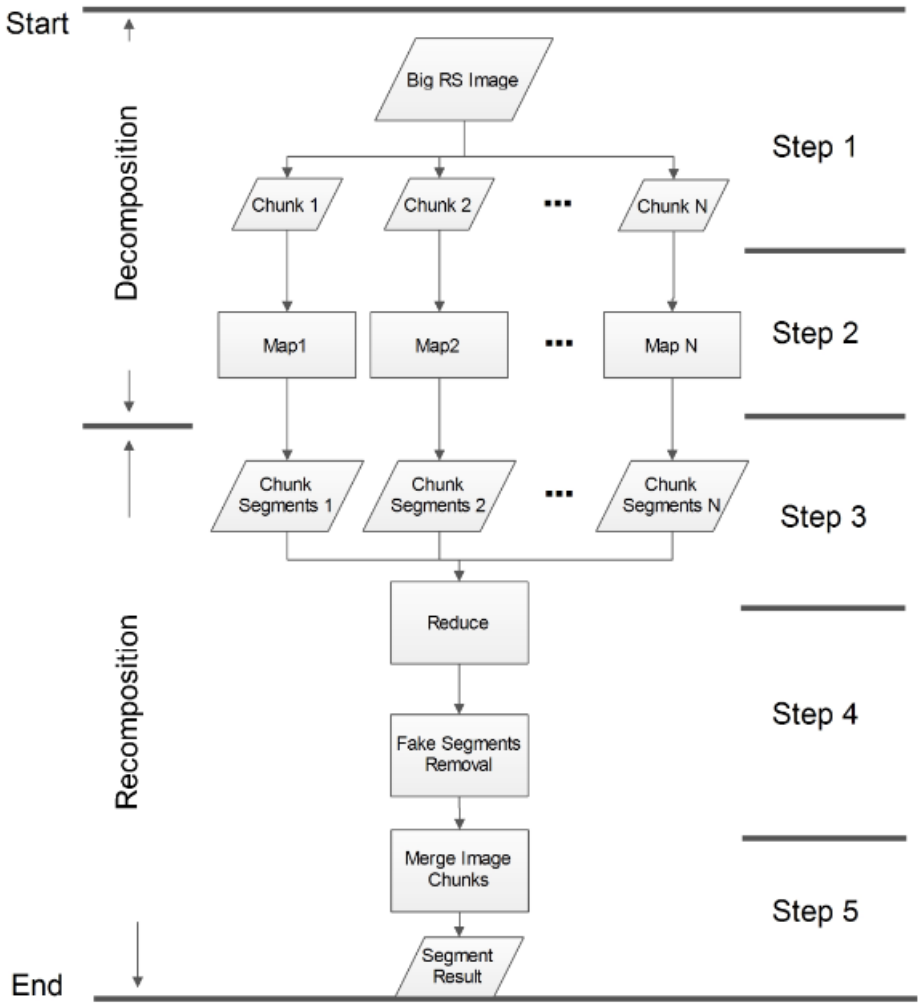

Figure 3-3 Steps of Decomposition/Recomposition with MapReduce

3.2.1 Spatial Extent Image Splitting Method

The splitting of big RS imagery plays a pivotal role in decomposition process. On one hand, the splitting process should generate chunks of small spatial extent because small size can be better handled by *map* tasks (Dean and Ghemawat 2008). However, a, smaller chunk size means a larger number of chunks, which impacts analysis. Liu *et al.* (2012) propose a pyramid partitioning algorithm to split the big RS imagery into small chunks with different levels of resolutions for MapReduce processing. However, big RS imagery, which cannot be loaded into the memory of computers, prevents the generation of the pyramid hierarchy. And this method

does not account for the memory sharing problem. Several *map* tasks might be scheduled on one computing node so frequent swapping operation caused by large image chunks will significantly deteriorate the computing performance.

We propose a two-tiered spatial extent image splitting method layered onto a areal-based splitting method that generates image chunks with equal size. The areal-based splitting divides a big RS imagery into equal-area sub-rectangles (or squares) according to the abscissa and ordinate values (Maulik and Sarkar 2012). The spatial extent image splitting method calculates the size of each image chunk as the lower bound of the GCD (greatest common divisor) of the average memory allocated to each *map* task and the data size allocated to each *map* task, as:

$$chunk\_size = \left\lfloor GCD\left(\frac{S \times k}{m}, \frac{N}{m}\right) \right\rfloor \quad (3.2.1\text{-}1)$$

N represents the total number of pixels in the big RS imagery; m is the number of *map* tasks; S is the memory size of each computing node; and k represents the number of computing nodes. We assume all the slave nodes have the same computing resource and image chunks are split equally (chunk size may vary at the border of big RS imagery). $\frac{N}{m}$ is the largest chunk size that balances the load, whereas $\frac{S \times k}{m}$ is the largest chunk size can be processed by each *map* task at the same time. This spatial extent splitting method ensures the load balancing and computing performance of each *map* task.

3.2.2 Moving Window based Fake Segment Removal

We utilize the moving window (Papps 1992) based clustering method to remove the fake segments generated by the artificial splitting border. Some segments will be joined to reduce the overall number of segments; other segments will be removed because they reflect edges of image chunks (see red lines in Figure 3-3). When the resulting segments are collected from the *map* tasks, segments that were extracted at the domain borders of each image chunk are marked. The

size of the moving window is set to the same value as the image chunk. The image segmentation algorithm (called image clustering in Pham 2001) is employed with the 8 neighbour chunks, as shown in Figure 3-3. This clustering process does not create any new segments, but tests whether the segments at the border of the image chunk can be merged with the neighbouring segments. Our test is comprised of using K-means algorithm a second time to identify new edge segments. The original segments are overlaid and subtracted. If pieces of segments remain then we know to combine the segments from adjoining chunks. This process continues until all the image chunks have been checked. In this way, the artificial border challenge is resolved.

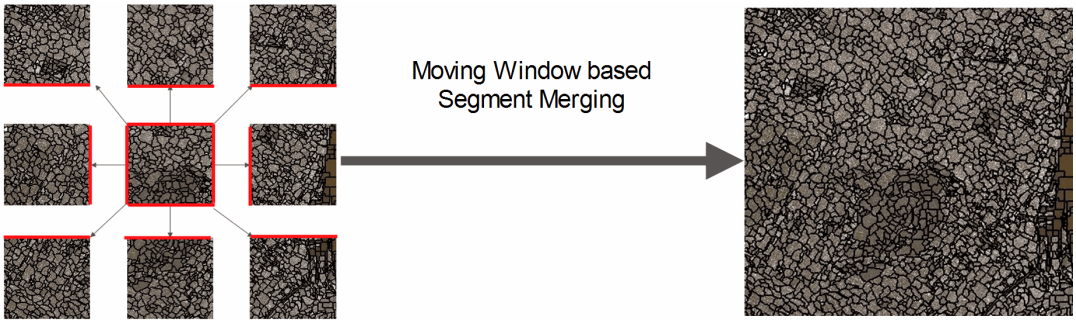

Figure 3-3 Moving Window based Segment Merging Process

Algorithm: Moving Window based Segments Merging

Input: image chunk array C, and corresponding segments overlay S
Output: new image segments overlay S'

```
for each image chunk ci in C:
    load(corresponding si);
    mark all the segment on the border of si and store them as B;
    N=load(neighbors of ci);
    A=merge(ci,N);
    B'=cluster(A);
    for each border segment bi in B:
        load(corresponding bi' from B');
        difference=compare(bi',bi);
        if (difference > threshold) then
            si=si - bi;
            bi=merge(bi ,bi');
            si=si + bi;
    end for
    i=i+1;
    load(ci);
end for
S'=merge(s1'…sn');
end
```

3.3 Image Segmentation Layer

In our GCI, the image segmentation layer provides various algorithms for data handling and image segmentation, including the pre-processing methods (Meinel and Neubert 2004), accuracy assessment approaches (Möller and Lymburner 2007), as well as different image segmentation algorithms (e.g, fuzzy c-means, k-means, and region-growing method). The appropriate algorithms can be automatically deployed to the separate computing nodes, as "moving code to the data" mechanism of Hadoop.

After the image is split and distributed, standard RS pre-processing algorithms conducts atmospheric and radiometric correction. Then the image segmentation algorithms are executed. All pre-processing and image segmentation is done on the individual *map* computing nodes. When the image segmentation process is complete on the individual nodes, the workflow layer resumes control with the reduce phase. Control is returned to the image segmentation layer if an accuracy assessment (e.g., calibration) is required.

4. Evaluation of the GCI for Image Segmentation of Big RS Imagery

4.1 Image Segmentation in Two Deployments

We utilize the GCI as an approach to handle big RS imagery and to conduct image segmentation. To evaluate the architecture, we used a 312.07GB RGB aerial photo mosaic (60 cm, taken at Costa Rica 2004). The image segmentation algorithm we choose is k-means based image segmentation (Ray *et al.* 1999), due to its popularity and robust computational complexity. The splitting method is our spatial extent splitting methods in Section 3.2.1, and artificial borders and corresponding fake segments are removed with our moving window based approach in Section 3.2.2. Although we choose k-means image segmentation algorithm, other types of image segmentation can be deployed as well.

We evaluated our GCI with two different deployments, using private and public cloud. We chose two deployments as it reflects the realities of modern implementations, such as resource restraints of researchers (e.g., cost of hardware and software). To eliminate the difference between public and private cloud, we setup the VMs with the same configuration, using Eucalyptus and Amazon EC2. We choose Eucalyptus to build the private cloud because it provides the same interface as Amazon EC2. In this way, we can create virtual machine (VM) instances with negligible difference between the private and public cloud within our GCI. Hadoop 1.0.0 version is selected as the implementation of MapReduce, which is installed on VMs with CentOS 6.4 as the operating system. The detailed information about our testbed is listed in Table 4-1. The physical computing resource refers to the hardware configuration, while the virtual resource is the configuration of VMs (the information of physical machines from Amazon EC2 at running time cannot be obtained).

In these two different deployments, 10 *map* VM and 1 *reduce* VM are utilized respectively. After the testing image is uploaded to HDPS, approximate 500MB image chunks are created by our spatial extent splitting method. Then k-means image segmentation is conducted in *map* VMs and moving window based segment merging algorithm is schedule in the single *reduce* VM. The computation time and cost of the two deployments are delineated in Table 4-2 and 4-3, respectively.

Table 4-1 Details of the Two Testbeds

	Private Cloud	Public Cloud
Physical CPU	Four Intel® six-core XEON E5-2620 2.0 GH	N/A
Virtual CPU	One for *map* VM and four for *reduce* VM (One VCPU= 2.0 GHz 2007 Xeon processor)	One for *map* VM and four for *reduce* VM (One VCPU= 2.0 GHz 2007 Xeon processor)
Physical Memory	64 GB	N/A
Virtual Memory	3.75 GB for *map* VM and 15 GB for *reduce* VM	3.75 GB for *map* VM and 15 GB for *reduce* VM
Physical Network	1 Gbpbs	N/A
Virtual Network	1 Gbpbs for all VMs	Medium for *map* VM and high for *reduce* VM
Physical Storage	4 TB	N/A
Virtual Storage	410 GB for *map* VM and 80GB for *reduce* VM	410 GB for *map* VM and 80GB for *reduce* VM
OS	CentOS 6.4	CentOS 6.4
VMs	10 *map* and 1 *reduce* VMs	10 *map* and 1 *reduce* VMs

Table 4-2 Cost of Image Segmentation Test in Amazon EC2

	Time (Hours)	Cost (Dollars)
Data Uploading	~82.5	~$28.5
Decomposition Computing	~68.4	~$8.21*10
Recomposition Computing	~98.7	~$44.42
Result Downloading	~33.3	~$3
Total	~282.9	~$158.02

Table 4-3 Cost of Image Segmentation Test in Eucalyptus Cloud

	Time (Hours)	Cost (Dollars)
Data Uploading	N/A	N/A
Decomposition Computing	~61.27	N/A
Recomposition Computing	~90.4	N/A
Result Downloading	N/A	N/A
Total	~151.67	N/A

By comparing the segment results before and after the recomposition process in Amazon EC2, we find 487 fewer segments. We also note that data transfer has taken approximately 41 percent of the computation time and 20 percent of the total costs with public cloud computing.

In the Eucalyptus private cloud, there is no data upload and download, but only decomposition and recomposition processes. The decomposition requires 40 percent of the total computation time, while the recomposition requires the rest. There are operating costs with the testing with Amazon EC2; the hardware costs are front in the private cloud. We also compare the segmentation results before and after the recomposition process and find 379 segments have been removed.

4.2 Discussion

Big RS image segmentation is evaluated with two different deployments, a private and a public cloud. Image segmentation appeared to be successful. We observed segments extracted and some of these segments were removed or combined. A windshield would be needed to fully test the efficacy of our method, but it appears that the segments induced by splitting the images, were rejoined effectively. That is, they were identified in the *map* VMs and then combined in *reduce* VM, via the moving window. However, the bottleneck in the recomposition process cannot be neglected, which needs to be parallelized in future research.

From Table 4-2 and 4-3, it appears that the private cloud offers a better choice for image segmentation in big RS imagery datasets due to lessened computation time and costs. However, the cost of purchasing the hardware, setting up the private cloud, and maintaining the cloud environment cannot be neglected. Moreover, the hardware resource in our private cloud testbed is quite small compared with the public cloud, which limits scalability as images grow larger. We cannot use additional *map* tasks in our experiment because the hardware computing resource of the private cloud is insufficient. Considering all the hidden costs, using public cloud for big RS imagery processing appears to be more economical for short-term projects.

Not surprisingly, big data in RS resulted in high I/O costs, regardless of the deployment. Research scientists may choose private cloud to avoid part of these costs, but moving big data

across *map* and *reduce* VMs is quite expensive. Authors agree that research in cloud computing should be devoted to a full accounting of big data I/O costs (Khajeh-Hosseini *et al*. 2012; Kondo *et al.* 2009).

It should be noted that a public cloud computing also involves security and privacy issues (Yu *et al.* 2010). Because different applications and services share the same computing resource pool in public cloud computing, we cannot easily guard against information leakages or surveillance. Considering the current development of cloud computing, a private cloud may be preferred for big RS image segmentation.

5. Conclusion

In this paper, we have discussed the specific characteristics of big RS imagery dataset, and pointed out challenges of image segmentation processing with the big data. A new GCI that coordinates cloud computing, MapReduce, and image segmentation algorithms is proposed, with decomposition/recomposition workflow management framework. The decomposition process splits the big RS imagery into small image chunks and processes them with image segmentation algorithms in parallel as the *map* phase in MapReduce. The recomposition process collects extracted segments from each *map* task, and utilizes a moving window based segment merging method to remove the fake features generated by artificial borders, as the *reduce* phase. We evaluate the performance of our proposed GCI with both public cloud computing and private cloud computing implementation, which shows promising results.

The largest bottleneck in the image segmentation at this time appears to be in the testing of removing artefacts created by artificial borders as mentioned in Table 2-1. Here we might employ a sampling technique to examine the exact nature of the segments that are left over from the moving window. The bottleneck of our GCI mainly lies in two aspects: the first one is that

reduce cannot be scheduled before the finish of all the *map* tasks; the second lies in the nonparallel execution of recomposition process. In the future, we will investigate how to extend recomposition as hierarchical recomposition process for parallelization. The workflow of MapReduce may further be optimized for big RS imagery datasets processing. Intensive I/O operation in our GCI should also be taken into account. We plan to explore parallel I/O framework and the compression method (Lee *et al.* 2012) to improve the performance of our GCI for image segmentation in big RS imagery datasets. In conclusion, using GCI to integrate cloud computing and MapReduce presents great opportunity for big RS imagery analysis.

Acknowledgement

Reference

Almeer, M.H., 2012. Cloud Hadoop Map Reduce For Remote Sensing Image Analysis. Journal of Emerging Trends in Computing and Information Sciences 3, 637–644.

Amazon, E.C., 2010. Amazon elastic compute cloud (Amazon EC2). Amazon Elastic Compute Cloud (Amazon EC2).

Arbelaez, P., Maire, M., Fowlkes, C., Malik, J., 2011. Contour detection and hierarchical image segmentation. Pattern Analysis and Machine Intelligence, IEEE Transactions on 33, 898–916.

Armbrust, M., Fox, A., Griffith, R., Joseph, A.D., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I., 2010. A view of cloud computing. Communications of the ACM 53, 50–58.

Backer, M., Tünnermann, J., Mertsching, B., 2013. Parallel k-means image segmentation using sort, scan and connected components on a GPU, in: Facing the Multicore-Challenge III. Springer, pp. 108–120.

Bhat, M.A., Shah, R.M., Ahmad, B., 2011. Cloud Computing: A solution to Geographical Information Systems (GIS). International Journal on Computer Science & Engineering 3.

Blaschke, T., 2010. Object based image analysis for remote sensing. ISPRS journal of photogrammetry and remote sensing 65, 2–16.

Borthakur, D., 2007. The hadoop distributed file system: Architecture and design.

Bughin, J., Chui, M., Manyika, J., 2010. Clouds, big data, and smart assets: Ten tech-enabled business trends to watch. McKinsey Quarterly 56, 75–86.

Cohen, J., Dolan, B., Dunlap, M., Hellerstein, J.M., Welton, C., 2009. MAD skills: new analysis practices for big data. Proceedings of the VLDB Endowment 2, 1481–1492.

Congalton, R.G., 1991. A review of assessing the accuracy of classifications of remotely sensed data. Remote sensing of Environment 37, 35–46.

Dean, J., Ghemawat, S., 2008. MapReduce: simplified data processing on large clusters. Communications of the ACM 51, 107–113.

Díaz, L., Granell, C., Gould, M., Huerta, J., 2011. Managing user-generated information in geospatial cyberinfrastructures. Future Generation Computer Systems 27, 304–314.

Du, Q., Fowler, J.E., 2007. Hyperspectral image compression using JPEG2000 and principal component analysis. Geoscience and Remote Sensing Letters, IEEE 4, 201–205.

Ediger, D., Jiang, K., Riedy, J., Bader, D.A., Corley, C., Farber, R., Reynolds, W.N., 2010. Massive social network analysis: Mining twitter for social good, in: Parallel Processing (ICPP), 2010 39th International Conference on. pp. 583–593.

Golpayegani, N., Halem, M., 2009. Cloud computing for satellite data processing on high end compute clusters, in: CLOUD 2009 - 2009 IEEE International Conference on Cloud Computing. pp. 88–92.

Gupta, R., Gupta, H., Mohania, M., 2012. Cloud Computing and Big Data Analytics: What Is New from Databases Perspective?, in: Big Data Analytics. Springer, pp. 42–61.

Hampton, S.E., Strasser, C.A., Tewksbury, J.J., Gram, W.K., Budden, A.E., Batcheller, A.L., Duke, C.S., Porter, J.H., 2013. Big data and the future of ecology. Frontiers in Ecology and the Environment 11, 156–162.

Harrison, C., 2013. How Far Can “Big Data” Take Us Towards Understanding Cities? September 19th–21st, 2013 Santa Fe Institute. Department of Civil and Environmental Engineering, Massachusetts Institute of Technology.

Hey, A.J., Tansley, S., Tolle, K.M., 2009. The fourth paradigm: data-intensive scientific discovery.

Jain, A.K., 2010. Data clustering: 50 years beyond K-means. Pattern Recognition Letters 31, 651–666.

Khajeh-Hosseini, A., Greenwood, D., Smith, J.W., Sommerville, I., 2012. The Cloud Adoption Toolkit: Supporting cloud adoption decisions in the enterprise. Software - Practice and Experience 42, 447–465.

Kondo, D., Javadi, B., Malecot, P., Cappello, F., Anderson, D.P., 2009. Cost-benefit analysis of cloud computing versus desktop grids, in: Parallel & Distributed Processing, 2009. IPDPS 2009. IEEE International Symposium on. IEEE, pp. 1–12.

Lee, J.B., Woodyatt, A.S., Berman, M., 1990. Enhancement of high spectral resolution remote-sensing data by a noise-adjusted principal components transform. Geoscience and Remote Sensing, IEEE Transactions on 28, 295–304.

Lee, K.-H., Lee, Y.-J., Choi, H., Chung, Y.D., Moon, B., 2012. Parallel data processing with MapReduce: a survey. ACM SIGMOD Record 40, 11–20.

Li, J., Humphrey, M., Agarwal, D., Jackson, K., van Ingen, C., Ryu, Y., 2010. escience in the cloud: A modis satellite data reprojection and reduction pipeline in the windows azure platform, in: Parallel & Distributed Processing (IPDPS), 2010 IEEE International Symposium on. pp. 1–10.

Liang, S., Chen, S., Huang, C., Li, R., Chang, Y., Badger, J., Rezel, R., 2010. GeoCENS: geospatial cyberinfrastructure for environmental sensing, in: Proceedings of GIScience 2010—Sixth International Conference on Geographic Information Science.

Lin, F.-C., Chung, L.-K., Wang, C.-J., Ku, W.-Y., Chou, T.-Y., 2013. Storage and processing of massive remote sensing images using a novel cloud computing platform. GIScience & Remote Sensing 50, 322–336.

Liu, J.Y., Buhe, A., 2000. Study on spatial-temporal feature of modern land use change in China: Using remote sensing techniques. Quaternary Sciences 20, 229–239.

Liu, S., Cheng, Y., 2012. Research on k-means algorithm based on cloud computing, in: Proceedings - 2012 International Conference on Computer Science and Service System, CSSS 2012. pp. 1762–1765.

Liu, Y., Chen, L., Xiong, W., Liu, L., Yang, D., 2012. A mapreduce approach for processing large-scale remote sensing images, in: 2012 20th International Conference on Geoinformatics (GEOINFORMATICS). Presented at the 2012 20th International Conference on Geoinformatics (GEOINFORMATICS), pp. 1–7.

Lu, D., Mausel, P., Brondizio, E., Moran, E., 2004. Change detection techniques. International journal of remote sensing 25, 2365–2401.

Lucas-Simarro, J.L., Moreno-Vozmediano, R., Montero, R.S., Llorente, I.M., 2013. Scheduling strategies for optimal service deployment across multiple clouds. Future Generation Computer Systems 29, 1431–1441.

Lv, Z., Hu, Y., Zhong, H., Wu, J., Li, B., Zhao, H., 2010. Parallel K-means clustering of remote sensing images based on mapreduce, in: Web Information Systems and Mining. Springer, pp. 162–170.

Lynch, C., 2008. Big data: How do your data grow? Nature 455, 28–29.

Manyika, J., Chui, M., Brown, B., Bughin, J., Dobbs, R., Roxburgh, C., Byers, A.H., 2011. Big data: The next frontier for innovation, competition, and productivity.

Mather, P., Tso, B., 2010. Classification methods for remotely sensed data. CRC press.

Maulik, U., Sarkar, A., 2012. Efficient parallel algorithm for pixel classification in remote sensing imagery. Geoinformatica 16, 391–407.

Meinel, G., Neubert, M., 2004. A comparison of segmentation programs for high resolution remote sensing data. International Archives of Photogrammetry and Remote Sensing 35, 1097–1105.

Michael, K., Miller, K.W., 2013. Big data: New opportunities and new challenges [guest editors' introduction]. Computer 46, 22–24.

Möller, M., Lymburner, L., Volk, M., 2007. The comparison index: A tool for assessing the accuracy of image segmentation. International Journal of Applied Earth Observation and Geoinformation 9, 311–321.

Nurmi, D., Wolski, R., Grzegorczyk, C., Obertelli, G., Soman, S., Youseff, L., Zagorodnov, D., 2009. The eucalyptus open-source cloud-computing system, in: Cluster Computing and the Grid, 2009. CCGRID'09. 9th IEEE/ACM International Symposium on. pp. 124–131.

Pappas, T.N., 1992. An adaptive clustering algorithm for image segmentation. Signal Processing, IEEE Transactions on 40, 901–914.

Pham, D.L., 2001. Spatial models for fuzzy clustering. Computer vision and image understanding 84, 285–297.

Pohl, C., Van Genderen, J.L., 1998. Review article multisensor image fusion in remote sensing: concepts, methods and applications. International journal of remote sensing 19, 823–854.

Radke, R.J., Andra, S., Al-Kofahi, O., Roysam, B., 2005. Image change detection algorithms: a systematic survey. Image Processing, IEEE Transactions on 14, 294–307.

Rajasekar, A., Moore, R.W., Wan, M., Schroeder, W., 2010. Cyber infrastructure for Community Remote Sensing, in: International Geoscience and Remote Sensing Symposium (IGARSS). pp. 1891–1894.

Ramamurthy, M.K., 2006. A new generation of cyberinfrastructure and data services for earth system science education and research. Advances in Geosciences 8.

Ray, S., Turi, R.H., 1999. Determination of number of clusters in k-means clustering and application in colour image segmentation, in: Proceedings of the 4th International Conference on Advances in Pattern Recognition and Digital Techniques. pp. 137–143.

Ren, D., Ma, Y., 2010. Research on feature extraction from remote sensing image, in: ICCASM 2010 - 2010 International Conference on Computer Application and System Modeling, Proceedings. pp. V144–V148.

Richards, J.A., 2013. Remote sensing digital image analysis: an introduction. Springer, 4-20.

Sieber, R.E., Wellen, C.C., Jin, Y., 2011. Spatial cyberinfrastructures, ontologies, and the humanities. Proceedings of the National Academy of Sciences 108, 5504–5509.

Subashini, S., Kavitha, V., 2011. A survey on security issues in service delivery models of cloud computing. Journal of Network and Computer Applications 34, 1–11.

Vaquero, L.M., Rodero-Merino, L., Caceres, J., Lindner, M., 2008. A break in the clouds: towards a cloud definition. ACM SIGCOMM Computer Communication Review 39, 50–55.

Wang, S., Anselin, L., Bhaduri, B., Crosby, C., Goodchild, M.F., Liu, Y., Nyerges, T.L., 2013. CyberGIS software: A synthetic review and integration roadmap. International Journal of Geographical Information Science 27, 2122–2145.

Wang, S., Liu, Y., 2009. TeraGrid GIScience gateway: bridging cyberinfrastructure and GIScience. International Journal of Geographical Information Science 23, 631–656.

Wang, Y., Wang, S., Zhou, D., 2009. Retrieving and Indexing Spatial Data in the Cloud Computing Environment, in: Jaatun, M.G., Zhao, G., Rong, C. (Eds.), Cloud Computing, Lecture Notes in Computer Science. Springer Berlin Heidelberg, pp. 322–331.

Wright, D.J., Wang, S., 2011. The emergence of spatial cyberinfrastructure. Proceedings of the National Academy of Sciences 108, 5488–5491.

Xia, Y., Kuang, L., Li, X., 2011. Accelerating geospatial analysis on GPUs using CUDA. Journal of Zhejiang University SCIENCE C 12, 990–999.

Xue, T., Diao, M., 2010. Geospatial data cyber-infrastructure based on geology metadata standard and web service, in: CAR 2010 - 2010 2nd International Asia Conference on Informatics in Control, Automation and Robotics. pp. 239–241.

Yang, C., Xu, Y., Nebert, D., 2013. Redefining the possibility of digital Earth and geosciences with spatial cloud computing. International Journal of Digital Earth 6, 297–312.

Yang, C., Goodchild, M., Huang, Q., Nebert, D., Raskin, R., Xu, Y., Bambacus, M., Fay, D., 2011. Spatial cloud computing: how can the geospatial sciences use and help shape cloud computing? International Journal of Digital Earth 4, 305–329.

Yang, C., Raskin, R., Goodchild, M., Gahegan, M., 2010. Geospatial cyberinfrastructure: past, present and future. Computers, Environment and Urban Systems 34, 264–277.

Yang, C., Raskin, R., 2009. Introduction to distributed geographic information processing research. International Journal of Geographical Information Science 23, 553–560.

Yu, S., Wang, C., Ren, K., Lou, W., 2010. Achieving secure, scalable, and fine-grained data access control in cloud computing, in: INFOCOM, 2010 Proceedings IEEE. pp. 1–9.

Zahariev, A., 2009. Google app engine. Helsinki University of Technology.

Zhang, L. -P., Liu, G. -L., Jiang, T., 2005. Feature extraction and classification of hyperspectral remote sensing image oriented to easy mixed-classified objects. Transactions of Nonferrous Metals Society of China (English Edition) 15, 160–163.

Zhang, T., Tsou, M.-H., 2009. Developing a grid-enabled spatial Web portal for Internet GIServices and geospatial cyberinfrastructure. International Journal of Geographical Information Science 23, 605–630.

Zhao, W., Ma, H., He, Q., 2009. Parallel k-means clustering based on mapreduce, in: Cloud Computing. Springer, pp. 674–679.