Scale Handling for Land Use/Cover Change in an Era of Big Data

Jin Xing

Department of Geography
McGill University, Montreal

November 2017

A thesis submitted to McGill University in partial fulfillment of the requirements of the degree of Doctor of Philosophy.

Table of Contents

List of Figures

List of Tables

Abstract

Big data promises numerous benefits for Land Use/Cover Change (LUCC) research, in terms of increased volume, velocity, and variety of remotely sensed imagery datasets. However, it challenges traditional approaches to identifying LUCC. The increased volume (file size) and velocity (speed) of big data mean that existing data handling frameworks may not be able to effectively distribute spatial data and computation across a large number of computers. Previous LUCC workflows are not designed for big data and they cannot be easily deployed on big data computing tools such as cloud computing or the Hadoop framework. High levels of scale heterogeneity mean that images can cover different spatial and temporal granularities and extents. Theoretically, it becomes difficult to handle the data because these multiple and conflicting scales exist contemporaneously. Because we are working with big data, geographic entities may be recorded at different granularities and extents than should be detected as LUCC, but cannot be. Finally, no one has yet combined each of these distinct problems to fully examine all of the big data challenges facing LUCC.

I present six advances to address each of the big data challenge in LUCC: (1) a theoretical concept called Scope, (2) a spatially sensitive decomposition/recomposition method, (3) a scale invariant change detection method, (4) a spatial-temporal model for LUCC big data, (5) a change boundary optimization algorithm, and (6) a LUCC-specific Geospatial CyberInfrastructure. In this manuscript, I first propose Scope as a concept to model spatial-temporal scales by explicitly merging granularity, extent, time, and property. Second, I develop a new decomposition/recomposition framework to manage data decomposition, distribution, and recombination in a distributed computing environment. Third, a scale invariant change detection method identifies LUCC by combining regional and point features from datasets at multiple spatial granularities and extents. Fourth, I theorize a spatial-temporal object model to improve the integration of space and time within LUCC research. The spatial-temporal object model and, the fifth advance, a change boundary optimization algorithm handle data noise and better organize the spatial-temporal object changes. Finally, a Geospatial CyberInfrastructure combines these separate approaches with cloud computing and distributed computing frameworks as a holistic approach for the big data challenge in LUCC research. These six advances are tested in a series of case studies using datasets collected from 2005-2012, at the Greater Montreal Area.

Abrégé

Les mégadonnées promettent de nombreux avantages pour la recherche sur les Changements dès l'Utilisation/Couverture des Terres (CUCT), en termes d'augmentation du volume, de la rapidité et de la variété des images de télédétection. Cependant, il conteste les approches traditionnelles pour identifier des CUCT. Le volume accru (taille du fichier) et la rapidité (vitesse) des mégadonnées signifient que les cadres de gestion des données existants ne peuvent pas distribuer efficacement des données spatiales et des computations sur un grand nombre d'ordinateurs. Les flux de travail des CUCT précédents ne sont pas conçus pour des mégadonnées et ne peuvent pas être facilement déployés sur des outils mégadonnées tels que le cloud computing et le cadre Hadoop. Les niveaux élevés d'hétérogénéité des échelles signifient que les images peuvent couvrir granularités et étendues différentes, spatiales et temporelles. Théoriquement, il devient difficile de s'occuper les données à cause de significations d'échelle multiples et conflictuelles. Parce que nous travaillons avec les mégadonnées, les entités géographiques peuvent être enregistrés à granularités et extensions différentes qui devraient être détectées comme des CUCT mais ne peuvent pas être. Enfin, ces problèmes n'ont pas été explorés ensemble pour les défis des mégadonnées dans les CUCT.

Je présente six avancées pour répondre aux défis des mégadonnées dans les CUCT: (1) un concept théorique s’appelé Scope, (2) une méthode de décomposition / recomposition spatialement sensible, (3) une méthode de détection de changement invariant à l'échelle, (4) un modèle spatio-temporel pour les mégadonnées dans les CUCT, (5) un algorithme d'optimisation des limites de changement, et (6) une geospatial cyberinfrastructure spécifique aux CUCT. Dans ce manuscrit, je propose d'abord le Scope comme un conception pour modéliser les échelles spatio-temporelles en fusionnant explicitement la granularité, l'étendue, le temps et la propriété. Deuxièmement, je développe un nouveau cadre de décomposition / recomposition pour gérer la décomposition, la distribution, et la recombinaison des données dans un environnement computation distribué. Troisièmement, une méthode de détection de changement invariable à l'échelle identifie CUCT en combinant les caractéristiques régionales et ponctuelles ensembles des données à plusieurs granularités et étendues spatiales. Quatrièmement, je théorise un modèle d'objet spatio-temporel pour améliorer l'intégration de l'espace et du temps dans la recherche des CUCT. Le modèle d'objet spatio-temporel, et la cinquième avancée, un algorithme d'optimisation des limites de changement gèrent le bruit des données et organisent mieux les changements de

l'objet spatial-temporel. Enfin, une geospatial cyberinfrastructure combine ces approches distinctes avec le cloud computing et les cadres computations distribués comme une approche holistique pour le défi des mégadonnées dans la recherche des CUCT. Ces six avancées sont testées dans une série d'études de cas en utilisant des données collectées depuis 2005 à 2012, dans la région plus grande de Montréal.

Dedication

To Wen Wu and Ewan Xing

For their love and support

Acknowledgments

Foremost, I would like to express my sincere gratitude to my primary supervisor Prof. Renée Sieber, who is an associate professor at the Department of Geography in McGill University, for her continuous guidance and support over the six years of my Ph.D. studies. With her inspiration, patience, and encouragement, I have learned how to put forth an argument as a research scientist and completed my Ph.D. work.

I also give special thanks to my co-supervisor Prof. Margaret Kalacska, who provided considerable support on remote sensing image analysis. Prof. Kalacska is an associate professor at the Department of Geography, McGill University. I sincerely appreciate her advice on how to handle big data and with scale invariant image analysis algorithms.

I also express my gratitude to my committee member Prof. Shaowen Wang, who is a professor at the Department of Geography and Geographic Information Science, University of Illinois Urbana-Champaign, for his priceless advice on Geospatial Cyber-Infrastructures and CyberGIS. Without his insights, I could not have finished Chapters 3 and 6.

Help from Prof. Terrence Caelli was invaluable. Prof. Caelli is a professor at the Department of Electrical and Electronic Engineering, The University of Melbourne. His knowledge in computer vision and guidance helped me develop the scale invariant change detection method.

My own work was also supported by multiple organizations. The cloud computing resources utilized in this study were provided by Amazon Elastic Cloud Computing platform and Microsoft Azure cloud. My research funding came from the Global Environmental and Climate Change Centre at McGill University and Geothink, a partnership research grant funded by the

Social Sciences and Humanities Research Council of Canada. I also extend my gratitude to McGill University's Geographic Information Centre for the remotely sensed imagery datasets they provided and their kind assistance.

Finally, I acknowledge the limit of my English. I have worked diligently to find and correct these errors, although there likely will be some typographical errors and grammatical issues remaining in this dissertation. I am thankful to my supervisors Prof. Renée Sieber and Prof. Margaret Kalacska; my committee members Prof. Shaowen Wang and Prof. Terrence Caelli; and my friends Dr. Drew Bush and Mr. Suthee (Peck) Sangiambut for their help in polishing this dissertation.

Preface and Contribution of Authors

I. Dissertation Format

My dissertation is written in manuscript format according to the online instructions from the Graduate and Postdoctoral Studies Office and the Graduate Student Handbook from the Department of Geography, McGill University. I delineate the main research question and the overview of my dissertation in the Introduction Chapter. I review the literature and the main challenges of my research in Chapter 2. In Chapter 3-6, I focus on solving specific problems in my research questions. These four chapters (Chapter 3-6) have either been published in the peer-reviewed journals or in preparation for submission. Statements before each of these four chapters connect the individual chapters to the larger research questions and assist in the flow from one chapter to the next. The Conclusion Chapter summarizes the entire body of my work and describes my future research directions.

II. Contribution of Authors and Statement of Originality

I am the first author on all the four articles included (Chapter 3-6) in this dissertation. My supervisor Prof. Sieber helped me in the development of research questions, methodology design, data analysis, and result interpretation for all four manuscripts. The details about contribution of authors and the statement of originality are listed as follows.

Chapter 3 is my paper "*A New Measurement for Multiscale Analysis in GIScience*", which is being prepared for submission for *International Journal of Geographic Information Science*. I wrote this paper in collaboration with Prof. Sieber and Prof. Wang. I provided the idea of creating the new scale model with the initial concept of Scope; Prof. Sieber helped me in refining the model. Prof. Wang helped me develop the Scope Set and Scope quadruple projection

method. This chapter introduced the concept of Scope, which combined spatial granularities, spatial extents, time, and property to model scale changes in different geospatial analyses.

Chapter 4 was published as "*The challenges of image segmentation in big remotely sensed imagery data*", in *Annals of Geographic Information Science*, 2014. I co-authored this paper with Prof. Sieber and Prof. Kalacska. The decomposition/recomposition workflow was inspired by a discussion with Prof. Sieber, and we co-drafted this paper. Prof. Kalacska provided the testing data and enhanced the workflow. The chapter covered the decomposition/recomposition framework to handle the big data workflow, especially within the distributed computing environment.

Chapter 5 comes from my paper "*Scale Invariant Land Use/Cover Change Detection Method*", coauthored with Prof. Sieber and Prof. Caelli. I proposed the approach of integrating spatial, spectral, and scale information with the Scale Invariant Feature Transformation (SIFT)-based change detection algorithm. Prof. Sieber helped me refine the concept. Prof. Caelli provided technical and language support. This paper is currently under review in *ISPRS Journal of Photogrammetry and Remote Sensing*. The scale invariant Land Use/Cover Change detection method extracts change areas by comparing scale invariant image features, calculated from imagery datasets with different spatial granularities and extents.

Chapter 6 was originally published as "*A land use/land cover change geospatial cyberinfrastructure to integrate big data and temporal topology*" in *International Journal of Geographic Information Science*, 2016. I co-authored this paper with Prof. Sieber. My original contribution is the idea of combining spatial optimization and computation optimization methods within geospatial cyberinfrastructure. Prof. Sieber added temporal topology and spatial-temporal object model as a just-in-time optimization for LUCC. We drafted this paper together and Prof.

Sieber improved readability. This chapter introduced the methodology of using geospatial cyberinfrastructure as a methodology to integrate spatial optimization, temporal optimization, and computation optimization to solve the big data challenge in LUCC.

Chapter 1. Introduction

The rapid development of Remote Sensing (RS) platforms and Geographic Information Systems (GIS) has produced data at an unprecedented speed. This increased availability of data has benefited Geographic Information Science (GIScience) research and brought details and more coverage to sub-discipline of Land Use/Cover Change (LUCC). Disadvantages exist as well, including the growing heterogeneous scales, miscellaneous spectral bands, diverse data formats, and complex analysis methods. In the past, LUCC researchers could assume that the underlying levels of computation were adequate. No longer. The big data challenge has exceeded traditional methods of analysis, and RS-based LUCC begins to rely on analytics distributed across varying computing platforms. Advances in computation have given researchers the opportunity to address spatial-temporal scale heterogeneity with varying levels of computation support (Wang, 2010). In this dissertation, I investigate RS-based LUCC analysis with big data that accommodates heterogeneous spatial-temporal scale analysis.

RS-based LUCC is the process of detecting differences in geographic entities or phenomena at the same location across different time periods using RS imagery datasets. For more than 50 years, the analysis of multi-temporal RS datasets to better understand LUCC has been an active research field in both GIS and RS (Singh, 1989). RS-based LUCC research aims to address three questions: (1) Is there any actual LUCC (i.e., not caused by the noisy information) when comparing two or more temporally distanced datasets?; (2) What are these changes quantitatively?; and (3) What are the change rates and trajectories of the LUCC?. The advent of earth observation systems has also provided an opportunity to collect geographic information relevant to LUCC at different spatial, spectral and temporal scales (Longley, 2002). However, advances in sensing platforms have resulted in an unprecedented big data challenge

(Miller and Goodchild, 2015) because of the huge volume and increased velocity of various multi-temporal data sources. Much of this data also contains noisy information.

Big data provides opportunities for RS-based LUCC research while also complicating many investigations. Big data is characterized as the "4Vs": volume, variety, velocity, and veracity (Laney, 2001). Each characteristic creates domain-specific challenges in RS-based LUCC research. The large volumes of big data have provided finer spatial resolutions for detailed change identification over larger areas and longer time spans. This high level of variety means that change information can be extracted in different file formats (e.g., GIS shape files, hyperspectral RS datasets, aerial photos, and text records), and at different scales (i.e., spatial, spectral, and temporal granularities and extents). Increased velocity (i.e., the speed at which data is produced) can also provide shorter data collection intervals that will result in more data and, potentially, more varied temporal scales of LUCC. For example, a wild fire can burn down a forest in several hours but the revitalization of the forest will likely take decades (Huettl, 1988). RS-based LUCC studies require shorter intervals for the moment of a wild fire (at minutes or seconds intervals) and longer time spans to capture forest (instead of such fine temporal resolutions). Veracity is another important factor for RS-based LUCC detection because there must be ways to "ground-truth" the results generated by the sheer volume, variety, and velocity of data. However, the ability to ground-truth LUCC detection results often would be hampered by the volume and variety of data. Previous research developed numerous RS-based LUCC detection algorithms in an era of "scarce data." Researchers believe these methods may now prove insufficient for big data (Miller and Goodchild, 2015).

Big data also brings specific computational challenges to RS-based LUCC research. Because RS imagery now routinely exceeds the computational infrastructure of most desktop

computers, researchers must split up large RS datasets so they can be analyzed on a number of distributed computing nodes. This splitting requires consideration of parallel computing management, resource provisioning, and dataflow control over the distributed systems. Big data brings additional computational challenges because datasets are acquired at various scales; and detecting LUCC across heterogeneous datasets requires additional computation for spatial scaling operations (Yunfeng et al., 2013) or extracting scale invariant features for LUCC identification (Chiu et al., 2013). Finally, the workflow complicated by the addition of big data to RS-based LUCC will consume more computing resources for data Input/Output (I/O) and job scheduling. Consequently, the advent of big data has driven RS-based LUCC research to rely more on Geospatial CyberInfrastructure (GCI) for computation management (Wang et al., 2013).

RS-based LUCC research in an era of big data also requires a re-examination of the concept of scale. The challenge results not only from a variety of complex meanings for scale but also RS-based LUCC requires a methodology for comparing imagery datasets with spatial-temporal scale heterogeneity. One method is to transform the heterogeneous scale data into homogeneous scale via geospatial scaling operations. However, the spatial granularity and extent difference in scale heterogeneous datasets prevent researchers from directly applying pixel-based LUCC detection algorithms without introducing noise. Often such noise comes from the up/down-sampling and other scaling techniques that regularize granularity and extent. Another method is to look for image features that are spatial scale invariant. But the LUCC field still lacks an efficient approach to model this kind of scale invariance. In other words, a need exists to ensure that the spatial scale heterogeneity of LUCC data is not so big that it would invalidate the scale invariant image features. Finally, combining the spatial-temporal scale models in big data analysis with the computational scale (Frey et al., 2002) is also important to better handle computing resource provisioning and dataflow management.

In this dissertation, I address the scale challenges of big data in RS-based LUCC detection with a new scale model, a new dataflow management framework, a scale invariant LUCC detection algorithm, and a LUCC-GCI computing platform. To model scales within the complex context of big data analysis in LUCC, I propose the concept of Scope. I develop a decomposition/recomposition framework for data handling across the distributed systems. I also invent a scale invariant LUCC detection algorithm that compares spatial scale invariant image features to avoid the additional errors incurred by the geospatial scaling operations in RS-based LUCC detection. Finally, I employ GCI as the methodology to synthesize all these separate solutions with cloud computing platforms and parallel computing frameworks (e.g., Apache Hadoop and Storm) (Borthakur, 2007) for big data analysis in RS-based LUCC detection.

The novel contributions of my dissertation are four-fold. First, I clarify various meanings of scale and propose the concept of Scope to model spatial-temporal scaling operations in LUCC. Second, I develop a dataflow management framework for big data handling that can be applied to GIS and RS big data analysis. Third, I create the scale invariant LUCC detection algorithm based on the Scale Invariant Feature Transformation (SIFT) (Lowe, 2004) and Maximally Stable Extremal Region (MSER) (Matas et al., 2004) to avoid the noisy information incurred by geospatial scaling operations in LUCC. Fourth, I integrate LUCC workflow and advanced computing techniques within a Geospatial CyberInfrastructure (GCI) as LUCC-GCI to address the big data challenge in LUCC. My dissertation combines the new scale modeling method, dataflow management framework, change detection algorithms, and advanced computation techniques to better handle scale in big RS-based LUCC research.

The scientific findings of my dissertation are summarized as follows. First, I use sematic modelling to clarify and integrate complex theories in scale and scaling in GIScience and RS

research. The dataflow management does not only provide a general geospatial data handling approach, but also highlight the topological and geometric information which distinguish geospatial big data from an aggregation of small data. The success of the scale invariant LUCC detection algorithm proves that computer vision algorithms can be employed to address the scale heterogeneity challenge in RS image analysis, with a careful handling of spatial-temporal scales. Finally, LUCC-GCI illustrates the increasing integration of domain specific knowledge and high performance computing, shaping GCI as a subdomain of GIScience rather than a tool.

The rest of this thesis is organized as follows. Chapter 2 reviews the recent research in RS-based LUCC with a focus on big data processing articles. In it, I highlight the spatial-temporal scale challenge incurred by big data. Chapter 3 introduces the concept of Scope to combine spatial granularity, extent, time, and property for scale modelling in GIScience. Chapter 4 focuses on dataflow handling of big data by proposing the decomposition/recomposition framework on top of a distributed computing system. Chapter 5 delineates a new scale invariant LUCC detection algorithm that relies on the comparison of scale invariant image features for RS-based LUCC detection. Chapter 6 presents a LUCC-GCI to combine the spatial optimization algorithm, the decomposition/recomposition framework, and spatial-temporal geographic models with cloud computing and Apache Storm framework (Apache, 2017). The LUCC-GCI offers a holistic solution for big data analysis in LUCC. Finally, I conclude my dissertation with its implications and point out future research directions in Chapter 7.

Reference

Apache, Apache Storm (2017). Online: https://storm.incubator.apache.org/

Borthakur, D. (2007). The hadoop distributed file system: Architecture and design. Hadoop Project Website, 11(2007), 21. 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.

Chiu, L. C., Chang, T. S., Chen, J. Y., & Chang, N. Y. C. (2013). Fast SIFT design for real-time visual feature extraction. IEEE Transactions on Image Processing, 22(8), 3158-3167.

Frey, J., Tannenbaum, T., Livny, M., Foster, I., & Tuecke, S. (2002). Condor-G: A computation management agent for multi-institutional grids. Cluster Computing, 5(3), 237-246.

Huettl, R. F. (1988). "New Type" Forest Declines and Restabilization/Revitalization Strategies. In Restoration of Aquatic and Terrestrial Systems (pp. 95-109). Springer Netherlands.

Laney, D. (2001). 3D data management: Controlling data volume, velocity and variety. META Group Research Note, 6, 70.

Longley, P. A. (2002). Geographical information systems: will developments in urban remote sensing and GIS lead to 'better'urban geography?. Progress in Human Geography, 26(2), 231-239.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. International journal of computer vision, 60(2), 91-110.

Matas, J., Chum, O., Urban, M., & Pajdla, T. (2004). Robust wide-baseline stereo from maximally stable extremal regions. Image and vision computing,22(10), 761-767.

Miller, H. J., & Goodchild, M. F. (2015). Data-driven geography. GeoJournal, 80(4), 449-461.

Singh, A. (1989). Review article digital change detection techniques using remotely-sensed data. International journal of remote sensing, 10(6), 989-1003.

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(11), 2122-2145

Wang, S. (2010). A CyberGIS framework for the synthesis of cyberinfrastructure, GIS, and spatial analysis. Annals of the Association of American Geographers, 100(3), 535-557.

Yunfeng, H., Zhiying, X., Yue, L., & Yan, Y. (2013). A review of the scaling issues of geospatial data. Advances in Earth Science, 28(3), 297-304.

Chapter 2. Literature Review and Scale Challenges in Land Use/Cover Change for Big Data Analysis

Abstract

In this chapter, I review the refereed literature on Remote Sensing (RS)-based Land Use/Cover Change (LUCC) research and summarize the new challenges brought by geospatial big data, especially those related to scale. This review first starts by defining RS-based LUCC before turning to an examination of the main data sources and detection methods used in the field. Second, I focus on spatial-temporal scale in big data for RS-based LUCC research and pay particular attention to the meaning of scale, the modelling of scale, and the role of heterogeneous scales in RS-based LUCC studies. I also examine the most recent big data processing articles that involve geospatial cyberinfrastructure because such tools have only recently been used in RS-based LUCC research. Finally, I summarize my Ph.D. research questions in relation to the challenges created by using big data in LUCC research. These include the handling of spatial-temporal scales, scale modelling challenges, big data workflow challenges, scale heterogeneous LUCC detection challenges, and big data computational challenges. Because big data exhibits various scale challenges as a whole, we cannot treat is as a simple aggregation of small data.

2.1 LUCC Literature Review

There has been a long history of using Remote Sensing (RS) imagery in Land Use/Cover Change (LUCC) research. Such work incorporated a broad range of Geographic Information Science (GIScience) and Remote Sensing (RS) topics. Previous to the use of such technologies, land cover change and land use change were considered separate research domains. Land cover change research investigated changes to the Earth's surface including to forest (Lambin, Geist, and Lepers, 2003), water (Vörösmarty et al., 2000), soil (Lal, 2004), desert (Alpers and Brimhall, 1988), and wetland (Erwin, 2009) systems. In contrast, research into land use change sought to

describe the transformations to landscapes caused by human activities such as agriculture (Yu and Lü, 2008), urbanization (Brenner, 1998), and the building of roads and public transportation (Waddell, 2002). Turner, Meyer, and Skole (1994) reported on a tight linkage between land cover change and land use change and, consequently, argued for their combination into the field of 'LUCC' or land use and cover changes. In this dissertation, I mainly focus on employing RS datasets for LUCC study. Therefore, I use LUCC to stand for RS-based LUCC.

Researchers have studied LUCC across a wide variety of spatial-temporal scales. Examples of the spatial scales (or spatial extents) considered included studies on local LUCC in Zhujiang Delta area, China (Weng, 2002) and a review of all global LUCC research internationally (Xiubin, 1996). The temporal scales studied included Goldewijk (2001) who presented a geo-database covering 300 years of LUCC, and Yuan et al. (2005) who detected LUCC by comparing images taken at four different times (1986, 1991, 1998, and 2002) for the Twin Cities (Minneapolis and St Paul, MN) in the United States. The increasing variety of spatial-temporal scales has contributed to the diversity of LUCC research.

There has been a broad range of interpretations on what drives LUCC. Lambin and Meyfroidt (2001) saw the important drivers of LUCC as agriculture, urbanization, and globalization. Research scientists also employed social (Foley et al., 2005) and economic (Veldkamp and Lambin, 2001) factors to interpret LUCC. Consideration of LUCC contextual factors involving scale served as the basis for accurate analyses, especially in an era of big data. I define big data as datasets that are so huge and complex that traditional analytical methods are inadequate to handle. In the following literature review, I first present exiting datasets and detection methods in LUCC study. I then delineate the scale challenges incurred by big data. To

finish, the computational support needed for LUCC processing is examined in relation to the demands of big data.

2.1.1 Dataset and Analysis Approaches in Land Use/Cover Change Detection

During the 1950s to 1970s, airborne sensors were the main source of image datasets for LUCC (Dueker and Horton, 1972). That changed in the 1970s when satellite imagery became more frequently used and increased the spatial, spectral, and temporal resolutions offered (Bianchin and Bravin, 2008). One of the most widely studied RS satellites was Landsat family (L1-L8). Their sensors included Thematic Mapper (TM), Enhanced Thematic Mapper (ETM), Enhanced Thematic Mapper plus (ETM+), Multi-Spectral Scanner (MSS), and panchromatic. This family of RS sensors provided moderate resolution images (~30m) and were actively used in LUCC research. Another moderate resolution satellite family was Terra satellite with Advanced Spaceborne Thermal Emission and Reflection Radiometer (ASTER) sensor and MODerate-resolution Imaging Spectroradiometer (MODIS) sensor (Stefanov and Netzband, 2005). Moreover, high- and very high-resolution satellite sensing systems included IKONOS, QuickBird, SkySat, WorldView and GeoEye, with spatial resolutions of those sensing systems equalled to or finer than 1m. Various airborne sensors supplemented spaceborne sensors for the study of LUCC. The most often utilized was the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) (Schowengerdt, 2006).

To enhance the detection of LUCC, researchers sometimes incorporated three-dimensional information with the Light Detection And Ranging (LIDAR) sensor. LIDAR utilized a laser pulse (usually at near infrared band) to measure the time distance from the sensor to a reflecting object. Using LIDAR imagery, three-dimensional position and reflectance characteristics of the studying object could be calculated. In 2004, Vu et al. argued that LIDAR

constituted an efficient tool for change detection even of complex urban infrastructures. Moreover, Synthetic Aperture Radar (SAR) images could provide complementary observation information in some conditions when it was difficult for optical sensors to acquire high quality imagery dataset (Henderson and Lewis, 1998). The growing number of sensing platforms provided rich sources of data for LUCC. As will be discussed, they also increased the variety of data and the difficulty of LUCC analysis (Hardie and Parks, 1997).

Most LUCC utilized raster data that was structured via a field-based model. Geospatial vector data was not as frequently utilized in LUCC as raster data. Malczewski (2004) reviewed the Geographic Information System (GIS) approaches in LUCC and summarized the geospatial vector datasets. In LUCC, these GIS datasets included map sheets, plan maps, surveys, cartographic models, database records, and social media data (e.g., harvested comments from platforms like Twitter). Dai, Lee and Zhang (2001) integrated topography, surficial and bedrock geology, groundwater conditions, and historic geologic hazards GIS (vector or object-based) datasets to study LUCC in Lanzhou, China. Li and Yeh (2002) built a cellular automata land use model based on the urban centre, road, and administrative boundary vector layers. Researchers also combined GIS and RS approaches for LUCC detection. For example, Weng (2002) applied vector spatial-temporal modelling with raster imagery datasets for LUCC analysis in Zhujiang Delta area, China. Vector data tended not be used as much because it could record and model LUCC with higher flexibility than RS—and, as a result, required additional homogenization works for LUCC identification (Parker et al., 2003). Most of the time, vector data was only utilized to assess the accuracy of the raster data used in a LUCC study (Dai, and Khorram, 1998).

So far, this literature review has covered the basic platforms that provide data for LUCC analysis. The workflow for analyzing data in a LUCC study could be summarized as three steps:

(1) pre-processing (although vector-based data might not require such pre-processing), which reduced the impact of differences in image data caused by factors other than LUCC (e.g., atmospheric conditions, sensing system difference, illumination and viewing angles' impact, and soil moisture); (2) selection and application of change detection algorithm(s) to the pre-processed (if any) dataset; and, (3) accuracy assessment of the change detection results. These three steps are depicted in Figure 2-1. The boundaries between these three steps were not fixed with any two or even three of steps often integrated as an inseparable process (Mucher et al., 2000). For Step (1), the most widely applied techniques are: atmospheric correction, radiometric correction, geometric correction, and image registration (Lu et al., 2004). For Step (3), an error matrix is the most widely employed tool for the accuracy assessment in LUCC studies (Singh, 1989). In the error matrix, the columns represented the ground-truth data while the change detection results are shown in rows (Morisette and Khorram, 2000). An error could either be a changed area that is mislabelled as "no-change," or an unchanged area that might be mistaken as "change." An error matrix could also be employed using classification labels (Congalton, 1991) to better investigate the accuracy of specific classes or features (Chen et al., 2012).

The rest of the LUCC detection review focused on Step (2), or the change detection algorithms that identified LUCC from the GIS and RS datasets. In Step (1), most RS data had already been processed with the atmospheric correction, radiometric correction, and geometric correction with the modern sensing platforms. Most of these algorithms have been well-studied (Kaufman et al., 1997; Teillet, 1986; Toutin, 2004). Since big data introduced numerous heterogeneous spatial resolutions and extents, geo-referencing was more frequently applied than the image registration techniques (Xiang and Tian, 2011). The former, which linked image entities with latitude and longitude, was preferred over the latter, which found similar image features as control points that are used to align different images. I did not cover the

implementation details of Step (1) because most RS datasets in this dissertation have already been processed with atmospheric correction, radiometric correction, geometric correction, and geo-referencing. However, I followed the predominant literature to use error matrices as a measure of accuracy for step (3), as will be demonstrated in Chapters 5 and 6.

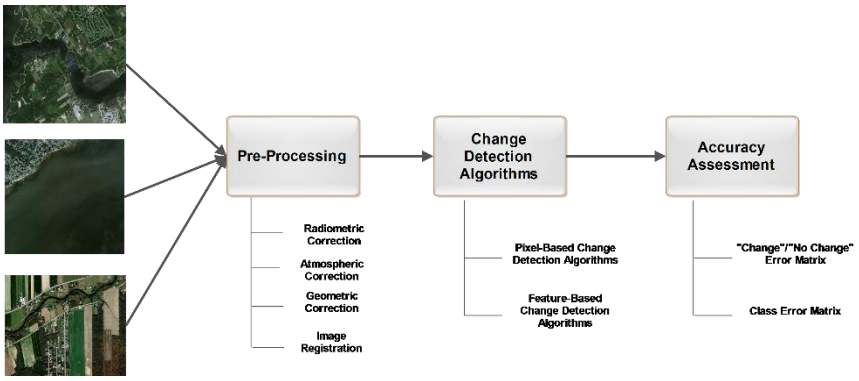


Figure 2-1. Three Steps of LUCC detection. Images are extracted from DMTI StreetView imagery datasets recorded at Montreal, 2006 (DMTI Spatial Inc., 2006).

Almost all LUCC was identified via the application of change detection algorithm(s). Generally, the change detection algorithms are applied to at least two images that are taken at the same location but at different times. However, there are many change detection algorithms used for this purpose that could be categorized into two groups based on the data structure they wish to extract: pixel-based or feature-based. Pixel-based approaches extract LUCC from the difference among pixel values at the same location from at least two images. Feature-based algorithms extract various image features and identify LUCC by comparing these features. Singh (1989) published the first survey of change detection algorithms and listed the most widely applied pixel-based approaches. Lu et al. (2004) presented another review of pixel-based change

detection algorithms that together with thresholding techniques that could be used to determine the level at which a pixel was labelled as no-change or change. Lu et al.'s (ibid.) paper differed from Singh's (1989) because it emphasized a LUCC workflow instead of change detection algorithms. There are more pixel-based change detection algorithms than could be named here but a few include image differencing (Stauffer and McKinney, 1987), image ratioing (Short, 1982), spectral index differencing (Lunetta et al., 2006), change vector analysis (Chen et al., 2003), principal component analysis (Byrne, Crapper, and Mayo, 1980), multivariate alternation detection (Liang et al., 2011), and Kauth-Thomas transformation (Kauth and Thomas, 1976). Pixel-based methods depend on pre-processing methods such as image registration. That is because most pixel-based methods require the accuracy of image registration to be less than the pixel size (Townshend et al., 1992). These methods also assume that all the images have the same spatial and spectral resolutions (Coppin and Bauer, 1996). However, advances in sensing platforms are producing imagery datasets with increasingly fine spatial, spectral, and temporal resolutions over increasingly large spatial-temporal extents. I argue that existing pixel-based methods and the associated research did not easily accommodate this variability. Consequently, limitations exist for the direct application of the pixel-based change detection algorithms to these new and scale heterogenous datasets.

I contend that the high variety of big data in LUCC (which I write about below) demands a shift from a pixel-based to a feature-based change detection algorithm. In change detection algorithms, a feature is defined as a group of pixels that are different from their neighbours. Features in such algorithmic processes should not be confused with features as defined in an object-based (vector) data structure. Because I am examining LUCC detection algorithms in this review, however, I mainly concern myself with features from raster image data. An example of feature-based change detection methods is the entropy texture-based change detection used by

Rosin and Ioannidis (2003). In their work, the authors used entropy to identify the degree of local data heterogeneity (i.e., the sum of pixel value difference between one pixel and its neighbours) that can be quantified within the given analysis window. Other researchers, such as İlsever and Ünsalan (2012), had utilized a fixed (11*11 pixels) analysis window. Once the difference of entropy between two images are calculated, thresholding techniques can classify the images as change or no-change areas. One consequence of this process is that image features are less dependent on a single pixel value when used in LUCC detection. This makes change detection algorithms far more robust when compared to pixel-based methods (Toure et al., 2016).

There are also different types of feature-based change detection algorithms. Furthermore, a large body of features could be employed in LUCC including: (1) image features such as color features, shape features, texture features, local features, and global features (Ping Tian, 2013); (2) RS features such as normalized difference vegetation index, enhance vegetation indices, principal components, and canonical varieties (Pohl and Van Genderen, 1998); and, (3) GIS features such as agent-based models, cellular automata snapshots, vector polygons, plan maps, urban growth models, expert systems, and crowd-sourcing (Goodchild, 2010). These features are usually combined for LUCC study. For example, Benz et al. (2004) merged RS image cluster and texture features with polygon GIS maps for LUCC detection in Austria. Their study indicated the important role of feature-based approached in bridging raster and vector datasets for use in LUCC detection.

One feature-based algorithm that attracts considerable interest is the segmentation-based or object-based (Chen et al., 2012) change detection algorithm. This method merges various image features (e.g., geographic area, shape, and texture) to improve LUCC and reduces data "noise" (Blaschke, 2010) by combining the advantages of these features. The general steps of

segmentation-based change detection are illustrated in Figure 2-2. The image segmentation technique (Pal and Pal, 1993) first groups similar pixels into "segments." LUCC is identified by searching for corresponding segments in other datasets and comparing the characteristics of the segment-level features such as boundaries, shapes, areas, and spectral statistics (Gong et al., 2008). The accuracy of the algorithm largely depends on the accuracy of the segment edges and the search for the spatial-temporally "corresponding" segments (i.e., geographic objects at the same location but different times). A segmentation-based change detection method could further employ various raster and vector analysis methods such as edge detection (Maini and Aggarwal, 2009), grey level co-occurrence matrix (Marceau et al., 1990), and spatial buffering analysis (Lunetta et al., 2006). Chen et al. (2012) argued the merging of different methods could significantly improve the accuracy of LUCC detection.

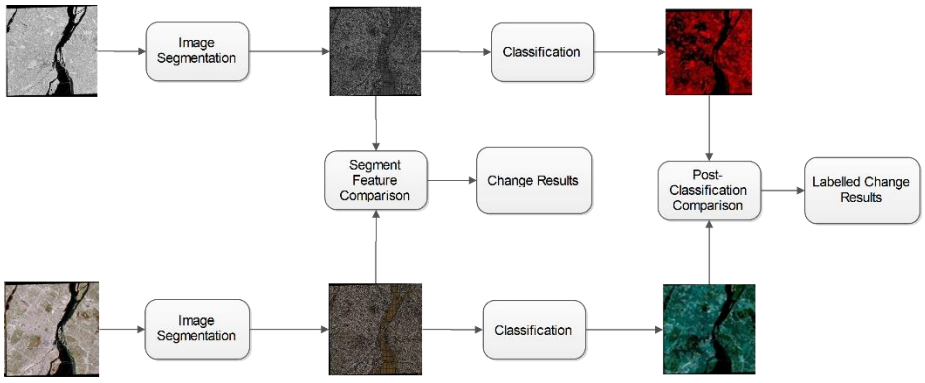


Figure 2-2. Segmentation-Based Change Detection Algorithms, illustrated using DMTI StreetView image taken at Montreal in 2006 and 2009, respectively (DMTI Spatial Inc., 2006 and 2009).

Finally, LUCC researchers have explored the fusion of pixel- and feature-based change detection methods. Li and Davis (2008) extracted image features from scale heterogeneous images and employed fuzzy logic rules to classify the difference between the image features for LUCC detection in the Cities of Phoenix and Springfield in the United States, using Quickbird and Ikonos image datasets. Liu et al. (2011) presented a decision-level LUCC framework that

utilized multiple different images generated by the image differencing algorithms and a Chi-square transformation algorithm to produce the combined change map. Such fusions of various change detection methods often improve the accuracy of LUCC studies (Hecheltjen et al., 2014). However, the fusion of pixel- and feature-based LUCC detection algorithms also increases the computational workload and complicates the LUCC process, leaving the fused algorithm more sensitive to the errors and noisy information in each change detection algorithm that it integrates.

2.1.2 Scale in LUCC

Scale has played a pivotal role in LUCC because LUCC areas are expressed at specific granularities and extents spatial-temporally. Data enteres the LUCC analysis at a certain spatial granularity or spatial resolution (e.g., 250m of MODIS data), and datasets for at least two points in time must be compared to detect LUCC. In such an analysis, it must be determined whether an object, for example a forest, had changed based on any spatial or temporal scales (i.e., seasonal forest changes are not considered as valid LUCC). Most change information also could only be extracted within a fixed spatial-temporal scale (Cash et al., 2006). A forest, for example, exists for a certain temporal duration and at a certain spatial extent and, if it is recorded as RS images, is acquired at given spatial resolution and extent. At too high a granularity we see the forests as individual trees; at too low a granularity the forest may get lost amid other features. Usually, we tend to analyze LUCC at fixed temporal scales, for example, every five years. If we receive data at different time periodicities, then we have to interpolate. Therefore, spatial and temporal scales constitute an innate part of LUCC that interact with all the processes in a LUCC analysis.

In GIScience, scale has been defined in various ways and this presented a problem to LUCC. Goodchild (2011) defined scale as the combination of granularity and extent needed to describe the quality of geospatial datasets. Wu and Li (2009) summarized the meanings of scale

with respect to different research domains, including observation, modelling, operation, geography, policy and cartography. Zubin (1989) presented four cognitive geography scales, ranging from the small everyday objects to regions that exceed our existing experience and knowledge. In social theory, Marston, Jones, and Woodward (2005) employed an ontology-based graph model to define and then problematize scale as hierarchical social entities. We still lacked agreement on a single definition of scale because the meanings of scale changed under different theories, operations, and algorithms (Goodchild, 2001). In LUCC, scale tends to refer to the spatial granularity of RS datasets (Celik, 2009) but it also frequently refers to spatial (Lambin and Meyfroidt, 2011) and temporal (Goldewijk, 2001) extents.

Scale can appear throughout the data handling process from observation and collection to analysis and visualization (Atkinson, 2001). The mismatch of data scale (i.e., the spatial-temporal resolution and extent at which the data was made available) and analysis scale (i.e., the spatial-temporal resolution and extent at which the analysis algorithm worked) has made scale modelling even more challenging. Scientists depict both data and analysis scales using fractal (Emerson, 1998), variogram (Legendre and Fortin, 1989), spatial entropy (Li and Yeh, 2004), and regression models (Lee, 2005). Data might be collected at one granularity but the analysis algorithm might operate on another. For example, we can not extract precise road information from a Landsat8 imagery dataset because most roads are narrower than the 30m resolution provided. A large number of spatial scaling operations (granularity, extent, and time) have been proposed to address this data/analysis scale mismatch. They include image scaling (Celik, 2009), pixel aggregation (Flowerdew, Geddes, and Green, 2001), spatial interpolation (Flowerdew and Green, 1993), granularity regularization (Atkinson, 2001), and zone design (Alvanides, Openshaw, and Macgill, 2001). All these techniques require different corresponding contextual information to determine the appropriate scales of both data and the corresponding analysis.

Big data analysis shifts defining scale even more complex by adding the concept of computational scale. Computational scale describes the computing resource allocation and data workflow required—especially within a distributed computing environment. Computational scale can be viewed as the configuration of computing nodes (granularity) and the number (extent) of computing nodes that covered a large number of parallel computing-based case studies (Stewart, 2015). If the computational scale is not coordinated with the LUCC workflow (Figure 2-1), researchers could face the risk of an input-output (I/O) bottleneck, data loss, job scheduling errors, and even system crashes (Lee, 2008). For example, inadequate computer memory could incur I/O bottlenecks when streaming large amounts of data. Any LUCC analysis applied to partial datasets in such contexts will lead to spurious results. Therefore, merging the concept of computational scale into the process of scale modelling is necessary for big data analysis.

2.1.3 Computation Support for Big Data Analysis and Geospatial Cyberinfrastructures

Kitchin (2013) noticed the sheer volume challenge attracted the most research interest among the "4Vs" of big data: volume, velocity, variety, and veracity. Since the data volume has far exceeded the capacity of a single computer, numerous computing approaches are employed in GIS and RS research to distribute the data among numerous machines for parallel analysis (Plaza et al., 2011). Among these approaches, cluster computing was widely adopted in the early stage of big data analysis (Ma et al., 2015). It connects a number of computers at the same physical location. Grid computing, by contrast, integrates computing nodes from different geographic locations and relies on middleware (i.e., a special piece of software that acted as the communication and management glue among other software or hardware) for job scheduling and resource consolidation (Sun et al., 2005). Cloud computing has been extensively used in GIScience research (Yang et al., 2011) for big data processing because many cloud computing

providers offer “pay-as-you-go” resource provisioning that offload the hardware management and maintenance to distant datacenters. Cloud computing also offers scalability because research scientists can subscribe thousands of Virtual Machines (VMs) without worrying about the hardware. By merging advanced computing techniques with GIS analysis methods, the goal is to develop efficient tools for big data handling.

The word CyberInfrastructure (CI) was first proposed by Richard Clarke in a White House press briefing in 1998 (Stewart et al., 2010). Later, the National Science Foundation established a blue-ribbon review team for CI research, in which the definition of CI was officially forged (Atkins, 2003). There are various definitions of CI and I cite the definition from NSF’S Cyberinfrastructure Vision for 21st Century Discovery (National Science Foundation, 2005, page 4):

> “The comprehensive infrastructure needed to capitalize on dramatic advances in information technology has been termed cyberinfrastructure.”

There is broad application of CI. Buetow (2005) summarized the wide application of CI in biology and medical research. Kim and Heller (2006) indicated the prominence of CI to host chemistry datasets for a large body of study including environment, public heath, food security, and transportations. Plale et al. (2006) developed a GCI for multiscale weather forecast. CI was also proposed for more than data handling. Hey and Trefethen (2005) highlighted the important role CI played in general scientific knowledge discovery.

CI has also been employed in GIScience, as the Geospatial CI (GCI). Yang et al. (2010, pp.265) defined GCI as:

> "Geospatial CI (GCI) refers to infrastructure that supports the collection, management, and utilization of geospatial data, information, and knowledge for multiple science domains."

GCI was reviewed by Yang et al. (ibid.), and the CI for RS research was covered by Gamon et al. (2010). GCI is a special type of CI that integrated geospatial data management, advanced computing techniques (both hardware and software), data analysis approaches, and GIScience as a new methodology for geospatial knowledge discovery and decision-making (Yang et al., 2010). Research in GCI includes numerous topics. For example, there is an ongoing research in job scheduling optimization for geospatial data analysis (Zhang and Tsou, 2009), semantic web with ontology knowledge systems (Sieber, Wellen, and Jin, 2011), climate and environment modelling (Droegemeier et al., 2004), applications to virtual organizations (Cummings et al., 2008), usage of volunteered geographic information analysis (Armstrong et al., 2011), and GCI-based education (Real, 2008). The diverse topics of GCI illustrate its broad potential in combining high performance computing with geospatial analysis. Therefore, GCI is gradually becoming a theoretic framework to combine domain knowledge and high performance computing than a computational tool, as big data challenge shifts RS-based LUCC more computation dependent.

GCIs aggregates up (i.e., worked across multiple computing nodes or virtual machines) and disaggregates down (i.e., handled a large volume dataset via decomposition) geospatial computing for big data analysis (Wright and Wang, 2011). In such processes, huge volumes of data could be decomposed by splitting the data into a large number of small chunks and processing those chunks in separate computing nodes. This process utilized by GCIs can be implemented with various computing techniques. Liang et al. (2010) proposed a GCI based on social networks and hybrid Peer-to-Peer techniques to enable sharing and visualization of big

environmental sensing datasets over a vast number of computers. Li et al. (2016) employed MapReduce-based (see below) GCI to retrieve and visualize scale heterogeneous datasets from 64 different sensors in real-time. However, GCI has not been employed to address the challenges big data introduced to the field of LUCC research.

Two important computing techniques enable GCI for big data processing (Yang et al., 2017). One is cloud computing that enabled on-demand computing resource provisioning via the Internet (Yang et al., 2011). The other one conneects the big data computation models needed to analyze such large, rapid, and heterogeneous data, including: the MapReduce-based distributed computing framework, parallel graph-based computation (e.g., Apache Spark) (Sun et al., 2015), bulk synchronous parallel computing model (e.g., Apache Hama) (Krause, Tichy, and Giese, 2014), and point-to-point computing model (e.g., message passing interface) (Qin, Zhan, and Zhu, 2014). Cloud computing enables the rapid setup of distributed computing platforms without the need for hardware configuration and maintenance. Big data computation models can then offload the management of distributed computing tasks and dataflow to the computation models.

The most common big data computation model is MapReduce (e.g., as implemented in the open source software Hadoop). It provides the software platform to distribute computing tasks and data over a large number of machines for parallel processing (Lee et al., 2012). Generally, there are two phases in MapReduce: the *map* phase and the *reduce* phase. The *map* phase decomposes a huge volume data into a large number of chunks as the *key/value* pairs (we usually use the file {chunk} id as the *key* and the file content as the *value*) and executes data analysis algorithms in parallel with the *key/value* pairs. The *reduce* phase receives the output from *map* nodes and combines them to generate the final results. MapReduce manages the execution of all *map* and *reduce* tasks and reschedules any failed tasks automatically onto other

computing nodes (Dean and Ghemawat, 2008). To manage the data distribution and combination operations in MapReduce, Shvachko et al. (2010) invented Hadoop Distributed File System. Examples of the use of MapReduce in GCIs include Schnase et al. (2016) using it for global climate change research, and Li et al. (2014) using it for contiguity weights matrix calculation in geospatial big data analysis. The combination of cloud computing and MapReduce framework has proven useful in big data analysis (Nurian et al., 2012), and GCI has also benefited from it (Li et al., 2016). For example, Gao et al. (2014) utilized Hadoop on Amazon Elastic Cloud Computing (EC2) (Amazon, 2017) to harvest and analysis crowd-sourced gazetteer entries from social media. However, these advanced techniques have not been employed for LUCC research yet.

2.2 Research Questions

In this chapter, I review the refereed literature on LUCC workflow and LUCC detection algorithms, scales in LUCC, and GCIs for big data analysis. In LUCC, big data brings large volumes of data with increasing scale heterogeneity. I argue four challenges exist that are incurred by the usage of big data in LUCC studies: (1) the scale modelling challenge for heterogenous datasets; (2) the workflow challenge to address big data volume; (3) the LUCC detection algorithm challenge for scale heterogeneity handling; and (4) the computational challenge of big data analysis in LUCC.

2.2.1 Scale Modelling Challenge

LUCC research requires a new model to clarify the complex meanings of scale. Different disciplines like RS, GIS, and image analysis bring their own definitions of scale and the idea of computational scale makes “scale” even more confusing. For example, LUCC research can be conducted on the connectivity changes of a road network over several years (coarse granularity,

large extent, and long time span) but a navigation study may only pay attention to the routing and speed limits of the roads in the path for hours or days (fine granularity, small extent, and short time span) (Tsutsumi and Seya, 2008). Analysis of road connectivity changes requires aggregation across different granularities and extents to extract LUCC patterns. The navigation study relies more on solving the spatial optimization problem with various constraints. In the resultant visualization and evaluation, LUCC studies might present the change maps and evaluate them via ground-truthing. But the navigation study instead depicts the result as the road navigation map and needs to be compared with other routing services for evaluation. If we call all these elements (i.e., granularity, extent, and time) as 'scale', we cannot distinguish these two studies. Without a clear model of scales, it is very difficult to take advantage of big data in LUCC research.

. Big data does not only bring finer granularity and larger extents for LUCC research; but also more complicated relationship among granularity, extent, time, and corresponding properties. The new model of scale should cover spatial granularity, spatial extent, time, and property. Goodchild (2011) defined scale as granularity, extent, and time. But these three elements vary in research domains with differing topics, questions, data availability, formats, models, processing workflow, analysis algorithms, computation, and even the scientific assumptions. The concept of scale is not static but an evolving notion that continuously integrates various theories and tools (Cuzzocrea et al., 2011). Unfortunately, contextual information about data properties has not been incorporated into the scale modelling (Goodchild, Yuan, and Cova, 2007). Without such contextual information, we may treat data with different spatial granularity in the same way as data with the same spatial granularity but different properties. Some data manipulation techniques (e.g., dark object subtraction {Chavez, 1988})

may keep the granularity, extent, and time intact but changes the property. Any attempt to identify scale in LUCC without consideration of the property has been problematic.

Geospatial scaling operations (i.e., spatial granularity and extent transformation operations) can be applied to LUCC detection to homogenize scales (Celik, 2009). However, most of the spatial granularity scaling operations incurs additional noise and errors (Prashanth et al., 2009). For example, Figure 2-3 (A) and (B) are both acquired for downtown Montreal, at 2006 and 2007, respectively. Figure 2-3 (C) is the granularity scaled image of (B) using discrete wavelet transformation algorithm (Van de Wouwer, Scheunders, and Van Dyck, 1999). One can observe some noisy information at the boundaries of buildings and some change information gets lost due to the convolution of wavelet-based image scaling (e.g., the blue boxes in the red circle), from the change map (D) generated using the image differencing and percentile thresholding algorithms (Ward, 2003). Our new scale model seeks to find methods to track the impact of the geospatial scaling operations for LUCC research.

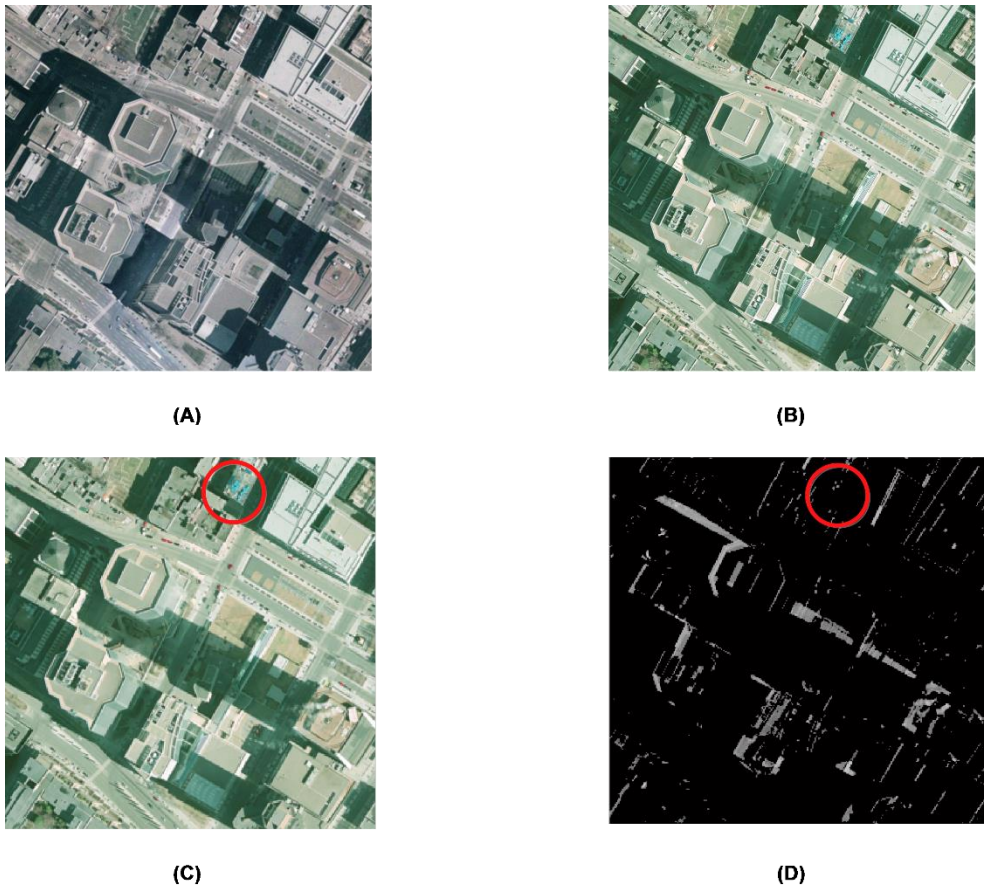


Figure 2-3. Drawbacks of Image Scaling-based LUCC. (A) is an RGB-sharpened image taken at the downtown Montreal in 2006, with 0.6m spatial resolution (DMTI Spatial Inc., 2006); (B) is a Montreal Metropolitan Community Orthophotos (MMCO) taken at the same location in 2007, with 0.3m spatial resolution (Communauté métropolitaine de Montréal, 2007); (C) the image generated using Haar discrete wavelet transformation, as 0.6m spatial resolution; and (D) the change map generated by employing the image differencing technique and the percentile thresholding.

To summarize, the new scale model needs to integrate various representation of scale theories and scaling operations. Based on semantic modelling, the new scale model enhances traceability and maneuverability of multiscale geospatial analysis, in an era of big data.

2.2.2 Workflow Challenge for LUCC

Big data is much more than a simple aggregation of small data chunks, but the distributed computing treats it as. I name this mismatch as the workflow challenge. Most RS-based LUCC

routinely requires decomposition to split big data into smaller chunks and distribute them across a large number of computing nodes for parallel processing. Researchers must compare at least two decomposed data chunks to identify LUCC and this necessitates their recomposition in the distributed computing environment. These distributed LUCC results also then need to be re-combined for generating the consolidated change map during the recomposition process. Because decomposition may distort the image features at the splitting borders, researchers also require a recomposition process that removes these specious features. Therefore, a new dataflow management framework that automates the decomposition and recomposition processes in a distributed computing environment becomes necessary to address the workflow challenge of RS-based LUCC research. Any such dataflow management framework needs to be integrated with advanced computing techniques.

The workflow challenge also reflects the importance of geogenic and topological information in big data, which are the main reasons that why geospatial big data cannot be treated as a simple aggradation of data chunks.

2.2.3 Scale Heterogenous LUCC Detection Challenge

Big data means comparing RS datasets with different scale for LUCC detection has become more frequent. Although the segmentation-based change detection method achieves high LUCC accuracy by merging spatial and spectral information, it still cannot handle scale heterogeneity and requires the employment of geospatial scaling operations to homogenize scale differences (Desclée, Bogaert, and Defourny, 2006). The spatial granularity of scaling operations may incur additional errors in LUCC, such as shown in Figure 2-3 (D). On the other hand, the spatial extent scaling may cut the image segments across data chunks and produce fake LUCC segments. Finally, temporal scaling operations always assume consistency with any original datasets and

therefore are open to criticism that they possess a high risk of missing LUCC detection (Pijanowski et al., 2002).

We need a new LUCC detection method that is invariant to scale heterogeneity and that avoids these cumbersome geospatial scaling operations. Recent progress in LUCC highlights the increasing application of local image features for change identification (Mikolajczyk and Tuytelaars, 2015) at small scales such as local neighbourhoods that are less dependent on global scene information. There is a large number of these features, including Scale Invariant Feature Transformation (SIFT) (Lowe, 2004), discrete cosine transform (Song and Li, 2013) and Speed-Up Robust Features (SURF) (Bay, Tuytelaars, and Van Gool, 2006). All these methods help LUCC detection by generating image features that are noisy (e.g., scale difference and view angle difference) resistant. Among local image feature-based image change detection study, the SIFT algorithm has received considerable interest in the field of computer vision (Mikolajczyk and Schmid, 2005). SIFT can detect, describe, and match maxima/minima points of difference of Gaussians (Burt and Adelson, 1983) across images that are invariant to scale, rotation, affine distortion, translation, and illumination differences (Liu, Yuen, and Torralba, 2011).

Algorithms like SIFT may handle the heterogeneous spatial granularities of big data but also pose other problems for change detection in LUCC. A lack of geo-registration and spatial extent specification in this process, however, increases the occurrences of the similar geographic entities (e.g., similar buildings and roads) and the misidentification of those features. When used with larger spatial extents, the comparative images may introduce far more similar geographic entities and decreases the distinctiveness of SIFT. For example, two identical buildings at different locations might be encoded with the same image feature value in SIFT. Moreover, SIFT is point-based and may not be able to cover all the change areas (Matas et al., 2004).

Consequently, it often results in the missing of LUCC identification. Therefore, we need to further enhance the scaling and handling of SIFT for LUCC study.

By integrating computer vision approaches in LUCC research, RS-based LUCC can relies on scale invariant image features for change detection, without resorting to resampling. However, the integration is not easy and considerable work is needed to customize computer vision features and feature extraction algorithms.

2.2.4 Computational Challenges for Big Data Analysis in LUCC

Big data challenge also requires the three challenges above to be handled together with the high performance computing, not separately. Because it is more complex than the simple aggregation of data chunks. Geospatial data differs from other data and therefore complicates the application of CI to LUCC. We argue that three missing methodologies prevent the employment of GCI in LUCC research. First, problems occur in data decomposition and recomposition for LUCC dataflow management within GCI. Geospatial data has inexplicit topology within the data structure, which means it is organized within a dataset. If a large dataset is split into smaller tiles then this likely slices up many objects (e.g., forests). The edges of split images can generate fake objects or object distortion. Standard recomposition is a problem because it is hard to maintain the spatial-temporal correspondence of geographic features across decomposed data and re-combine distributed intermediate results to form the final ones. Second, the deployment of LUCC workflow has not been explored within GCI. LUCC workflow presents domain specific challenges and requires special configurations (e.g., computing framework, synchronization mechanism, and GCI architecture) of GCI. Third, there needs to be a parallel computing and resource provisioning strategy for LUCC workflow in GCI. A resource provisioning strategy (e.g., computing resource allocation methods) has been explored separately for cloud

applications (Chaisiri, Lee, and Niyato, 2012) and MapReduce (Verma, Cherkasova, and Campbell, 2011), but not for a GCI yet. Due to these reasons, GCI has not been applied in LUCC and the idea of computational scale remains unexplored in LUCC. By investigating LUCC specific GCI, I tend to not only solve the computational challenge in RS-based LUCC research, but also prove the pivotal role of GCI in combining domain specific knowledge and high performance computing as a theoretical framework.

To summarize, all four challenges in LUCC involve scale. Scale modelling is about clarifying the meanings of scale and tracking the geospatial scaling operations. The workflow challenge alters the spatial extent of RS data via decomposition and recomposition operations. Consequently, a new scale invariant LUCC detection method is needed to handle scale heterogeneity across decomposed data chunks. Finally, the computational challenge with GCI relates to the scale of computation. I argue that scale becomes an essential problem in LUCC big data analysis and requires a careful investigation within each of these four challenges.

In this dissertation, I focus on the spatial-temporal scale handling of big data in LUCC by covering scale modelling methods, dataflow management framework, scale invariant LUCC detection algorithms, and LUCC-based GCIs.

I summarize the objectives of my Ph.D. research as the following:

1. Inventing a new methodology to model scale with spatial granularity, extent, time, and property to integrate and clarify various theories of scale and scaling;

2. Developing a decomposition/recomposition framework for LUCC dataflow management in the distributed computing environment and prove the essential role of geometric and topological information in geospatial big data analysis;

3. Investigating new scale invariant change detection algorithms that identify LUCC by comparing the scale invariant image features and do not rely on the resampling methods to homogenize the various spatial-temporal scales, which also serve as a new bridge between computer vision and RS-based LUCC research;

4. Developing a LUCC specific GCI with scalable computing resource provisioning and distributed computing support to integrate LUCC workflow, spatial-temporal models, and the change boundary optimization algorithm as a comprehensive solution for big data analysis in LUCC, in which GCI is explored as a scientific framework to integrate domain specific knowledge and high performance computing.

In the following chapters, I will present approaches for each challenge in the list. My methodologies are mainly evaluated by conducting urban-rural LUCC detection studies in the Greater Montreal Area from 2005-2012 using scale heterogeneous datasets acquired from various sensing platforms.

My dissertation builds multiple bridges between GIScience and RS research. The new scale modelling work integrates different concepts of scale and scaling in both GIScience and RS. The decomposition/recomposition dataflow management framework is developed as a solution for RS image splitting, but it also serves as a general geospatial big data handling method. The scale invariant LUCC detection algorithm is designed with computer vision algorithms for RS image analysis, but it also integrates the concept of spatial variance and location referencing in GIScience. The LUCC-GCI is based on the GCI methodologies in GIScience, but it also includes considerable domain knowledge from RS, such as image

read/write, noise removal, and geo-referencing. In conclusion, the scale challenge in big data LUCC study requires a tight combination of GIScience and RS research.

2.3 Reference

Alpers, C. N., & Brimhall, G. H. (1988). Middle Miocene climatic change in the Atacama Desert, northern Chile: Evidence from supergene mineralization at La Escondida. Geological Society of America Bulletin, 100(10), 1640-1656.

Alvanides, S., Openshaw, S., & Macgill, J. (2001). Zone design as a spatial analysis tool. Modelling Scale in Geographical Information Science, London, 141-157.

Amazon Elastic Cloud Computing (EC2) (2017). Online: http://aws.amazon.com/ec2/

Armstrong, M. P., Nyerges, T. L., Wang, S., & Wright, D. (2011). Connecting geospatial information to society through cyberinfrastructure. The SAGE Handbook of GIS and Society. London, Sage Publications, 109-22.

Atkins, D. (2003). Revolutionizing science and engineering through cyberinfrastructure: Report of the National Science Foundation blue-ribbon advisory panel on cyberinfrastructure.

Atkinson, P. M. (2001). Geostatistical regularization in remote sensing. Modelling Scale in Geographical Information Science, 237-260.

Bay, H., Tuytelaars, T., & Van Gool, L. (2006). Surf: Speeded up robust features. In European conference on computer vision (pp. 404-417). Springer Berlin Heidelberg.

Benz, U. C., Hofmann, P., Willhauck, G., Lingenfelder, I., & Heynen, M. (2004). Multi-resolution, object-oriented fuzzy analysis of remote sensing data for GIS-ready information. ISPRS Journal of photogrammetry and remote sensing, 58(3), 239-258.

Bianchin, A., & Bravin, L. (2008). Remote sensing and urban analysis. In International Conference on Computational Science and Its Applications (pp. 300-315). Springer Berlin Heidelberg.

Blaschke, T. (2010). Object based image analysis for remote sensing. ISPRS Journal of Photogrammetry and Remote Sensing, 65(1), 2-16.

Brenner, N. (1998). Global cities, glocal states: global city formation and state territorial restructuring in contemporary Europe. Review of International Political Economy, 5(1), 1-37.

Buetow, K. H. (2005). Cyberinfrastructure: empowering a" third way" in biomedical research. Science, 308(5723), 821-824.

Burt, P., & Adelson, E. (1983). The Laplacian pyramid as a compact image code. IEEE Transactions on communications, 31(4), 532-540.

Byrne, G. F., Crapper, P. F., & Mayo, K. K. (1980). Monitoring land-cover change by principal component analysis of multitemporal Landsat data. Remote Sensing of Environment, 10(3), 175-184.

Cash, D. W., Adger, W. N., Berkes, F., Garden, P., Lebel, L., Olsson, P., ... & Young, O. (2006). Scale and cross-scale dynamics: governance and information in a multilevel world. Ecology and society, 11(2), 8.

Celik, T. (2009). Unsupervised change detection in satellite images using principal component analysis and $ k $-means clustering. IEEE Geoscience and Remote Sensing Letters, 6(4), 772-776.

Chaisiri, S., Lee, B. S., & Niyato, D. (2012). Optimization of resource provisioning cost in cloud computing. IEEE Transactions on Services Computing, 5(2), 164-177.

Chavez, P. S. (1988). An improved dark-object subtraction technique for atmospheric scattering correction of multispectral data. Remote sensing of environment, 24(3), 459-479.

Chen, G., Hay, G. J., Carvalho, L. M., & Wulder, M. A. (2012). Object-based change detection. International Journal of Remote Sensing, 33(14), 4434-4457.

Chen, J., Gong, P., He, C., Pu, R., & Shi, P. (2003). Land-use/land-cover change detection using improved change-vector analysis. Photogrammetric Engineering & Remote Sensing, 69(4), 369-379.

Communauté métropolitaine de Montréal, 2005, 2007. Montreal Metropolitan Community Orthophotos / Orthophotographies de la Communauté métropolitaine de Montréal, 2005 and 2007.

Congalton, R. G. (1991). A review of assessing the accuracy of classifications of remotely sensed data. Remote sensing of environment, 37(1), 35-46.

Coppin, P. R., & Bauer, M. E. (1996). Digital change detection in forest ecosystems with remote sensing imagery. Remote sensing reviews, 13(3-4), 207-234.

Cummings, J., Finholt, T., Foster, I., Kesselman, C., & Lawrence, K. A. (2008). Beyond being there: A blueprint for advancing the design, development, and evaluation of virtual organizations.

Cuzzocrea, A., Song, I. Y., & Davis, K. C. (2011). Analytics over large-scale multidimensional data: the big data revolution!. In Proceedings of the ACM 14th international workshop on Data Warehousing and OLAP (pp. 101-104). ACM.

Dai, F. C., Lee, C. F., & Zhang, X. H. (2001). GIS-based geo-environmental evaluation for urban land-use planning: a case study. Engineering geology,61(4), 257-271.

Dai, X., & Khorram, S. (1998). The effects of image misregistration on the accuracy of remotely sensed change detection. IEEE Transactions on Geoscience and Remote sensing, 36(5), 1566-1577.

Dean, J., & Ghemawat, S. (2008). MapReduce: simplified data processing on large clusters. Communications of the ACM, 51(1), 107-113.

Desclée, B., Bogaert, P., & Defourny, P. (2006). Forest change detection by statistical object-based method. Remote Sensing of Environment, 102(1), 1-11.

DMTI, 2006, 2009, 2012. Montreal Satellite StreetView, 3_1- 9_7_MONTREAL-S3XM, Markham ON: DMTI Spatial Inc., 2006, 2009 and 2012.

Droegemeier, K. K., Chandrasekar, V., Clark, R., Gannon, D., Graves, S., Joseph, E., ... & Yalda, S. (2004). Linked environments for atmospheric discovery (LEAD): A cyberinfrastructure for mesoscale meteorology research and education. In 20th Conf. on Interactive Information Processing Systems for Meteorology, Oceanography, and Hydrology.

Dueker, K. J., & Horton, F. E. (1972). Urban-change detection systems: Remote-sensing inputs. Photogrammetria, 28(3), 89-106.

Emerson, C. W. (1998). Multi-scale fractal analysis of image texture and pattern.

Erwin, K. L. (2009). Wetlands and global climate change: the role of wetland restoration in a changing world. Wetlands Ecology and management, 17(1), 71.

Flowerdew, R., Geddes, A., & Green, M. (2001). Behaviour of regression models under random aggregation. Modelling scale in geographical information science, 89-104.

Flowerdew, R., & Green, M. (1993). Developments in areal interpolation methods and GIS. In Geographic Information Systems, Spatial Modelling and Policy Evaluation (pp. 73-84). Springer Berlin Heidelberg.

Foley, J. A., DeFries, R., Asner, G. P., Barford, C., Bonan, G., Carpenter, S. R., ... & Helkowski, J. H. (2005). Global consequences of land use. Science, 309(5734), 570-574.

Frey, J., Tannenbaum, T., Livny, M., Foster, I., & Tuecke, S. (2002). Condor-G: A computation management agent for multi-institutional grids. Cluster Computing, 5(3), 237-246.

Gamon, J. A., Coburn, C., Flanagan, L. B., Huemmrich, K. F., Kiddle, C., Sanchez-Azofeifa, G. A., ... & Rahman, A. F. (2010). SpecNet revisited: bridging flux and remote sensing communities. Canadian Journal of Remote Sensing, 36(S2), S376-S390.

Gao, S., Li, L., Li, W., Janowicz, K., & Zhang, Y. (2014). Constructing gazetteers from volunteered big geo-data based on Hadoop. Computers, Environment and Urban Systems, doi:10.1016, in press.

Goldewijk, K. K. (2001). Estimating global land use change over the past 300 years: the HYDE database. Global Biogeochemical Cycles, 15(2), 417-433.

Gong, J., Sui, H., Sun, K., Ma, G., & Liu, J. (2008). Object-level change detection based on full-scale image segmentation and its application to Wenchuan Earthquake. Science in China Series E: Technological Sciences, 51, 110-122.

Goodchild, M. F. (2001). Models of scale and scales of modelling. Modelling scale in geographical information science, 3-10.

Goodchild, M. F. (2010). Twenty years of progress: GIScience in 2010.Journal of spatial information science, 2010(1), 3-20.

Goodchild, M. F. (2011). Scale in GIS: An overview. Geomorphology, 130(1), 5-9.

Goodchild, M. F., Yuan, M., & Cova, T. J. (2007). Towards a general theory of geographic representation in GIS. International journal of geographical information science, 21(3), 239-260.

Hardie, I. W., & Parks, P. J. (1997). Land use with heterogeneous land quality: an application of an area base model. American Journal of Agricultural Economics, 79(2), 299-310.

Hecheltjen, A., Thonfeld, F., & Menz, G. (2014). Recent advances in remote sensing change detection–a review. In Land Use and Land Cover Mapping in Europe (pp. 145-178). Springer Netherlands.

Henderson, F. M., & Lewis, A. J. (1998). Principles and applications of imaging radar. Manual of remote sensing: Volume 2.

Hey, T., & Trefethen, A. E. (2005). Cyberinfrastructure for e-Science.Science, 308(5723), 817-821.

İlsever, M., & Ünsalan, C. (2012). Texture Analysis Based Change Detection Methods. In Two-Dimensional Change Detection Methods (pp. 35-39). Springer London.

Kaufman, Y. J., Tanré, D., Gordon, H. R., Nakajima, T., Lenoble, J., Frouin, R., ... & Teillet, P. M. (1997). Passive remote sensing of tropospheric aerosol and atmospheric correction for the aerosol effect. Journal of Geophysical Research: Atmospheres, 102(D14), 16815-16830.

Kauth, R. J., & Thomas, G. S. (1976). The tasselled cap--a graphic description of the spectral-temporal development of agricultural crops as seen by Landsat. In LARS Symposia (p. 159).

Kim, S., & Heller, M. (2006). Emerging cyberinfrastructure: challenges for the chemical process control community. Computers & chemical engineering, 30(10), 1497-1501.

Kitchin, R. (2013). Big data and human geography Opportunities, challenges and risks. Dialogues in human geography, 3(3), 262-267.

Krause, C., Tichy, M., & Giese, H. (2014). Implementing graph transformations in the bulk synchronous parallel model. In International Conference on Fundamental Approaches to Software Engineering (pp. 325-339). Springer Berlin Heidelberg.

Lal, R. (2004). Soil carbon sequestration impacts on global climate change and food security. science, 304(5677), 1623-1627.

Lambin, E. F., & Meyfroidt, P. (2011). Global land use change, economic globalization, and the looming land scarcity. Proceedings of the National Academy of Sciences, 108(9), 3465-3472.

Lee, E. A. (2008). Cyber physical systems: Design challenges. InObject Oriented Real-Time Distributed Computing (ISORC), 2008 11th IEEE International Symposium on (pp. 363-369). IEEE.

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(4), 11-20.

Lee, S. (2005). Application of logistic regression model and its validation for landslide susceptibility mapping using GIS and remote sensing data.International Journal of Remote Sensing, 26(7), 1477-1491.

Legendre, P., & Fortin, M. J. (1989). Spatial pattern and ecological analysis. Vegetatio, 80(2), 107-138.

Li, W., Wu, S., Song, M., & Zhou, X. (2016). A scalable cyberinfrastructure solution to support big data management and multivariate visualization of time-series sensor observation data. Earth Science Informatics, 9(4), 449-464.

Li, X., & Yeh, A. G. O. (2002). Neural-network-based cellular automata for simulating multiple land use changes using GIS. International Journal of Geographical Information Science, 16(4), 323-343.

Li, X., & Yeh, A. G. O. (2004). Analyzing spatial restructuring of land use patterns in a fast growing region using remote sensing and GIS. Landscape and Urban planning, 69(4), 335-354.

Li, X., Li, W., Anselin, L., Rey, S., & Koschinsky, J. (2014). A MapReduce algorithm to create contiguity weights for spatial analysis of big data. In Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Analytics for Big Geospatial Data (pp. 50-53). ACM.

Li, Y., & Davis, C. H. (2008). Unsupervised change detection in high resolution satellite imagery from fusion of spectral and spatial information. In Geoscience and Remote Sensing Symposium, 2008. IGARSS 2008. IEEE International (Vol. 2, pp. II-109). IEEE.

Li, Z., Yang, C., Liu, K., Hu, F., & Jin, B. (2016). Automatic Scaling Hadoop in the Cloud for Efficient Process of Big Geospatial Data. ISPRS International Journal of Geo-Information, 5(10), 173.

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 (Vol. 6292). Zurich, Switzerland: Springer.

Liang, W., Hoja, D., Schmitt, M., & Stilla, U. (2011). Comparative study of change detection for reconstruction monitoring based on very high resolution optical data. In Urban Remote Sensing Event (JURSE), 2011 Joint(pp. 73-76). IEEE.

Liu, C., Yuen, J., & Torralba, A. (2011). Sift flow: Dense correspondence across scenes and its applications. IEEE transactions on pattern analysis and machine intelligence, 33(5), 978-994.

Liu, S., Du, P., Gamba, P., & Xia, J. (2011). Fusion of difference images for change detection in urban areas. In Urban Remote Sensing Event (JURSE), 2011 Joint (pp. 165-168). IEEE.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. International journal of computer vision, 60(2), 91-110.

Lu, D., Mausel, P., Brondizio, E., & Moran, E. (2004). Change detection techniques. International journal of remote sensing, 25(12), 2365-2401.

Lunetta, R. S., Knight, J. F., Ediriwickrema, J., Lyon, J. G., & Worthy, L. D. (2006). Land-cover change detection using multi-temporal MODIS NDVI data. Remote sensing of environment, 105(2), 142-154.

Ma, Y., Wu, H., Wang, L., Huang, B., Ranjan, R., Zomaya, A., & Jie, W. (2015). Remote sensing big data computing: Challenges and opportunities.Future Generation Computer Systems, 51, 47-60.

Maini, R., & Aggarwal, H. (2009). Study and comparison of various image edge detection techniques. International journal of image processing (IJIP),3(1), 1-11.

Malczewski, J. (2004). GIS-based land-use suitability analysis: a critical overview. Progress in planning, 62(1), 3-65.

Marceau, D. J., & Hay, G. J. (1999). Remote sensing contributions to the scale issue. Canadian journal of remote sensing, 25(4), 357-366.

Marston, S. A., Jones, J. P., & Woodward, K. (2005). Human geography without scale. Transactions of the Institute of British Geographers, 30(4), 416-432.

Matas, J., Chum, O., Urban, M., & Pajdla, T. (2004). Robust wide-baseline stereo from maximally stable extremal regions. Image and Vision Computing 22(10), 761-767.

Mikolajczyk, K., & Schmid, C. (2005). A performance evaluation of local descriptors. IEEE transactions on pattern analysis and machine intelligence,27(10), 1615-1630.

Mikolajczyk, K., & Tuytelaars, T. (2015). Local image features. Encyclopedia of Biometrics, 1100-1105.

Morisette, J. T., & Khorram, S. (2000). Accuracy assessment curves for satellite-based change detection. Photogrammetric Engineering and Remote Sensing, 66(7), 875-880.

Mucher, C. A., K. T. Steinnocher, F. P. Kressler, and C. Heunks. (2000). "Land Cover Characterization and Change Detection for Environmental Monitoring of pan-Europe." International Journal of Remote Sensing 21 (6-7): 1159–1181. doi:10.1080/014311600210128.

National Science Foundation, (2005). NSF's Cyberinfrastructure Vision for 21st Century Discovery, NSF Cyberinfrastructure Council, September 26th, 2005, Ver.4.0, pg 4.

Nurain, N., Sarwar, H., Sajjad, M. P., & Mostakim, M. (2012). An In-depth Study of Map Reduce in Cloud Environment. In Advanced Computer Science Applications and Technologies (ACSAT), 2012 International Conference on (pp. 263-268). IEEE.

Pal, N. R., & Pal, S. K. (1993). A review on image segmentation techniques. Pattern recognition, 26(9), 1277-1294.

Parker, D. C., Manson, S. M., Janssen, M. A., Hoffmann, M. J., & Deadman, P. (2003). Multi-agent systems for the simulation of land-use and land-cover change: a review. Annals of the association of American Geographers, 93(2), 314-337.

Pijanowski, B. C., Brown, D. G., Shellito, B. A., & Manik, G. A. (2002). Using neural networks and GIS to forecast land use changes: a land transformation model. Computers, environment and urban systems, 26(6), 553-575.

Ping Tian, D. (2013). A review on image feature extraction and representation techniques. International Journal of Multimedia and Ubiquitous Engineering,8(4), 385-396.

Plale, B., Gannon, D., Brotzge, J., Droegemeier, K., Kurose, J., McLaughlin, D., ... & Yalda, S. (2006). Casa and lead: Adaptive cyberinfrastructure for real-time multiscale weather forecasting. Computer, 39(11).

Plaza, A., Du, Q., Chang, Y. L., & King, R. L. (2011). High performance computing for hyperspectral remote sensing. Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal of, 4(3), 528-544.

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(5), 823-854.

Prashanth, H. S., Shashidhara, H. L., & KN, B. M. (2009). Image scaling comparison using universal image quality index. In Advances in Computing, Control, & Telecommunication Technologies, 2009. ACT'09. International Conference on (pp. 859-863). IEEE.

Qin, C. Z., Zhan, L. J., & Zhu, A. (2014). How to apply the Geospatial Data Abstraction Library (GDAL) properly to parallel geospatial raster I/O?.Transactions in GIS, 18(6), 950-957.

Radke, R. J., Andra, S., Al-Kofahi, O., & Roysam, B. (2005). Image change detection algorithms: a systematic survey. Image Processing, IEEE Transactions on, 14(3), 294-307.

Real, C. (2008). Making research and education cyberinfrastructure real. Educause Review, 43(4).

Rosin, P. L., & Ioannidis, E. (2003). Evaluation of global image thresholding for change detection. Pattern Recognition Letters, 24(14), 2345-2356.

Schnase, J. L., Lee, T. J., Mattmann, C. A., Lynnes, C. S., Cinquini, L., Ramirez, P. M., ... & Webster, W. P. (2016). Big Data Challenges in Climate Science: Improving the next-generation cyberinfrastructure. IEEE Geoscience and Remote Sensing Magazine, 4(3), 10-22.

Schowengerdt, R. A. (2006). Remote sensing: models and methods for image processing. Academic press, pp.22-31.

Short, N. M. (1982). The Landsat Tutorial Workbook, NASA RP-1078.

Shvachko, K., Kuang, H., Radia, S., & Chansler, R. (2010). The hadoop distributed file system. In Mass storage systems and technologies (MSST), 2010 IEEE 26th symposium on (pp. 1-10). IEEE.

Sieber, R. E., Wellen, C. C., & Jin, Y. (2011). Spatial cyberinfrastructures, ontologies, and the humanities. Proceedings of the National Academy of Sciences, 108(14), 5504-5509.

Singh, A. (1989). Review article digital change detection techniques using remotely-sensed data. International journal of remote sensing, 10(6), 989-1003.

Song, T., & Li, H. (2013). Local polar DCT features for image description.IEEE Signal Processing Letters, 20(1), 59-62.

Stauffer, M. L., & McKinney, R. L. (1978). Landsat image differencing as an automated land cover change detection technique. Interim report, NAS5-24350.

Stefanov, W. L., & Netzband, M. (2005). Assessment of ASTER land cover and MODIS NDVI data at multiple scales for ecological characterization of an arid urban center. Remote sensing of Environment, 99(1), 31-43.

Stewart, C. (2015). Cyberinfrastructure for Research: New Trends and Tools (Part 1 of 2). Presented at University of Vermont.

Stewart, C. A., Simms, S., Plale, B., Link, M., Hancock, D. Y., & Fox, G. C. (2010). What is cyberinfrastructure. In Proceedings of the 38th annual ACM SIGUCCS fall conference: navigation and discovery (pp. 37-44). ACM.

Sun, Q., Chi, T., Wang, X., & Zhong, D. (2005). Design of middleware based grid GIS. In Geoscience and Remote Sensing Symposium, 2005. IGARSS'05. Proceedings. 2005 IEEE International (Vol. 2, pp. 4-pp). IEEE.

Sun, Z., Chen, F., Chi, M., & Zhu, Y. (2015). A spark-based big data platform for massive remote sensing data processing. In International Conference on Data Science (pp. 120-126). Springer International Publishing.

Teillet, P. M. (1986). Image correction for radiometric effects in remote sensing. International Journal of Remote Sensing, 7(12), 1637-1651.

Toure, S., Stow, D., Shih, H. C., Coulter, L., Weeks, J., Engstrom, R., & Sandborn, A. (2016). An object-based temporal inversion approach to urban land use change analysis. Remote Sensing Letters, 7(5), 503-512.

Toutin, T. (2004). Review article: Geometric processing of remote sensing images: models, algorithms and methods. International journal of remote sensing, 25(10), 1893-1924.

Townshend, J. R., Justice, C. O., Gurney, C., & McManus, J. (1992). The impact of misregistration on change detection. IEEE Transactions on Geoscience and remote sensing, 30(5), 1054-1060.

Tsutsumi, M., & Seya, H. (2008). Measuring the impact of large-scale transportation projects on land price using spatial statistical models. Papers in Regional Science, 87(3), 385-401.

Turner, B. L., Meyer, W. B., & Skole, D. L. (1994). Global land-use/land-cover change: towards an integrated study. Ambio. Stockholm, 23(1), 91-95.

Van de Wouwer, G., Scheunders, P., & Van Dyck, D. (1999). Statistical texture characterization from discrete wavelet representations. IEEE transactions on image processing, 8(4), 592-598.

Veldkamp, A., & Lambin, E. F. (2001). Predicting land-use change. Agriculture, ecosystems & environment, 85(1), 1-6.

Verma, A., Cherkasova, L., & Campbell, R. H. (2011). Resource provisioning framework for mapreduce jobs with performance goals. InACM/IFIP/USENIX International Conference on Distributed Systems Platforms and Open Distributed Processing (pp. 165-186). Springer Berlin Heidelberg.

Vörösmarty, C. J., Green, P., Salisbury, J., & Lammers, R. B. (2000). Global water resources: vulnerability from climate change and population growth.science, 289(5477), 284-288.

Vu, T. T., Matsuoka, M., & Yamazaki, F. (2004). LIDAR-based change detection of buildings in dense urban areas. In Geoscience and Remote Sensing Symposium, 2004. IGARSS'04. Proceedings. 2004 IEEE International (Vol. 5, pp. 3413-3416). IEEE.

Waddell, P. (2002). UrbanSim: Modeling urban development for land use, transportation, and environmental planning. Journal of the American planning association, 68(3), 297-314.

Ward, G. (2003). Fast, robust image registration for compositing high dynamic range photographs from hand-held exposures. Journal of graphics tools, 8(2), 17-30.

Weng, Q. (2002). Land use change analysis in the Zhujiang Delta of China using satellite remote sensing, GIS and stochastic modelling. Journal of environmental management, 64(3), 273-284.

Wright, D. J., & Wang, S. (2011). The emergence of spatial cyberinfrastructure. Proceedings of the National Academy of Sciences, 108(14), 5488-5491.

Wu, H., & Li, Z. L. (2009). Scale issues in remote sensing: A review on analysis, processing and modeling. Sensors, 9(3), 1768-1793.

Xiang, H., & Tian, L. (2011). Method for automatic georeferencing aerial remote sensing (RS) images from an unmanned aerial vehicle (UAV) platform. Biosystems Engineering, 108(2), 104-113.

Xiubin, L. (1996). A review of the international researches on land use/land cover change [J]. Acta Geographica Sinica, 6.

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

Yang, C., Huang, Q., Li, Z., Liu, K., & Hu, F. (2017). Big Data and cloud computing: innovation opportunities and challenges. International Journal of Digital Earth, 10(1), 13-53.

Yang, C., Raskin, R., Goodchild, M., & Gahegan, M. (2010). Geospatial cyberinfrastructure: past, present and future. Computers, Environment and Urban Systems, 34(4), 264-277.

Yu, B. H., & Lü, C. H. (2008). Spatio-temporal characteristics and driving factors of farmland change on urban fringe: A case study of Shunyi District, Beijing Municipality. Scientia Geographica Sinica, 3, 009.

Yuan, F., Sawaya, K. E., Loeffelholz, B. C., & Bauer, M. E. (2005). Land cover classification and change analysis of the Twin Cities (Minnesota) Metropolitan Area by multitemporal Landsat remote sensing. Remote sensing of Environment, 98(2), 317-328.

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(5), 605-630.

Zubin, D. (1989). Oral presentation, NCGIA Initiative 2 Specialist Meeting, Santa Barbara. Reported in D. Mark (ed.): Languages of Spatial Relations: Researchable Questions & NCGIA Research Agenda, NCGIA Report 89-2A, NCGIA.

Connecting Statement: Addressing the Scale Challenge in LUCC with the Concept of Scope

In Chapter 2, I highlighted the essential role of scale in big data analysis for LUCC research. However, the definition of "scale" varies and the variation can affect how the concept is used in LUCC. For example, multi-scale data can refer to data with different spatial resolutions, as well as data with different spatial coverages. This confusion over meaning is made worse with big data, which adds many new resolutions, extents, spectra, and time periods. To clarify the meaning of scale in LUCC, I propose the concept of Scope. I define Scope as a function of spatial granularity, spatial extent, the time, and properties.

Chapter 3 will be submitted to the *International Journal of Geographical Information Science*. The manuscript contained in this chapter was co-authored with my supervisor, Prof. Renée Sieber, and members of my doctoral supervisory committee including Prof. Shaowen Wang. I was the primary author and contributed the realization of Scope, through its theoretical framework and its implementation in the case study. Prof. Sieber refined the Scope concept. Prof. Wang gave me guidance to develop the Scope Set and Scope quadruple projection concept, based on his spatial computational domain representation (Wang and Armstrong, 2009). Both co-authors edited this chapter for readability.

Chapter 3. A New Scale Representation for Multiscale Geospatial Analysis

Abstract

Geospatial big data is noted for variety and the opportunities and challenges it creates for multiscale analysis. One of the main challenges in comparing big geospatial datasets often is their very different spatial granularities or temporal periodicities. We are advantaged by the ability to derive, for example through the geospatial scaling operations (i.e., spatial up/down-sampling and extent decomposition/recomposition), many different datasets from the original big data. To handle complex meanings of scale, we propose the concept of Scope to model scale in geospatial big data as a quadruple that integrates spatial granularities, spatial extents, time, and properties (attributes). We develop the concept of Scope Set as the collection of related quadruples and Scope quadruple projection to model the scale transformations. Case studies illustrate how to use Scope to measure the effect of the geospatial scaling operations. Our findings indicate that Scope can represent the complex meanings and changes of the scale needed to compare big geospatial datasets and will become a fundamental part of multiscale analysis.

3.1 Introduction

Geospatial big data has caused a shift in Geographic Information Science (GIScience) research toward being more data-driven (Miller and Goodchild, 2015). This shift calls into question the ways in which GIScientists handle scale in their analysis. Scale heterogeneity in geospatial big data can be interpreted as an instance of "variety," since there are so many new geospatial data sources, sensors, and platforms that each collect at different spatial and temporal scales (Clarke and Gaydos, 1998; Clarke, 2003). If it is raster, the data often might be available from new Remote Sensing (RS) platforms at different electromagnetic (spectral) wavelengths. If it is

vector, we will instead face heterogenous properties, data structures, encodings, geo-referencing and even file formats (Eastman, 2001).

With the growing data volume, velocity, and variety, it is increasingly challenging to guarantee scale matches among geospatial big data and analysis algorithms. Many platforms provide high spatial resolution data that can present problems for comparison with legacy data at coarser resolutions. Usually, we employ spatial scaling operations (i.e., spatial up/down-sampling and extent decomposition/recomposition) to homogenize scale across datasets. For example, we down-scale Landsat images to 1m resolution in order to compare it with IKONOS imagery datasets. Heterogeneity of scales is problematic for some geospatial analysis algorithms like flow fields (Heeger, 1987) that, if applied, would generate numerous fake flow vectors because the algorithm cannot establish the correspondence of the same feature across datasets. No efficient scale modelling method currently exists that can cover these two geospatial scale mismatch problems.

Multiscale analysis has been further complicated by the numerous meanings of the term "scale". Woodcock and Strahler (1987) referred to "scale" as the different resolutions of raster datasets. In contrast, Inglada and Mercier (2007) defined "scale" as image clusters that covered various areas of land surface with the clusters generated based on spatial granularity scaling techniques (Atkins, Bouman, and Allebach, 2001). Ouyang et al. (2014) considered administrative district levels as scales, while Goodchild (2011) defined scale as both geospatial granularity (i.e., level of details) and extent (i.e., spatial-temporal coverage of the study area). Goodchild's definition also happens to solves one problem faced by researchers who regularly misunderstand the difference between the two (Schuurman et al., 2006). Wu and Li (2009) summarized the meanings of scale with respect to different research domains and geospatial

analysis algorithms. They found that scale manifested at all stages in data handling including observation, modelling (representation), analysis, and visualization. Similarly, Marceau and Hay (1999) focused on the meanings of scale in Remote Sensing (RS) granularities (resolutions) and emphasized spatial granularity scaling operations (e.g., the change of detail levels in an image from 1m to 30m).

Spatial-temporal scaling operations have further complicated the process of scale modelling. We apply diverse geospatial scaling operations to avoid the failure caused by scale mismatches (Spaccapietra, Parent, and Vangenot, 2000). Examples included Celik (2009) who conducted a Land Use/Cover Change (LUCC) study based on discrete wavelet-based up/down-sampling technique. They also included Aspinall (2001) who presented multiscale Bayesian models for the distribution of red squirrels, with geospatial granularities ranging from 1km to 20km in Scotland. Finally, this mismatch also affected a quadtree-based extent decomposition method that was proposed by Wang and Armstrong (2003) to tackle distributed geospatial interpolation problems on grid computing platforms. Multiscale research such as these studies often addressed either spatial granularity or the extent changes but not both. That was because they assumed one of the variables would remain constant. In the era of big data, however, simultaneous granularity and extent changes have been becoming more frequent (e.g., simultaneous localization and mapping {Floros et al., 2013}) and creating new challenges in scale modelling.

To address the issue of multiscale analysis in big data, we propose the concept of Scope to integrate granularity and extent with time and property. Time is fundamental in describing geographic processes and always must be considered as an independent element in scale modelling (Stommel, 1963). For example, the optical reflectance of tree leaves continuously

change at different times each day but analysis must not treat such daily changes as valid forest changes. The re-pavement of roads can happen in several days, but the building of a new highway will take years. Properties (i.e., the structures, attributes, and quality of the dataset) are tightly correlated to our understanding of those phenomena as well as geospatial analysis algorithms (Câmara et al., 2000; Cova and Goodchild, 2002; Jordan et al., 1998). Most geospatial analysis algorithms are property-specific (De Smith, Goodchild, and Longley, 2009). Properties provide the context in which granularity, extent, and time need to be specified.

By merging the generic field data model (Camara et al., 2014) and scale space theories (Witkin, 1984), we propose Scope to represent geospatial data with spatial granularity, spatial extent, time, and property, as a Scope quadruple. The generic field model is chosen because it explicitly considers granularities and extents (Goodchild, 2011). Based on scale space, Scope provide the formal means to organize numerous Scope quadruples as an *algebraic set* (Frank, 1999), called Scope Set. We argue that Scope can more efficiently represent scale in geospatial dataset and analysis algorithms, for both RS and Geographic Information System (GIS) related research.

The rest of this paper is organized as follows. We discuss current research on scale and our motivation for proposing Scope in Section 3.2. Then we introduce the concept of Scope in Section 3.3. Section 3.4 suggests how Scope could be employed for multi-granularity/multi-extent road classification study, using data entropy as the property. Section 3.5 concludes this paper with future work that applies an *Abelian Group* for scale modelling.

3.2 Challenges of Scale Modelling

There is a large body of challenges we are facing in scale modelling. First, it is necessary to clarify the complex meanings of scale. Second, scale in GIS object and field view need to be

synthesized. Third, the scale mismatch necessitates the tracking of the scale transformations. At last, the temporal scale requires a special investigation. The following discussion delineates these challenges sequentially.

Various meanings of scale are playing pivotal roles in multiscale analysis. Vlachos (2005) reviewed discrete and particle models for multiscale analysis mainly from examples in biochemistry and environment research. His work found that scale represents a large body of different concepts that include grain, area, length, grid, number and the distribution of computing resource. Wu et al. (2000) categorized multiscale analysis in GIS into direct and indirect approaches that considered the grain sizes and extents of a given dataset. Celik (2009) used multiscale analysis to detect Land User/Cover Change (LUCC) from RS images but his conception of scale only encompassed granularity. Atkinson and Tate (2000) reviewed the scale problem in multiscale analysis as a geo-statistics problem (i.e., variogram) as a special measurement of spatial variance. Unfortunately, these researchers do not agree in their definitions of scale. The inconclusiveness previous work requires the development of a new scale model to clarify the meanings of scale and represent the changes of scale in the multiscale analysis.

The fact that scale is defined by various meanings across GIS/RS research can seriously hinder the ability to conduct multiscale analysis that takes full advantage of today's big data. To do so, multiscale analysis in GIS/RS must be able to handle big data with different spatial granularities (Celik, 2009), differing spatial extents (Benz et al., 2004), and varying time spans (Coppin et al., 2004).

The new scale model needs to incorporate different geographic data models. Data models in GIS/RS stand for a large body of mathematical approaches of representing geographic

properties (Cash et al., 2006). Data models can be the object or field model in GIScience (Couclelis, 1992), in which the properties within the dataset serve as the indicators for the structure and quality of the datasets with respect to different research questions (Câmara et al., 1996). Thus, synthesizing the object and field models becomes another indispensable element in the new scale model.

To approximate the geographic world, object data models offer different ways to define points, lines, polygons, and volumes. The buildings in Figure 3-1 (A) might be labelled either as points (e.g., centroids) or polygons, depending on whether we need to consider the shape of buildings. Buildings can be further aggregated as "blocks," to be considered as super-objects of buildings. If the research question is to analyze district or community boundaries at the city extent, blocks may offer a more appropriate granularity than individual buildings. Although individual buildings might be renovated or discarded at different times, their changes over time have limited impact on the city due to the limited granularities. However, the object model represents granularity, extents, and time, in an inexplicit way by various objects (Goodchild, 2011) and these elements may be heterogeneous even within the same piece of data. Scale model works at a higher abstraction level than the object data model and needs to explicitly indicate the granularity, extents, and time (Store and Jokimäki, 2003).

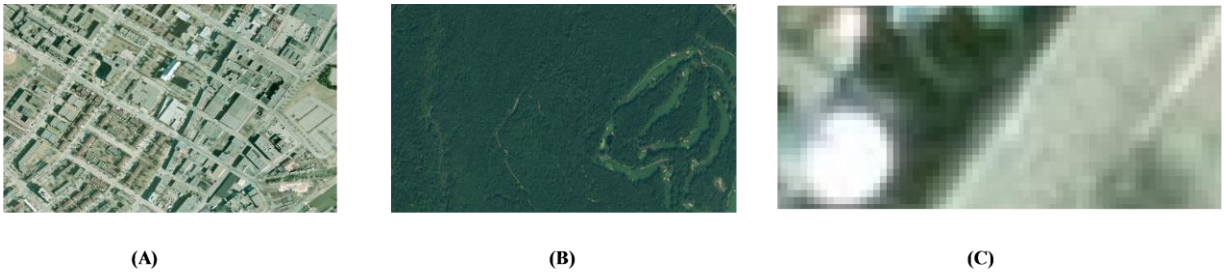


Figure 3-1. Examples of Spatial Granularity and Extent Variety in Scope. (A) 0.11m Montreal Metropolitan Community Orthophotos (MMCO) image acquired at downtown Montreal, 2005 (Communauté métropolitaine de Montréal, 2005); (B) 0.6m DMTI image collected at Montreal, 2006 (DMTI Spatial Inc., 2006); (C) ragged building boundaries obtained by up-sampling some buildings in (A) into 1.76 m.

The granularities and extents in the field data model are largely defined by the sampling schema (Couclelis, 1992; Goodchild, 2011). Conceptually, a field model provides infinitely fine granularities. In practice, researchers are limited by practical matters such as how the distance between the sampling points and the sensing platforms determine the level of granularity. Sampling ranges or regions of interest regulates the extent. Often geospatial scaling operations are employed as the changes of sampling distance and ranges (Celik, 2009; Pohl and Van Genderen, 1998). Researchers have proposed the concept of scale space (Perona and Malik, 1990), which is a collection of raster data up/down-sampled at various spatial granularities, to allow for multi-granularity analysis. However, geospatial scaling operations may incur artifacts and additional errors into the original field data, as illustrated by Prashanth et al. (2009). For example, the building boundaries in Figure 3-1 (A) that are acquired by the 0.11m granularity will become more ragged as shown in Figure 3-1 (C) at 1.76m with granularity scaling.

The new scale model needs to cover both object and field data models by merging of these two models (Goodchild, Yuan, and Cova, 2007). Liu et al. (2008) bridged these two models by embedding objects into fields. Their approach originated from the ‘plenum’ model of physics, and the field model provided the ontological support for the object extraction (Harding,

2002). Multiscale analysis also favors the field model that represents the granularity and extent changes explicitly within the spatial scaling operations. This is also consistent with the data structure of big data that is overwhelmingly field-based. Therefore, our new scale model will be field-based, but covers the object model as well.

We need a scale model that also efficiently tracks the scale transformations in multiscale analysis. Since scale is often interpreted as granularity or extent, the predominant solution in multiscale analysis is geospatial scaling operations that homogenize granularity (e.g., up/down-sampling) or extent (e.g., clipping and image stitching). Research has been conducted to assess the impact of such operations. Woodcock and Strahler (1987) explored the correlation between spatial granularity changes (i.e., the authors re-sample images at different spatial granularities) and RS classification accuracy (e.g., forest, road, agriculture, and urban/suburban classification). They pointed out how the accuracy of classification varied with respect to the granularity scaling. They found that finer granularity did not always achieve higher accuracy in classification. Tarnavsky et al. (2008) built off their work to assess the relationship between geospatial granularity scaling and spatial variability (i.e., the variance of normalized difference vegetation index) and concluded that the spatial variability generally decreased when up-sampling the spatial granularities. They also mentioned how changes in spatial extents could affect spatial variability. Wu et al. (2000) went a step further by employing hierarchical geo-statistical models to evaluate the variance of seventeen landscape metrics using data with different spatial granularities. Their study indicated that some landscape metrics varied along with the scaling of granularities and extents but some did not. All these works related to the spatial granularity and extent scaling operations with the changes of properties (e.g., RS classification accuracy and spatial variability measurements), and we followed this approach by

modelling the scale changes as the transformation among the quadruples that are composed of granularity, extent, time, and property.

Researchers have long noted that time possesses its own scale characteristics. Theoretically, time has been represented in linear and non-linear ways (e.g., branching and cyclical time {Huang, Luo, and Van Der Meyden, 2010}). Linear time models are the majority (Claramunt and Thériault, 1995), which enable the use of seconds, hours, and years as the temporal granularities (Goodchild, 2011). But the non-linear way might incorporate different spatial-temporal models (e.g., snapshots and episode cycles) as the granularities and extents (Peuquet, 1994). The temporal scale mismatch between datasets and geospatial analysis also has been documented (Eva and Lambin, 2000). Cash et al. (2006) defined this temporal scale mismatch as the conflict between relatively short analysis cycles and the long-term planning needs required in environmental management. Temporal scaling (e.g., interpolation of data in the middle of two periods) might create new properties, with specific spatial granularities and extents (Antonić et al., 2001). So we have treated time as an independent element from spatial granularity and extent, due to the variety of non-linear temporal scaling methods, such as the cyclical scaling, branch scaling, and the isochronic scaling (Chen et al., 1999).

Lastly, a new scale model is needed to accommodate the handling of big data. Because the volume and velocity of big data exceed the memory of a single commercial computer, the data needs to be split into small chunks and distributed across numerous computing nodes. Data decomposition/ recomposition operations have become routine (Kaisler et al., 2013), which not only change the file size of the data but also the spatial-temporal extents. Various extents play critical roles in knowledge representation (e.g., do the data cover the extent of the highway?), analysis workflow (e.g., need for pre-processing), parameter tuning (e.g., range of variables), and

computing resource allocation (e.g., number and distribution of computing nodes) (Herbst and Karagiannis, 1998). Scope is designed to capture these computational properties and coordinate it with other scale meanings in GIS/RS research (Wang and Armstrong, 2009).

To support multiscale analysis, we should first clarify the complex meanings of scale in the new scale model. Second, the scale transformations of granularity, extent, and time need to be encoded as different property changes. Although the selection of appropriate property depend on existing expertise and knowledge, a new scale model to integrate granularity, extent, time, and property has become quite necessary.

3.3 The Concept of Scope

We propose the concept of Scope as a scale model that contains four elements: spatial granularity, spatial extent, time, and properties. One Scope can therefore be considered as a quadruple of these four elements and we may need a collection of these quadruples in practice. The spatial granularity and extent can take single values in each quadruple, and time can be represented by a given temporal model. Property can take attributional characteristics of the dataset (e.g., data structure and classification) and may contain a collection of different descriptors. Properties also can be represented as the statistical calculations (Diggle, Tawn, and Moyeed, 1998) or feature extraction methods (Câmara et al., 1996). In any given Scope quadruple, the granularity, extent, time, and property (not property values) will be consistent.

A single dataset is not characterized by a single Scope quadruple. Any derivation of a dataset is accompanied by its own quadruple. For example, geospatial scaling operations will generate new Scope quadruples, which may result in different combinations of granularities, extents, time, and properties. Several properties might be combined into a new property (e.g., the normalized difference vegetation index is a combination of measurement from the red and near-

infrared wavelengths); some properties can be removed via filtering. For example, a band selection technique could reduce the number of RS bands, which will decrease the corresponding number of properties within subsequent Scope quadruples (Chang et al., 1999). In practice, we rely on a collection of the Scope quadruples, not a single Scope quadruple, to model this kind of multiscale geospatial analysis.

Our Scope concept focuses less on location and more on the abstraction of scale. In the Scope model, location is represented within properties, but an *(x,y)* location is hard to determine without the specific granularity, extent, and time. In other words, we must establish the concept of Scope before we can establish a field or object model with the specific locations.

To avoid emphasizing the smallest and largest objects in GIScience, we integrate scale space with the generic field model to build our Scope. The dilemma is that we neither understand the most basic element of the universe (the smallest spatial-temporal granularity), nor can we explore the exact boundaries of the universe (the "full" spatial-temporal extent). The generic field model is chosen because it provides ontological support for the object model (Harding, 2002). The transforming of the field model to the object model is relatively easy, but not vice versa (Liu et al., 2008). Another reason to choose the field model is the rapid growth of field-based high performance computing implementations in GIScience (Clark et al., 2003). Wang and Armstrong (2009) also chose the field model to develop spatial computational domain representation. The value of field model also lies in its consistency of granularity, extent, time, and properties inside the given data.

3.3.1 Scope Set

Facing a large number of Scope quadruples, we propose the concept of Scope Set to organize these quadruples as a mathematical *set* (Frank, 1999). The mathematical *set* is a collection of

distinct objects, and we specify the objects as Scope quadruples. Scope Set is based on scale space, which was initially proposed for multi-granularity signal representation in computer vision (Witkin, 1984). Scale space contains a large number of images with various spatial granularities scaled from the original dataset and arranges them in ascending or descending orders. The Scope Set concept extends the scale space by adding spatial extent, time, and property. In one Scope Set, the quadruples are correlated in property and can be converted via the Scope quadruple projection.

We illustrate the concept of Scope Set in Figure 3-2, with the Scope quadruple encoded as *<granularity, extent, time, property>*. In this figure, the property remains the same and the three quadruples stands for different combinations of spatial granularity, spatial extent, and time. Different granularities and extents are recorded explicitly in the quadruples *a*, *b*, and *c*. Quadruple *b* represents the multi-temporal datasets, which is modelled with the generic field as a collection of $\{t_{b1}, t_{b2}, \ldots, t_{bn}\}$, with the element in the collection (e.g., t_{b1}) representing the temporal granularity, and the collection standing for the temporal extent. The quadruple projection between quadruples *a* and *b* is implemented using granularity up-sampling and temporal interpolation; the projection between *b* and *c* is applied via the extent decomposition and the temporal down-sampling.

Multiscale object data can also be represented by the Scope Set. We transform the objects to fields, to avoid generating redundant quadruples for individual objects because each object may be described by different granularities, extents, and time. For example, a road can be modelled as a connection between two locations but can also be represented as a combination of several lanes. We employ the Scope quadruple projection not only for the scale transformation among the quadruples but also to project the object data as the generic field. In this process, a

Stommel diagram (Stommel, 1963) is essential to coordinate various spatial granularities, spatial extents, and time, within each Scope quadruples.

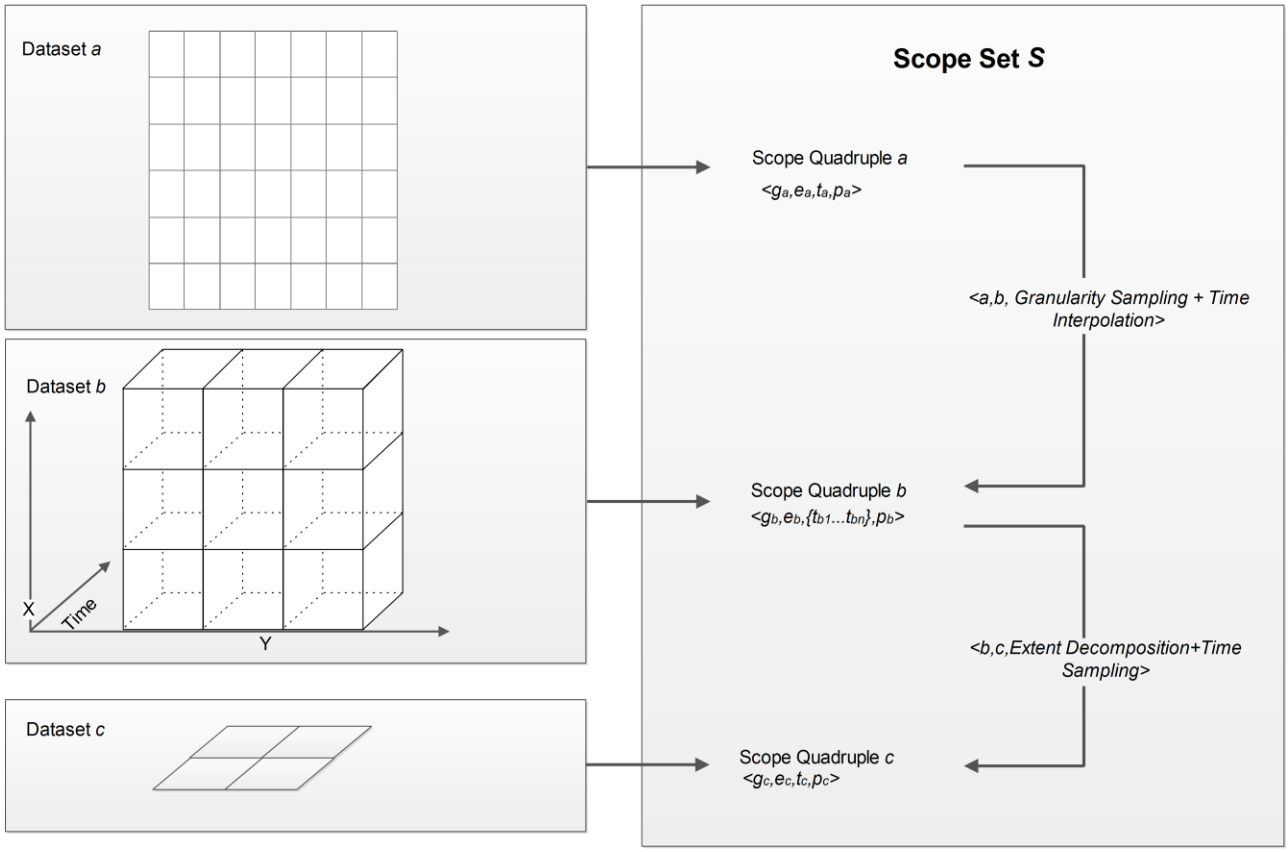


Figure 3-2. Illustration of Scope Set. Scope Set *S* contains three quadruples *a*, *b*, and *c* with the same property. Dataset *b* is scaled from dataset *a*, by keeping the same spatial extent, but changes the spatial granularity via down-sampling and is extended as multi-temporal data cube via temporal interpolation. Dataset *c* is obtained by extent decomposition and temporal sampling, from dataset *b*.

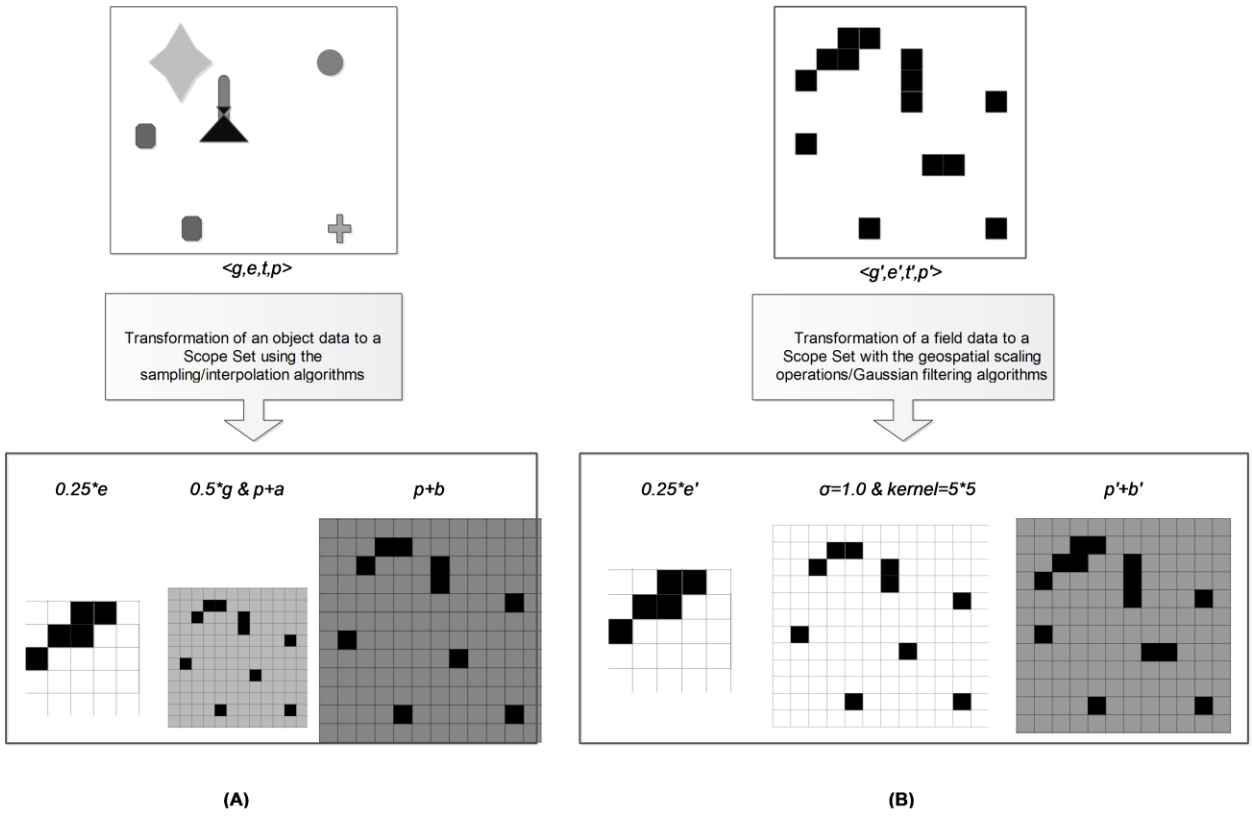


Figure 3-3. Scope Quadruple Projection. (A) is implemented by the transformation of object data representation with sampling/interpolation algorithms, using the equal region decomposition, granularity resampling, and property interpolation algorithms; while (B) is the Scope quadruple projection for field data with geospatial scaling operations and Gaussian filtering algorithms, using spatial extent scaling, Gaussian filtering (σ=1.0 and kernel=5*5), and property scaling.

3.3.2 Scope Quadruple Projection

Another main motivation of Scope is to track various scale transformations such as geospatial scaling operations. Scope Set becomes the metadata for scale representation in geospatial big data. In this section, we demonstrate that Scope also includes an "action item" to model the scale transformation of data. We name the "action item" as the Scope quadruple projection, which is encoded as a tuple <*SourceQuadruple, DestinationQuadruple, Algorithm*>. In Figure 3-2, the quadruple projections are modeled as the tuples <*a, b, Granularity Sampling + Time Interpolation*> and <*b, c, Extent Decomposition + Time Sampling*>, respectively.

We believe that the easiest way currently to connect *SourceQuadruple, DestinationQuadruple* is via linked data. Linked data (Cook et al., 1996) is the method to link related data over the Internet. The linkage is described by various schema, and can be searched and manipulated via semantic queries (Alvares et al., 2007). The linked data-based quadruple projection can also facilitate the future research in modelling geospatial scaling algorithms.

A typical algorithm for Scope quadruple projection is a combination of sampling/interpolation and Gaussian filtering. In the scale space theory, Gaussian filtering is a fundamental algorithm to generate the multi-granularity datasets (Babaud et al., 1986). On the other hand, Liu et al. (2008) concluded the general sampling/interpolation techniques play pivotal roles in specifying the spatial granularity and the extent of data that is structured by the field model. We note other algorithms can be applied as well.

The Scope quadruple projection not only covers the scale transformation between the generic field dataset but also the transformation from a raw object dataset. It should be noted that the implementation of algorithms in the Scope quadruple projection will vary in the object and field data models. For example, in discrete object data, the sampling/interpolation method would be conducted as the spatiotemporal points, isolines of properties, or regular/irregular arrays (e.g., triangulated irregular network) to determine property values, the sizes (granularity), and the distribution (extent) of the grid cells/cubes. In Figure 3-3 (A), the granularity and extent are defined by the sampling/interpolation method with discrete objects, and the value of property is calculated from the original dataset. It is possible to generate multiple Scope quadruples from one object dataset. For example, a traffic accident needs finer spatial-temporal granularity data recording at the center areas of the accident, to capture the traffic changes in real-time. Roads far away from the accident favor coarser granularities, as they are less correlated with the accident.

The algorithms in the Scope quadruple projection needs to be adjusted with respect to the research question.

For continuous field data, the combination of geospatial scaling operations (i.e., granularity and extent transformation) and Gaussian filtering is employed in our implementation. Field data always comes with explicit granularities and extents. Therefore, geospatial scaling operations can generate numerous Scope quadruples with different granularities, extents, and time. The Gaussian filtering algorithm works with pre-defined analysis windows over the field (Liu et al., 2008) to determine the appropriate values of properties. The Scope quadruple projection of field data is illustrated in Figure 3-3 (B).

We note the above implementation of the Scope quadruple projection is subject to the limitations (e.g., additional errors and information lose) of sampling/interpolation and Gaussian filtering algorithms (Prashanth and Shashidhara, 2009); advanced transformation methods can be incorporated (e.g., wavelet transformation {Li, Manjunath, and Mitra, 1995}) in a future study. Although the shapes of the grid cells are selected as rectangles in Figure 3-3, other tessellation methods (e.g., triangulation and hexagonization) also can be utilized.

The Scope Set theoretically provides infinite Scope quadruples to represent scale in multiscale analysis. These quadruples are linked via the Scope quadruple projections. We use the combination of sampling/interpolation, geospatial scaling operations, and Gaussian filtering as an example to implement the Scope quadruple projection. The case study illustrates the use of Scope as a methodology to model scale, providing a theoretical and practical base for multiscale analysis.

3.4 Case Study in Multi-granularity/Multi-extent Road Classification

To demonstrate how Scope can help multiscale analysis, we utilize a multi-granularity/multi-extent road classification as the case study. In this case study, we select data entropy as the property to implement the Scope, due to its wide application in data variance representation (Batty, 1974). The Scope quadruple projection is implemented as the combination of Gaussian filtering and region quadtree decomposition (Shusterman and Feder, 1994).

Road classification plays a critical role in transportation, urban planning, health geography, and disaster management. Classification represents a standard step in interpretation of remotely sensed images. The accuracy of road classification depends on both spatial granularity and extents. If the images containing roads are classified at small extents, some buildings might be mistakenly assigned the label “road” due to their similar spectral signatures and geometric attributes (Shackelford and Davis, 2003). If the spatial extent is too large, the diversity of road properties (e.g., width, length, material, and markers) may reduce the accuracy of the classification.

We use MMCO datasets taken in downtown Montreal in 2007 (0.125m, RGB) for road classification in this case study. If the road classification is implemented at the large extents in downtown Montreal (e.g., one piece of the data shown in Figure 3-4 (A)) then the result is too messy (e.g., Figure 3-4 (B) is the classification result of Figure 3-4 (A)). This messy result is caused by the difference of the road (e.g., width, length, marks, and pavement), shadows, cars, trees, buildings, and paved grounds. Running the road classification at smaller extents seems a better choice. As shown in Figure 3-4 (C) and (D), the road blocks are extracted at a smaller spatial extent (i.e., one sixty-fourth of Figure 3-4 (A)), which requires the post-processing to combine the blocks as the roads after the classification. Figure 3-4 (C) depicts road classification

at 0.125m granularity, in which the accuracy of the classification is deteriorated by the mixture of noisy information, such as cars, trees, and shadows. Figure 3-4 (D) presents the classification at coarser granularity (4m), which is less impacted by the noise but obfuscates some roads and buildings. This challenge leaves a great opportunity for Scope to clarify the correlation between spatial granularities and extents.

In this case study, we implement the classification as a combination of the normalized graph-cut image segmentation (Shi and Malik, 2000) and the support vector machine classification (Song and Civco, 2004), due to their high popularity in RS image analysis. The shadow and vehicle removal algorithms (Pohl and Van Genderen, 1998) are employed to the MMCO data before the classification, to reduce the impact of noisy information.

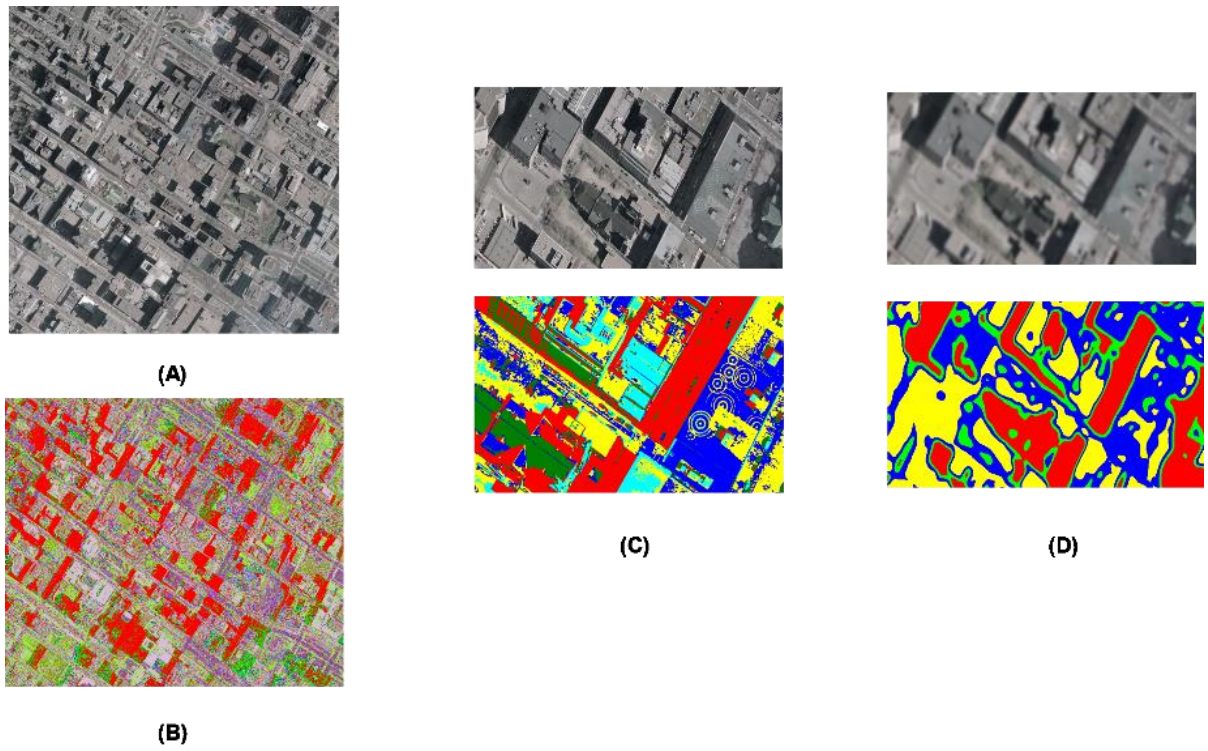


Figure 3-4. Road Classification with Scope. (A) 0.125m MMCO images recorded at downtown Montreal, 2007 (Communauté métropolitaine de Montréal, 2007) covering 774400m^2; (B) road classification using graph-cut segmentation-based classification; (C) road classification at smaller extent (12100 m^2); and (D) road classification with (C) after Gaussian filtering to change the spatial granularity to 4m.

In this case study, we apply an entropy algorithm as the property. The concept of entropy is first defined as a mathematical measurement for information variance (Lin, 1991),

$$H = -\sum_{i=1}^{n} P(s_i) \ln P(s_i) \qquad (3.1)$$

where S is the system with n finite number of possible events S_i, and $P(s_i)$ represents the possibility of the event S_i. The entropy value of homogenous data is zero and a higher entropy value indicates more heterogeneity.

In this case study, the spatial granularity scaling method is the Gaussian filtering and the extent scaling is the region quadtree decomposition. To lower the entropy value, the two spatial scaling operations are conducted together at the beginning. When the value drops below a pre-

defined threshold (100 for this case study), only the spatial granularity scaling (Gaussian filtering) is employed. This is because the spatial extent scaling changes entropy values more at the coarser spatial granularity than at the finer granularity (Wang et al., 2003). The implication of this on our results is the larger entropy difference between S_1 and S_2 (18.51) than the value between S_2 and S_3 (18.14), as shown in Table 3-1. Then the accuracy of the road classification is obtained via 500 ground-truthing control points and listed in Table 3-1.

The Scope Set is implemented as a collection of Scope quadruples. Since only one time is involved in this case study, we encode them as tuples: <*Granularity, Extent, Property*>, in which the property is calculated as the average entropy over all the tiles. As shown in Table 3-1, the average entropy is calculated as $\text{Entropy} * \log(\text{Extent}/\text{Granularity})$, in which the Entropy is obtained from equation (3.1) and normalized by the factor $\log(\text{Extent}/\text{Granularity})$. The normalization is implemented to distinguish data with smaller extent and higher entropy value because one covers larger extent but presents lower entropy.

Table 3-1. Road Classification with Scope

Scope Quadruple	Granularity (m)	Tile Extent (m^2)	Mean (Entropy* log(Extent/Granularity)) for Tiles	Scope Quadruple Projection	Overall Accuracy (%)[1]
S_1	0.13	774400	118.85	*<S_1,S_2,Quadtree Decomposition + Gaussian Filtering>*	66.1
S_2	0.25	193600	100.34	*<S_2,S_3,Quadtree Decomposition + Gaussian Filtering>*	72.1
S_3	0.50	48400	82.20	*<S_3,S_4, Gaussian Filtering>*	77.9
S_4	1.00	48400	73.92	*<S_4,S_5, Gaussian Filtering>*	81.0
S_5	2.00	48400	67.45	*<S_5,S_6, Gaussian Filtering>*	86.7
S_6	4.00	48400	59.16	N/A	81.3

From Table 3-1, the accuracy of road classification increases along with the spatial granularity up-scaling until the spatial[1] granularity reaches 4m. The quad-tree decomposition is only employed at Scope quadruples S_1 and S_2. If the spatial decomposition is employed for the rest of the Scope Set, the problems in Figure 3-4 (D) will happen. The highest accuracy appears at S_5, when most of the noisy information (cars and trees) become nearly negligible at 2m granularity. The 2m granularity and 48400m^2 extent combination may not be the optimal solution for road classification, as Scope Set provides infinite granularity and extent combinations. But this case study has illustrated the effectiveness of Scope in multi-granularity/multi-extent analysis.

3.5 Summary

In this paper, we create the concept of Scope to clarify the complex meanings of scale in geospatial analysis, and track different kinds of scale transformations. Scope is a scale model that integrates granularity, extent, time, and property. Time is treated as an independent element. Because time in geography is much more than the linear model, and its scaling presents considerable more diversity than the combination of temporal granularities and extents. Thus, time is always by apriority spatial-temporal models (Yuan, 1996). Property stands for the metadata as the data quality and structural descriptors, and the possible actions we can take with the dataset. Property can capture the scaling effect of the other three elements, such as the spatial granularity up/down-sampling and spatial extent decomposition/recomposition.

---

[1] We note there are granularity mismatches between the ground truth data and the classification results, so we apply the spatial granularity scaling to the ground truth data as well.

The concept of Scope is built on the generic field model and scale space theory, and Scope is composed of three parts—the Scope quadruple, Scope Set, and the Scope quadruple projection. The Scope Set consists of numerous Scope quadruples, as <*granularity, extent, time, property*>, to represent data with different scales. The Scope quadruple projection represents the transformation among Scope quadruples via the sampling/interpolation and Gaussian filtering approaches in this paper. We note that other methods can be employed as well for the quadruple projection. The case study verifies the effectiveness of Scope in tracking the scale changes for the multi-granularity/multi-extent analyses and improves the accuracy of road classification.

The main contribution of the Scope model is to provide the clarification and traceability for scale meanings and scaling operations. Moreover, Scope quadruple and Scope quadruple projection build the theoretical base for semantic modelling of various scaling operations, which can further boost the research of scale and multiscale analysis in GIScience.

In our future study, we plan to investigate the Scope set as an *Abelian Group* for scale modelling (Frank, 1999), which will further tighten the combination of Scale Set and the transformations of the quadruples in the Scope Set. The *Abelian Group* defines the commutative operations on group elements, and acts as the theoretical foundation for modelling the relationship among the spatial-temporal scale transformations and Scope quadruples. A Resource Description Framework implementation (Cook et al., 1996) will be developed for modelling the relationship among the Scope quadruple projections, to facilitate the query, select, and applications of various scaling operations. Different data structures can also be integrated to enhance the storage and retrieval of Scope quadruple and Scope quadruple projections, such as the Hybrid Spatio-Temporal Data Model and Structure (Sengupta and Yan, 2004). Although the *Abelian Group* enables the merge of Scope Set and Scope quadruple projection, how to choose

the appropriate scaling operations remains an open research question because not all spatial-temporal scale transformations are commutative among the Scope quadruples.

3.6 Acknowledgement

This research is funded by a Microsoft Azure Research Award.

3.7 Reference

Alvares, L. O., Bogorny, V., Kuijpers, B., de Macedo, J. A. F., Moelans, B., & Vaisman, A. (2007). A model for enriching trajectories with semantic geographical information. In Proceedings of the 15th annual ACM international symposium on Advances in geographic information systems (p. 22). ACM.

Antonić, O., Križan, J., Marki, A., & Bukovec, D. (2001). Spatio-temporal interpolation of climatic variables over large region of complex terrain using neural networks. Ecological Modelling, 138(1), 255-263.

Aspinall, R. J. (2001). Modelling wildlife distribution from multi-scale spatial data with GIS. Modelling scale in geographical information science. New York: John Wiley and Sons, 181-192.

Atkins, C. B., Bouman, C. A., & Allebach, J. P. (2001). Optimal image scaling using pixel classification. In Image Processing, 2001. Proceedings. 2001 International Conference on (Vol. 3, pp. 864-867). IEEE.

Atkinson, P. M., & Tate, N. J. (2000). Spatial scale problems and geostatistical solutions: a review. The Professional Geographer, 52(4), 607-623.

Babaud, J., Witkin, A. P., Baudin, M., & Duda, R. O. (1986). Uniqueness of the Gaussian kernel for scale-space filtering. IEEE transactions on pattern analysis and machine intelligence, (1), 26-33.

Batty, M. (1974). Geospatial entropy. Geographical analysis, 6(1), 1-31.

Benz, U. C., Hofmann, P., Willhauck, G., Lingenfelder, I., & Heynen, M. (2004). Multi-resolution, object-oriented fuzzy analysis of remote sensing data for GIS-ready information. ISPRS Journal of photogrammetry and remote sensing, 58(3), 239-258.

Camara, G., Egenhofer, M. J., Ferreira, K., Andrade, P., Queiroz, G., Sanchez, A., ... & Vinhas, L. (2014). Fields as a generic data type for big spatial data. In International Conference on Geographic Information Science (pp. 159-172). Springer, Cham.

Câmara, G., Monteiro, A. M. V., Paiva, J. A., Gomes, J., & Velho, L. (2000). Towards a unified framework for geographical data models. In Proc.

Câmara, G., Souza, R. C. M., Freitas, U. M., & Garrido, J. (1996). SPRING: Integrating remote sensing and GIS by object-oriented data modelling. Computers & graphics, 20(3), 395-403.

Cash, D., Adger, W. N., Berkes, F., Garden, P., Lebel, L., Olsson, P., ... & Young, O. (2006). Scale and cross-scale dynamics: governance and information in a multilevel world. Ecology and society, 11(2).

Celik, T. (2009). Multiscale change detection in multitemporal satellite images. IEEE Geoscience and Remote Sensing Letters, 6(4), 820-824.

Chang, C. I., Du, Q., Sun, T. L., & Althouse, M. L. (1999). A joint band prioritization and band-decorrelation approach to band selection for hyperspectral image classification. IEEE Transactions on Geoscience and Remote Sensing, 37(6), 2631-2641.

Chen, J. M., Liu, J., Cihlar, J., & Goulden, M. L. (1999). Daily canopy photosynthesis model through temporal and spatial scaling for remote sensing applications. Ecological modelling, 124(2), 99-119.

Claramunt, C., & Thériault, M. (1995). Managing time in GIS an event-oriented approach. In Recent Advances in Temporal Databases (pp. 23-42). Springer London.

Clarke, K. C. (2003). Geocomputation's future at the extremes: high performance computing and nanoclients. Parallel Computing, 29(10), 1281-1295.

Clarke, K. C., & Gaydos, L. J. (1998). Loose-coupling a cellular automaton model and GIS: long-term urban growth prediction for San Francisco and Washington/Baltimore. International journal of geographical information science, 12(7), 699-714.

Communauté métropolitaine de Montréal (2005). Montreal Metropolitan Community Orthophotos, 299-5038.

Cook, D., Majure, J. J., Symanzik, J., & Cressie, N. (1996). Dynamic graphics in a GIS: exploring and analyzing multivariate spatial data using linked software. Computational Statistics, 11(4), 467-480.

Coppin, P., Jonckheere, I., Nackaerts, K., Muys, B., & Lambin, E. (2004). Review ArticleDigital change detection methods in ecosystem monitoring: a review. International journal of remote sensing, 25(9), 1565-1596.

Couclelis, H. (1992). People manipulate objects (but cultivate fields): beyond the raster-vector debate in GIS. In Theories and methods of spatio-temporal reasoning in geographic space (pp. 65-77). Springer Berlin Heidelberg.

Cova, T. J., & Goodchild, M. F. (2002). Extending geographical representation to include fields of spatial objects. International Journal of Geographical Information Science, 16(6), 509-532.

De Smith, M. J., Goodchild, M. F., & Longley, P. A. (2009). Geospatial analysis. Matador.

Diggle, P. J., Tawn, J. A., & Moyeed, R. A. (1998). Model-based geostatistics. Journal of the Royal Statistical Society: Series C (Applied Statistics), 47(3), 299-350.

DMTI Spatial Inc., 2006, 2009, and 2012. Montreal Satellite StreetView, Markham ON: DMTI Spatial Inc., 2006, 2009 and 2012.

Eastman, J. R. (2001). Guide to GIS and image processing Volume. Clark Labs, 2, 1-144.

Eva, H., & Lambin, E. F. (2000). Fires and land-cover change in the tropics: a remote sensing analysis at the landscape scale. Journal of Biogeography, 27(3), 765-776.

Floros, G., van der Zander, B., & Leibe, B. (2013). Openstreetslam: Global vehicle localization using openstreetmaps. In Robotics and Automation (ICRA), 2013 IEEE International Conference on (pp. 1054-1059). IEEE.

Frank, A. U. (1999). One step up the abstraction ladder: Combining algebras-from functional pieces to a whole. In International Conference on Spatial Information Theory (pp. 95-107). Springer, Berlin, Heidelberg

Goodchild, M. F. (2011). Scale in GIS: An overview. Geomorphology, 130(1), 5-9.

Goodchild, M. F., Yuan, M., & Cova, T. J. (2007). Towards a general theory of geographic representation in GIS. International journal of geographical information science, 21(3), 239-260.

Harding, J. (2002). Geo-ontology Concepts and Issues. Report of a workshop on Geo-ontology. Ikley UK, September.

Heeger, D. J. (1987). Model for the extraction of image flow. JOSA A, 4(8), 1455-1471.

Herbst, J., & Karagiannis, D. (1998). Integrating machine learning and workflow management to support acquisition and adaptation of workflow models. In Database and Expert Systems Applications, 1998. Proceedings. Ninth International Workshop on (pp. 745-752). IEEE.

Huang, X., Luo, C., & Van Der Meyden, R. (2010). Improved bounded model checking for a fair branching-time temporal epistemic logic. InInternational Workshop on Model Checking and Artificial Intelligence (pp. 95-111). Springer Berlin Heidelberg.

Inglada, J., & Mercier, G. (2007). A new statistical similarity measure for change detection in multitemporal SAR images and its extension to multiscale change analysis. Geoscience and Remote Sensing, IEEE Transactions on, 45(5), 1432-1445.

Jordan, T., Raubal, M., Gartrell, B., & Egenhofer, M. (1998). An affordance-based model of place in GIS. In 8th Int. Symposium on Spatial Data Handling, SDH (Vol. 98, pp. 98-109).

Kaisler, S., Armour, F., Espinosa, J. A., & Money, W. (2013). Big data: issues and challenges moving forward. In System Sciences (HICSS), 2013 46th Hawaii International Conference on (pp. 995-1004). IEEE.

Li, H., Manjunath, B. S., & Mitra, S. K. (1995). Multisensor image fusion using the wavelet transform. Graphical models and image processing, 57(3), 235-245.

Lin, J. (1991). Divergence measures based on the Shannon entropy. IEEE Transactions on Information theory, 37(1), 145-151.

Liu, Y., Goodchild, M. F., Guo, Q., Tian, Y., & Wu, L. (2008). Towards a general field model and its order in GIS. International Journal of Geographical Information Science, 22(6), 623-643..

Marceau, D. J., & Hay, G. J. (1999). Remote sensing contributions to the scale issue. Canadian Journal of Remote Sensing, 25(4), 357-366.

Miller, H. J., & Goodchild, M. F. (2015). Data-driven geography. GeoJournal,80(4), 449-461.

Ouyang, Y., Wentz, E. A., Ruddell, B. L., & Harlan, S. L. (2014). A Multi-Scale Analysis of Single-Family Residential Water Use in the Phoenix Metropolitan Area. JAWRA Journal of the American Water Resources Association, 50(2), 448-467.

Perona, P., & Malik, J. (1990). Scale-space and edge detection using anisotropic diffusion. IEEE Transactions on Pattern Analysis and Machine Intelligence, 12(7), 629-639.

Peuquet, D. J. (1994). It's about time: A conceptual framework for the representation of temporal dynamics in geographic information systems. Annals of the Association of American Geographers, 84(3), 441-461.

Pohl, C., & Van Genderen, J. L. (1998). Review article multisensor image fusion in remote sensing: concepts, methods and applications. International Iournal of Remote Sensing, 19(5), 823-854.

Prashanth, H. S., Shashidhara, H. L., & KN, B. M. (2009). Image scaling comparison using universal image quality index. In Advances in Computing, Control, & Telecommunication Technologies, 2009. ACT'09. International Conference on (pp. 859-863). IEEE.

Schuurman, N., Fiedler, R. S., Grzybowski, S. C., & Grund, D. (2006). Defining rational hospital catchments for non-urban areas based on travel-time. International Journal of Health Geographics, 5(1), 43.

Sengupta, R. and Yan, C., 2004. A Hybrid Spatio-Temporal Data Model and Structure for Efficient Storage and Retrieval of Land Use Information. Transactions in GIS 8(3): 351-366.

Shackelford, A. K., & Davis, C. H. (2003). A combined fuzzy pixel-based and object-based approach for classification of high-resolution multispectral data over urban areas. IEEE Transactions on GeoScience and Remote sensing, 41(10), 2354-2363.

Shi, J., & Malik, J. (2000). Normalized cuts and image segmentation. IEEE Transactions on pattern analysis and machine intelligence, 22(8), 888-905.

Shusterman, E., & Feder, M. (1994). Image compression via improved quadtree decomposition algorithms. IEEE Transactions on image processing, 3(2), 207-215.

Song, M., & Civco, D. (2004). Road extraction using SVM and image segmentation. Photogrammetric Engineering & Remote Sensing, 70(12), 1365-1371.

Spaccapietra, S., Parent, C., & Vangenot, C. (2000). GIS databases: From multiscale to multirepresentation. In International Symposium on Abstraction, Reformulation, and Approximation (pp. 57-70). Springer Berlin Heidelberg.

Stommel, H. (1963). Varieties of oceanographic experience. Science,139(3555), 572-576.

Store, R., & Jokimäki, J. (2003). A GIS-based multi-scale approach to habitat suitability modeling. Ecological Modelling, 169(1), 1-15.

Tarnavsky, E., Garrigues, S., & Brown, M. E. (2008). Multiscale geostatistical analysis of AVHRR, SPOT-VGT, and MODIS global NDVI products. Remote Sensing of Environment, 112(2), 535-549.

Vlachos, D. G. (2005). A review of multiscale analysis: examples from systems biology, materials engineering, and other fluid–surface interacting systems. Advances in Chemical Engineering, 30, 1-61.

Wang, S., & Armstrong, M. P. (2003). A quadtree approach to domain decomposition for spatial interpolation in grid computing environments.Parallel Computing, 29(10), 1481-1504.

Wang, S., & Armstrong, M. P. (2009). A theoretical approach to the use of cyberinfrastructure in geographical analysis. International Journal of Geographical Information Science, 23(2), 169-193.

Wang, Z., Simoncelli, E. P., & Bovik, A. C. (2003). Multiscale structural similarity for image quality assessment. In Signals, Systems and Computers, 2004. Conference Record of the Thirty-Seventh Asilomar Conference on (Vol. 2, pp. 1398-1402). IEEE.

Witkin, A. (1984). Scale-space filtering: A new approach to multi-scale description. In Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'84. (Vol. 9, pp. 150-153). IEEE.

Woodcock, C. E., & Strahler, A. H. (1987). The factor of scale in remote sensing. Remote sensing of Environment, 21(3), 311-332.

Wu, H., & Li, Z. L. (2009). Scale issues in remote sensing: A review on analysis, processing and modeling. Sensors, 9(3), 1768-1793.

Wu, J., Jelinski, D. E., Luck, M., & Tueller, P. T. (2000). Multiscale analysis of landscape heterogeneity: scale variance and pattern metrics. Geographic Information Sciences, 6(1), 6-19.

Yuan, M. (1996). Temporal GIS and spatio-temporal modeling. In Proceedings of Third International Conference Workshop on Integrating GIS and Environment Modeling, Santa Fe, NM (Vol. 33).

Connecting Statement: Handling Big Data Volume via the Decomposition/Recomposition Framework

The concept of Scope has been delineated in Chapter 3. In Section 3.4, my case study indicated that Scope can efficiently track geospatial scaling operations in road classification. Since big data is routinely decomposed into smaller chunks for distributed computing, Scope becomes a useful method for handling scale in big data decomposition.

Decomposition alone is not enough for the scale challenges in big data because the distributed LUCC results need to be re-combined to generate final results. Therefore, I propose the decomposition/recomposition framework for big data analysis in the distributed computing environment in Chapter 4. Chapter 4 presents the decomposition/recomposition framework for big dataflow management on MapReduce and also highlights the artificial border challenge that is caused by the decomposition. Fortunately, this challenge is handled by the Recomposition process I cover in Chapter 4. The case study with big image segmentation proves the success of the decomposition/recomposition framework in big data volume handling.

Chapter 4 was published on *Annals of Geographical Information Science*, 2014. The manuscript contained in this chapter was co-authored with my supervisors, Prof. Renée Sieber, and Prof. Margaret Kalacska. I am the primary author and contributed the decomposition/recomposition framework and the Geospatial CyberInfrastructure-based implementation. Prof. Sieber noticed the artificial border challenge and helped me in writing this article. Prof. Kalacska gave me advice in image segmentation and also helped in writing this article.

Chapter 4. The Challenges of Image Segmentation in Big Remotely Sensed Imagery Data

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 predefined 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 data sets must be decomposed for the 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 artefacts. 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 312.07 GB high-resolution colour aerial photo with Hadoop. K-means-based image segmentation is selected as the case study. We deploy the architecture in private cloud and public cloud implementation, respectively. The results demonstrate the benefits of the decomposing and recomposing methods in segmenting images, removing fake artefacts and reducing information distortion. More general problems in big data are revealed, among them I/O problems, particularly in the amount of preprocessing and post-processing that will be required in any analysis of big imagery data. We conclude with implications for scalability and suggestions to speed up decomposition and recomposition.

Keywords: big data; image segmentation; geospatial cyberinfrastructure; spatial feature extraction; cloud computing; MapReduce; decomposition; recomposition.

4.1 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 (DigitalGlobe Inc., Longmont, CO, USA) sensor provides 1-m spatial resolution panchromatic images with 3 days revisiting interval (Richards 2013). These enhanced resolutions reveal detailed spatio-temporal 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 with the corresponding data analysis and the provisioning of computing resources. 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), Compute Unified Device Architecture (Xia, Kuang, and Li, 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 4.2 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 4.3. We then test this GCI in Section 4.4 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 4.5. 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.

4.2 Literature Review

4.2.1 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, Ma, and He (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.

4.2.2 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 4-1 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.

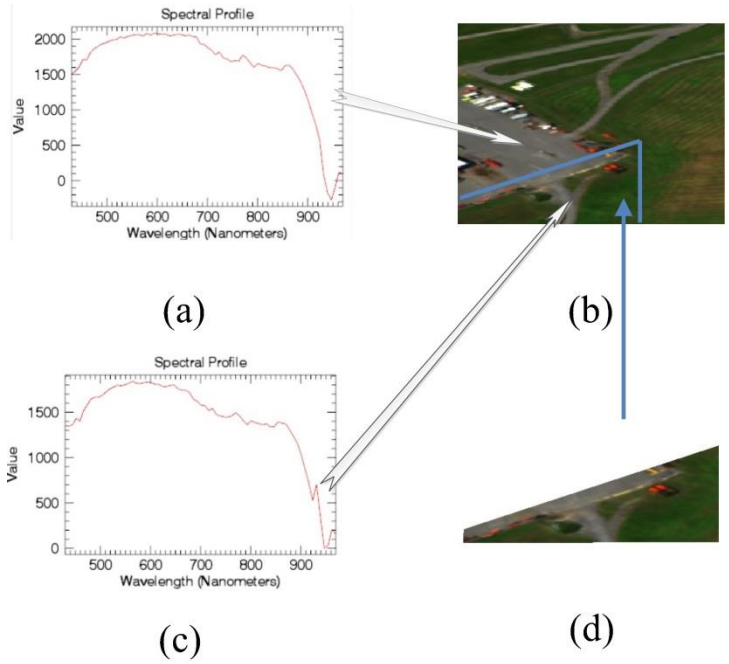


Figure 4-1. (a) Spectral signature of one sampling point on the parking ground; (b) Airborne Visible / Infrared Imaging Spectrometer image; (c) Spectral signature of one sampling point on the highway; (d) A fake "road" generated by image splitting.

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 4-2 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 4-1 (b), and label it as “road”. As we show in Figure 4-1 (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.

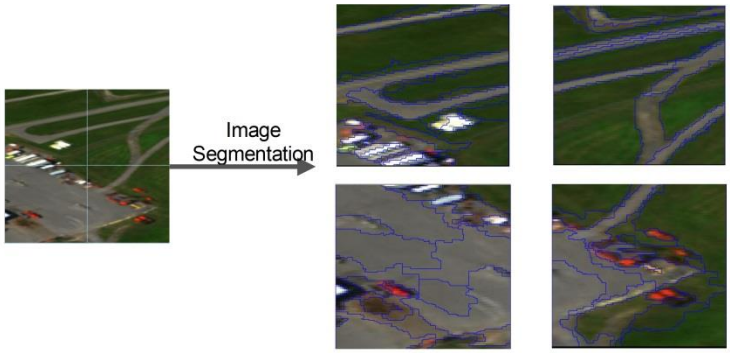


Figure 4-2. Splitting Figure 4-1 (b) into 2×2 chunks and segmenting each chunk, the image segmentation is generated by eCognition®, with scale=50 and color=0.5.

In Figure 4-2, 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 4-2, 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 4-1. These are a collection of five challenges in which image splitting causes fake features in image segmentation (Figure 4-2). All these challenges grow with big data and will likely see greater attention in GIS and RS research. The artificial border challenge covers the concept of both vector object border in GIS and the image object border in RS. Thus this challenge requires a closer integration of GIS and RS research for big data handling.

Table 4-1. 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

4.2.3 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. Moreover, cloud computing can free research scientist from the onerous testbed building and system administration, enabling them to focus on the scientific problem-solving (Yang *et al.* 2011). 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.

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

4.3.1 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 imagery data.

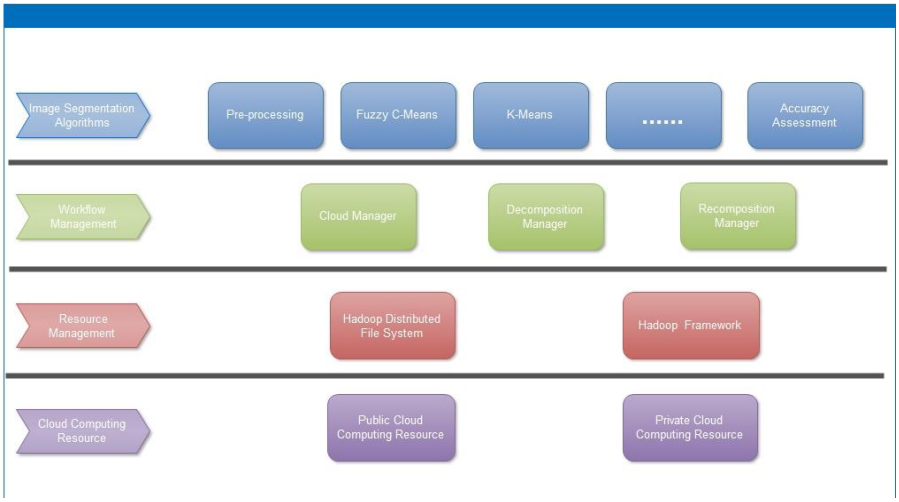


Figure 4-3. GCI Architecture

The architecture of our GCI is shown in Figure 4-3, 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 4.3.2. 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 4.3.3.

4.3.2 The Workflow Management Layer

The general workflow of segmenting big RS imagery is depicted in Figure 4-4, 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.

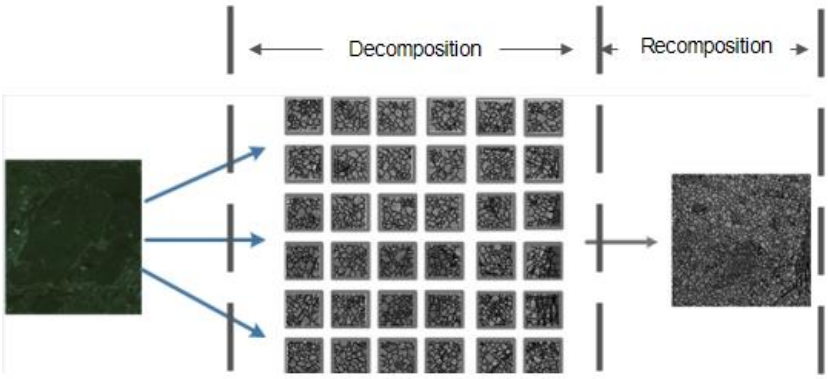


Figure 4-4. Overview of the Decomposition/Recomposition Workflow Management Framework

A detailed description of these steps are given is Figure 4-5. 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.

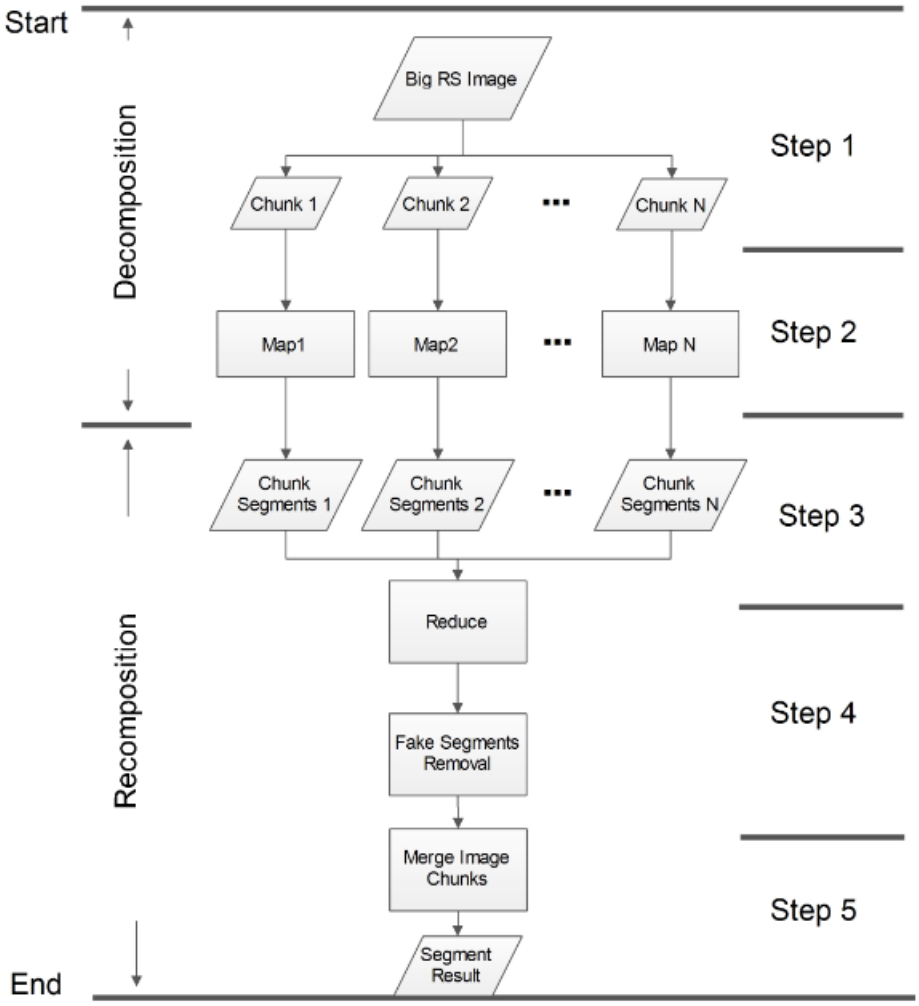


Figure 4-5. Steps of Decomposition/Recomposition with MapReduce

4.3.2.1 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:

$$chunk\_size = \left\lfloor GCD\left(\frac{S \times k}{m}, \frac{N}{m}\right) \right\rfloor \tag{4.1}$$

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.

4.3.2.2 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 4-6). 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 4-6. 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. The pseudo code of the moving window-based fake segments removal algorithm is depicted in Appendix I.

4.3.3 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, Lymburner, and Volk, 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.

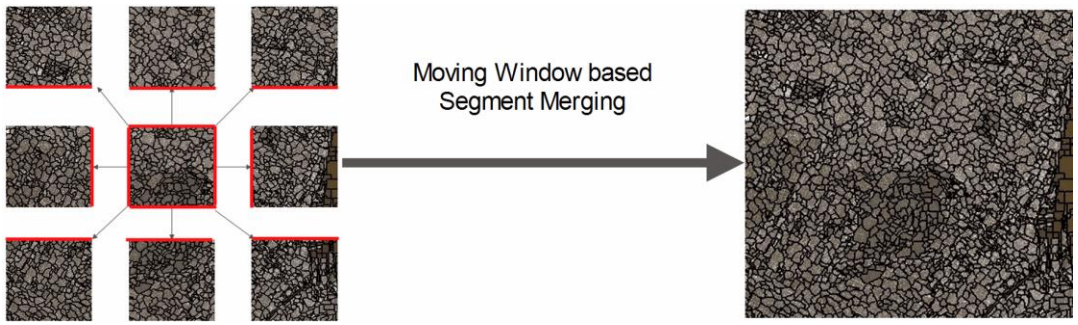


Figure 4-6. Moving Window based Segment Merging Process

4.4 Evaluation of the GCI for Image Segmentation of Big RS Imagery

4.4.1 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 and Turi, 1999), due to its popularity and robust computational complexity. The splitting method is our spatial extent splitting methods in Section 4.3.2.1, and artificial borders and corresponding fake segments are removed with our moving window based approach in Section 4.3.2.2. 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 4-2. 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* VMs 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 4-3 and 4-4, respectively.

Table 4-2. 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 4-3. 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 4-4. 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.

4.4.2 Discussion

Big RS image segmentation is evaluated with two different deployments, using private and public cloud computing respectively. The results from *map* VMs are combined in *reduce* VM, with fake segments removed. As RS data increasing, artificial border challenges will become more important and attract more research interests. However, the bottleneck in the recomposition process cannot be neglected, which needs to be parallelized in future research.

Our GCI combines advanced computing techniques with image segmentation algorithms successfully. It is proven to integrate different types of computing resource (e.g., public and private cloud) to provision image segmentation process, with corresponding resource management functionalities. Also MapReduce framework is embedded, to provide efficient big data processing. Moreover, the automatic deployment of various image segmentation algorithms, pre-processing and accuracy assessment methods can significantly ease big RS image segmentation. To extend our GCI, other parallel computing frameworks will be combined in future research, such as MPI (Message Passing Interface) and Apache Storm.

By comparing Tables 4-3 and 4-4, it seems using Eucalyptus private cloud is a better choice for image segmentation in big RS imagery datasets due to less computation time and costs. But 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 is quite small compared with the public cloud, which limits the scalability. We cannot use more *map* tasks in our experiment because the hardware computing resource of the private cloud is not enough. Considering all the hidden costs, using public cloud for big RS imagery processing is more economical for short-term projects.

On the other hand, big data challenges in RS research also resulted in the high I/O costs of both public and private. Research scientists may choose private cloud to avoid part of these costs, but moving big data across *map* and *reduce* VMs are still quite expensive. In the future research of cloud computing, big data I/O cost should be given special attention (Khajeh-Hosseini *et al.,* 2012; Kondo *et al.*, 2009). The high cost of transferring data across clouds becomes an important factor restricts the utilization of cloud. To summarize, high I/O cost is a new bottleneck in the development of cloud computing and big data research.

Using 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 guarantee there is no attack or information leaking. Considering the current development of cloud computing, private cloud is preferred for big RS image segmentation.

4.5 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 which 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 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.

4.6 Acknowledgement

The authors thanks Dr. Arroyo from the Geographic Information Center, McGill University, for the use of the aerial photography.

4.7 Funding

The funding for this research was provided by the Social Sciences and Humanities Research Council, Canada (SSHRC)and the Global Environmental and Climate Change Centre(GEC3), McGill University.

4.8 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., 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., 2009. Introduction to distributed geographic information processing research. International Journal of Geographical Information Science 23, 553–560.

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., 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.

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.

Appendix I: Moving Window-based Fake Segments Removal

Algorithm: Moving Window-based Fake Segments Removal (Xing *et al*., 2014)

Input: image chunk array *C*, and corresponding segments overlay *S*

Output: new image segments overlay *S'*

for each image chunk c_i in *C*:
load(corresponding s_i);
mark all the segment on the border of s_i and store them as *B*;
N=load(neighbors of c_i);
A=merge(c_i,*N*);
B'=cluster(*A*);
for each border segment b_i in *B*:
load(corresponding b_i' from *B'*);
difference=compare(b_i',b_i);
if (difference > threshold) then
$s_i=s_i - b_i$;
b_i=merge(b_i ,b_i');
$s_i=s_i + b_i$;
end for
$i=i+1$;
load(c_i);
end for
S'=merge($s_1'...s_n'$);

End

Connecting Statement: Using Scale Invariant Image Features for LUCC Detection

Chapter 4 handles the dataflow for big data analysis in the distributed environment. But we still need a new change detection algorithm that can help extract LUCC from imagery datasets with heterogeneous spatial granularities and extents. In Chapter 5, I invent the scale invariant LUCC detection algorithm. This algorithm identifies the change areas by comparing the scale invariant image features (i.e., Maximally Stable Extremal Region {MSER} and Scale Invariant Feature Transformation {SIFT}), to avoid the additional errors incurred by the image scaling operations. The decomposition is implemented with spatial entropy to roughly guarantee the similar number of image features in the tiles of image pairs. The recomposition removes the fake features generated due to the splitting borders. To some extent, the scale invariant LUCC detection method is developed based on decomposition/recomposition framework introduced in Chapter 4.

This chapter has been submitted for publication in *ISPRS Journal of Photogrammetry and Remote Sensing*. The manuscript contained in this chapter was co-authored with my supervisor, Prof. Renée Sieber, and members of my doctoral supervisory committee including Prof. Terrence Caelli. I was the primary author and contributed the initial idea of combining SIFT and MSER for LUCC study, the implementation of the scale invariant LUCC detection method, and the case study at the Greater Montreal Area. Prof. Renée Sieber enhanced the workflow of the scale invariant LUCC detection method and the logic of this article. Prof. Terrence Caelli improved the change map smoothing algorithm and polished this paper.

Chapter 5. A Scale Invariant Change Detection Method for Land Use/Cover Change Research

Abstract

Land Use/Cover Change (LUCC) detection relies increasingly on comparing remote sensing images with different spatial and spectral scales. Based on scale invariant image analysis algorithms in computer vision, we propose a scale invariant LUCC detection method to identify changes from scale heterogeneous images. We test its scale invariance with a LUCC case study in Montreal, Canada, 2005-2012.

Keywords: Land Use/Cover Change Detection; Scale Variance; Scale Invariant Feature Transformation; Maximally Stable Extremal Region; Hadoop; Cloud Computing.

5.1. Introduction

Big data provides us with numerous new approaches for LUCC research but it causes problems related big data's large volume, complex variety, increasing velocity, and growing difficulties in veracity (Miller and Goodchild, 2015). Among the four "Vs" of big data, the predominant focus in LUCC research is on volume (e.g., Hampton et al., 2013). Our paper emphasizes the variety and specifically the various scales that are offered (i.e., different spatial, spectral, and temporal granularities and extents) (Goodchild, 2011). Because LUCC uses two or more datasets to identify changes, this introduces potential issues in scale variance (Woodcock and Strahler, 1987).

Ordinarily, to identify LUCC one interpolates or re-samples one or more datasets to homogenize spatial granularities (i.e., resolutions) and extents (we define the spatial scale as the combination of spatial granularity and extent). These spatial scaling operations can cause various problems like the generation of erroneous artifacts (Kwok and Sun, 1993), loss of information

(Sheikh and Bovik, 2006), and distortion of geographic entities (Prashanth et al., 2009). As a result of these spatial scaling operations, LUCC accuracy can be significantly degraded by scale variance (Olofsson et al., 2014) particularly if we wish to take advantage of the high resolution characteristic of big data.

To avoid the drawbacks of using spatial interpolation or re-sampling techniques, research scientists have investigated novel solutions to handle the challenge of scale variance. For example, Chen et al. (2012) clustered pixels into image objects prior to comparison and then compared the geo-registered objects from datasets at two different scales. Singh (1989) bypassed the comparison of image pixels and explored a post-classification method to extract LUCC by comparing the class label maps. Both approaches assume that the scale variance in any LUCC would be minor and that geo-registration would be sufficient to compare image objects. Big data does not make these assumptions by creating new multi-scale challenges for the study of LUCC.

Computer vision algorithms have also been explored to tackle the scale variance challenge (Radke et al., 2005). These algorithms are interesting because they focus on the differentiation of objects within datasets and do not rely on geo-registration as the objects may be moving image to image. An example of the utility of computer vision for LUCC can be found in Dellinger et al. (2014) who proposed using the Scale Invariant Feature Transformation (SIFT) (Lowe, 2004) approach to LUCC detection by extracting and comparing stable scale-points between two remotely sensed images and clustering changed points (i.e., points that failed to find their corresponding points on the other image) as LUCC regions. However, they did not test the scale invariance of SIFT on heterogeneous scale (i.e., granularity and extent) data.

We propose a scale invariant LUCC detection method that draws from computer vision. This method integrates spatial decomposition, image feature (characteristics) comparison,

change map smoothing, and image LUCC labelling. We will show that: (1) LUCC can be extracted by comparing scale invariant image features directly without spatial interpolation or re-sampling methods; (2) discrimination of scale invariant image features can be enhanced by the integration of extent, shape, and spectral information for LUCC; and (3) high performance computing can provide significant support in the scale invariant LUCC detection workflow.

The rest of this paper is organized as follows. Section 5.2 enumerates the benefits and challenges of scale invariant algorithms derived from computer vision. Our scale invariant LUCC detection method is introduced in Section 5.3, which is based on the integration of SIFT and the Maximally Stable Extremal Region (MSER). Section 5.4 is a case study in the Greater Montreal Area from 2006-2012, which evaluates our scale invariant LUCC detection algorithm. This paper concludes in Section 5.5.

5.2. Handling Scale Variance with Computer Vision Algorithms

A large body of computer vision algorithms have been proposed to study scale variance. Scale space filtering is the most widely applied approach in computer vision for scale variance (Witkin, 1984). Scale space consists of multiple images that have been "filtered" from a single original image to generate specified granularities. Scale space filtering enables multi-granularity analysis to identify scale invariant image features (Huo et al., 2008). The filtering can be implemented by various algorithms, such as wavelet transformation (Celik, 2009), discrete cosine transformation (Merhav and Bhaskaran, 1996) and Gaussian convolution (Lowe, 2004). Image fusion is another important scale variance handling method in computer vision, which merges relevant information from at least two images at different spectral and spatial granularities to achieve higher granularities (Li et al., 1995). For example, image fusion with

multispectral IKONOS (4m, red, green, blue, and infra-red) and panchromatic (1m, grayscale) IKONOS images will generate a new image with 1m resolution and 4 bands of information.

Perona and Malik (1990) and others offered good examples of how computer vision studies differ from LUCC. Although they (ibid.) explored changes in image object boundaries at different spatial granularities, their study was conducted with everyday object extents (e.g., 1 mm at $1m^2$). LUCC works with larger extents and a broader range of granularities. Their study also was conducted with a single image but LUCC involves comparing images taken at different times. Their study considered changes in image object characteristics; however, LUCC functions at the image level and detects changes throughout the image extents. Ohn-Bar and Trivedi (2014) proposed a temporal interpolation algorithm to model the movement of human hand gestures. They studied a time span of deciseconds (100msec units or 0.1 of a second). The time span in LUCC datasets may be several years or decades. Non-linear temporal models (e.g., branch, cyclical, and isochronical models) may further complicate temporal scale variance (Jönsson and Eklundh 2004). Therefore, the scale variance in LUCC requires additional investigation before we can apply the computer vision algorithms.

SIFT has attracted considerable research interest among computer vision researchers (Lowe, 2004). SIFT is an algorithm designed to detect, describe, and match key points across images. SIFT points are considered to be invariant to spatial granularity, rotation, affine distortion, translation, and illumination differences. SIFT points are extracted from the derived scale variant images in the scale space, as the minima/maxima of Difference-of-Gaussians (Bundy, and Wallen, 1984). Image matching, clustering, and pattern recognition are then performed by matching SIFT points using a variety of techniques. For example, Majumdar and Ward (2009) applied SIFT for facial recognition by comparing SIFT points with ones calculated

from existing databases. They challenged the discrimination power of SIFT to match SIFT points image-to-image. Yi, Zhiguo, and Yang (2008) improved accuracy in multi-spectral Remote Sensing (RS) image registration by fusing SIFT with spectral information. Their study highlighted the discriminative deficiency of an "off-the-shelf" SIFT, and the necessity of encoding spectral information into SIFT. A SIFT-based building and urban area detection method was proposed by Sirmacek and Unsalan (2009). They employed spatial information (i.e., the shapes of buildings) to improve matching of SIFT points across various IKONOS images. Additions of characteristics (also called features) like discrimination, spectral information, and shape have not yet been explored for SIFT in LUCC.

5.2.1 Similarity of Land Use/Cover Entities

Previous work has highlighted the deficiency of SIFT in distinguishing similar land use/cover entities, which mainly occur in the dense urban areas (Tuermer et al., 2013; Sirmacek and Unsalan, 2009). As an example, in dense urban areas, entities such as those composed of cement (e.g., buildings and roads) can be very similar to each other (Yang et al., 2003). In its default state, SIFT is challenged to adequately discriminate between them. As shown in Figure 5-1, two images are carefully geo-registered but SIFT matching fails to work well due to a lack of uniqueness in SIFT characteristics (e.g., for the corners of roads and buildings).

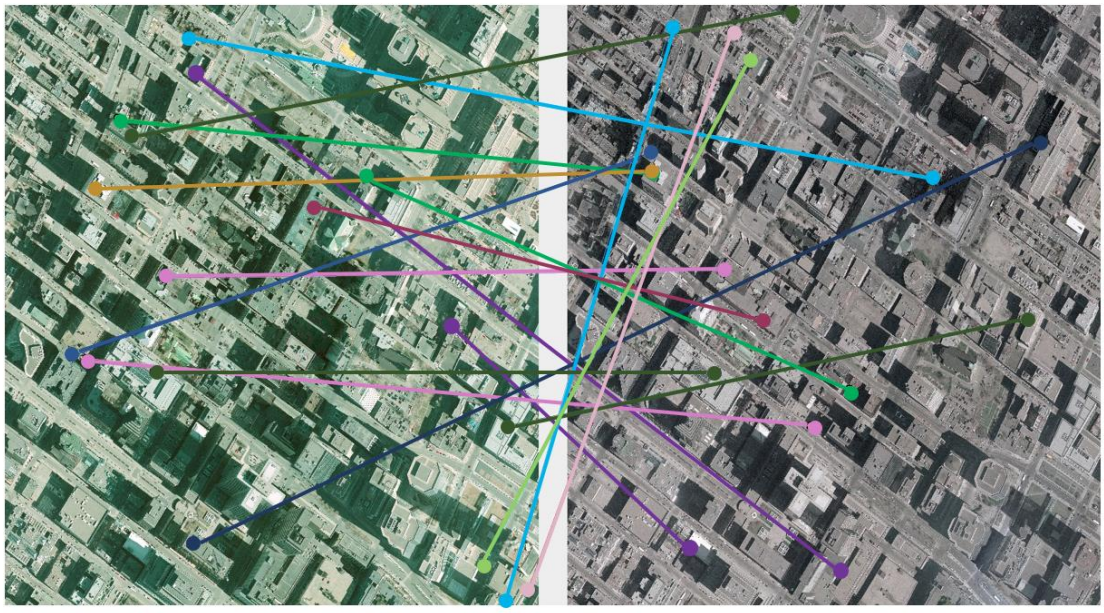


Figure 5-1. SIFT comparison using 10 key points extracted from left (0.11m Montreal Montreal Metropolitan Community Orthophotos / Orthophotographies {MMCO} images recorded at downtown Montreal, 2005 {Communauté métropolitaine de Montréal, 2005}) and right (0.13m MMCO recorded at downtown Montreal, 2007 {Communauté métropolitaine de Montréal, 2007}) respectively. The two images are carefully geo-registered, but seven SIFT mismatches occur because urban structures are very similar to each other. Because SIFT uses 128-bit encoding, multiple pairs can have exactly the same values (illustrated with the same colour). 17 point pairs here are counted 10 SIFT key point pairs (illustrated by different colours).

5.2.2 Use of Shape Information

Region shapes have been found to be sensitive to spatial granularity changes (Luo and Min, 2010). Region shapes are defined by geometric and topological connections among positions and features. SIFT points can be used to compare images directly and mark the clusters of unmatched points as changed regions (Dellinger et al., 2014). Although change information can be represented by individual pixels, our approach encodes regional image features over multiple scales, which is more robust and more useful for LUCC. Regional image features not only provide more information about LUCC (e.g., change boundaries and areas) but also should prove more resistant to noisy information. As shown in Figure 5-2, numerous changed SIFT key points (red points) are caused by the noise or artifacts, such as shadows, vehicles, trees, and building

decorations. Few of these changed points represent actual LUCC. To overcome these issues, we can combine SIFT with regional image features, such as MSER (Matas et al., 2004).

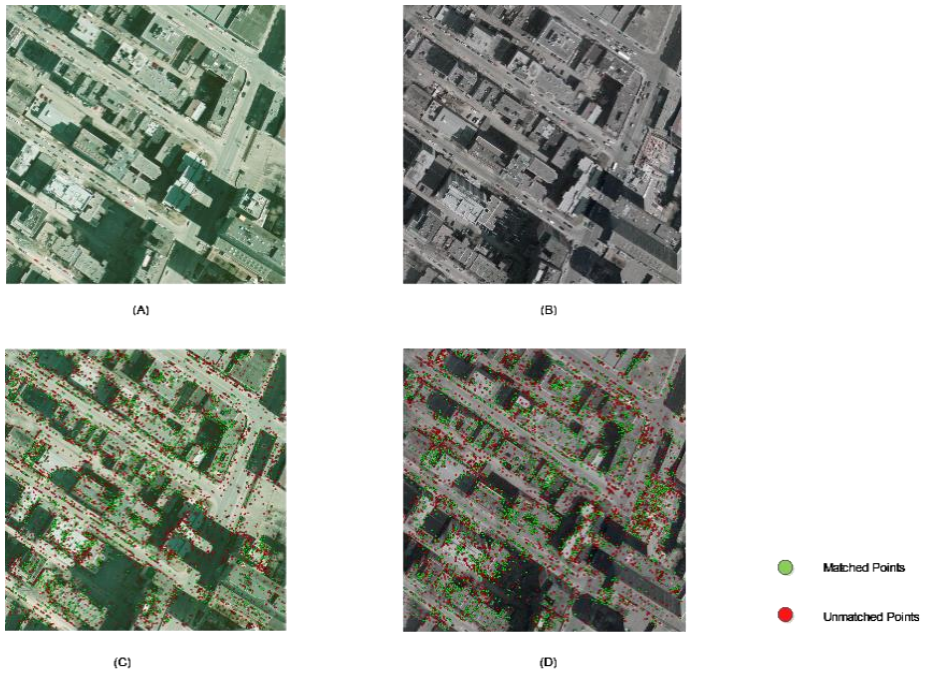


Figure 5-2. SIFT matching-based change detection. (A) upper left corner tile (one ninth) of left image in Figure 5-1; Figure (B) is the corresponding upper left corner tile (one ninth) of right image in Figure 5-1; in (C) and (D), green points stand for the unchanged SIFT key points, and the red ones represent the changed SIFT key points. Matching is implemented with BoofCV using the same parameters as Figure 5-1.

5.2.3 Integration of Spectral Information

SIFT is designed for grey scale images and does not consider spectral information. Since LUCC imagery datasets are acquired by increasing numbers of RS platforms, their spectral channels (bandwidth) can be diverse. Moreover, the spectral information provides recorded values at different wavelengths, which are widely utilized for land use/cover classification and entities recognition (Xu and Gong, 2007). LUCC also benefits from the multispectral and hyperspectral imagery datasets and can be used to further enhance the labeling of LUCC types (Singh, 1989). Default SIFT only considers the image contrast intensity, which may match different image points with similar image intensity.

As shown in Figure 5-1, several point changes among roads and buildings occur since they have very similar intensity values in terms of grey scale. Abdel-Hakim and Farag (2006) pointed out that the original SIFT algorithm provides reasonable geometric distinction for object recognition but its ability to account for spectra is inadequate – by definition. To address this limitation, they modified SIFT to use spectral information, which they argued can enhance SIFT comparison performance for image object recognition. In Figure 5-3 the modified SIFT, or Colour-SIFT (CSIFT), generates a fifteen percent improvement in matching. According to the authors (ibid.), SIFT with spectral information tends to extract more key points than the standard SIFT, which identifies more SIFT key points for LUCC detection and potentially lowers the risk of mismatching.

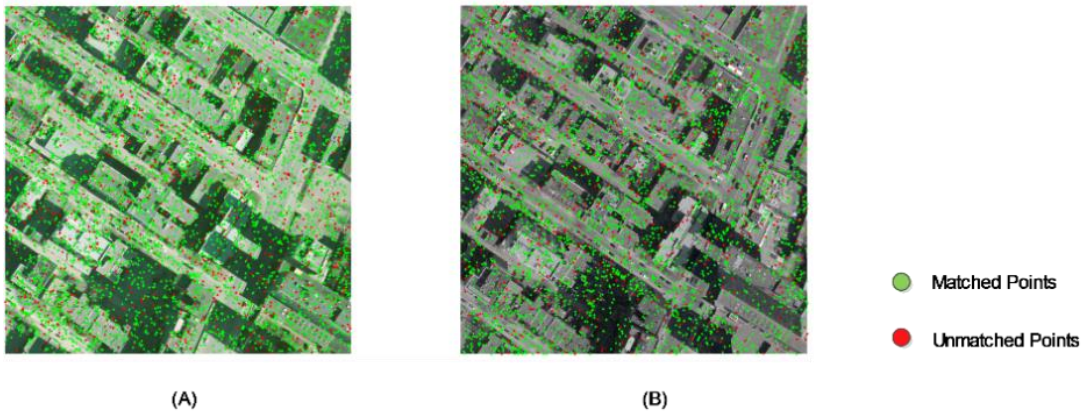


Figure 5-3. Colour SIFT matching. (A) and (B) are the spectral SIFT matching of Figure 5-2 (A) and (B), respectively. I follow Abdel-Hakim and Farag (2006) using the Gaussian colour model for SIFT computation in BoofCV.

5.3. A Scale Invariant LUCC Detection Method

To address the three main challenges of using scale invariant image features in LUCC, we propose a scale invariant LUCC detection method, which compares images of differing spatial scales (i.e., granularity and extent) without altering the original images via interpolation/re-sampling. The proposed method has five steps, each of which is described below. Due to the data size (again, ‘big data”), the first step is to decompose the image into small tiles using a spatial

entropy formulation. Second, MSER extracts scale invariant regions after which we perform a many-to-many matching operation to determine potential changed regions. Third, SIFT points are computed, many-to-many matched, and then combined with change-specific MSERs to detect LUCC regions where changes are not due to scale variance. The above steps can generate noisy LUCC information (e.g., vehicles, trees, and shadows). Consequently, fourth, a change map consistency method was used to smooth and so reduce irrelevant information (e.g., shadows, trees, and vehicles). Finally, a classification algorithm labels the changes in the change map tiles. The workflow of the scale invariant LUCC detection method is illustrated in Figure 5-4.

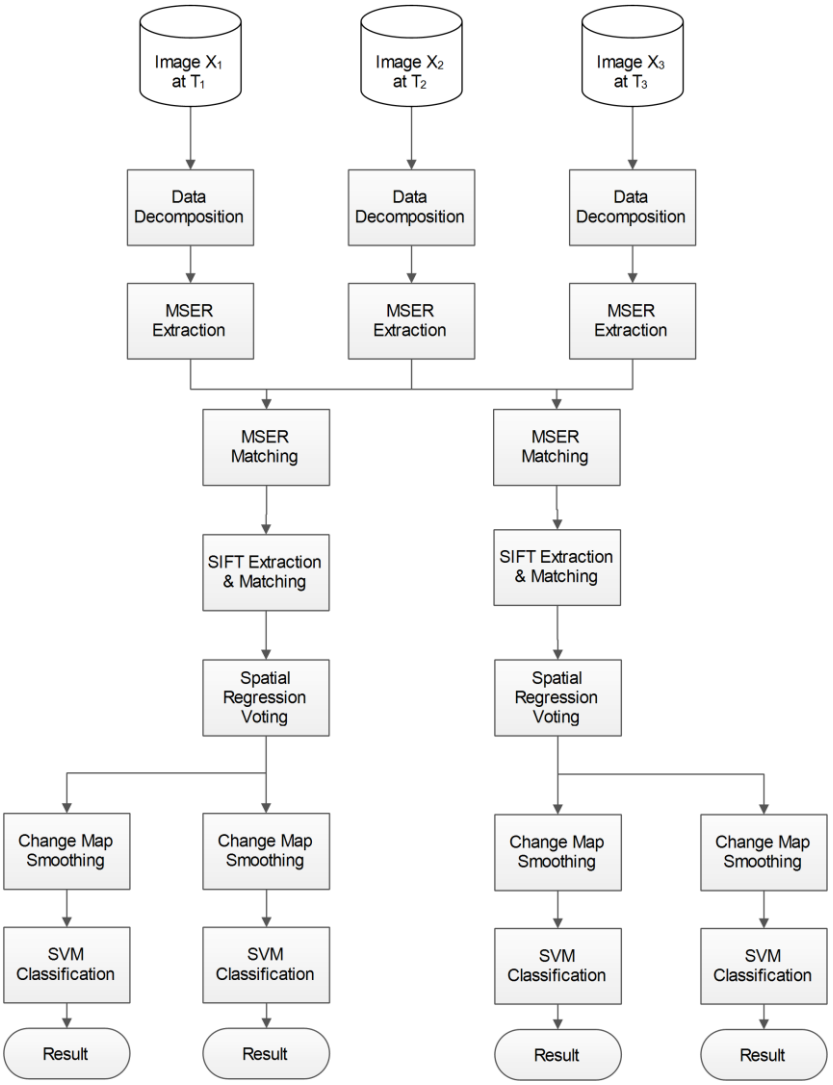


Figure 5-4. Workflow of the scale invariant LUCC detection method.

5.3.1 Data Decomposition

To decompose the large images into smaller tiles, we use an entropy-based splitting method (Tan et al., 2007). The goal of this method is to evenly distribute the data variance across the decomposed image tiles (Uijlings, Smeulders, and Scha, 2009). If we simply decompose big RS imagery data into equal spatial extents then we may extract thousands of small MSERs at a fine granularity and only a few large MSERs from the coarse image. When we compare two images, this will generate thousands of changed regions that do not come from LUCC but from scale

variance. SIFT extraction depends on the pixel value variance—the spatial entropy. Key to this entropy-based image decomposition is to improve matching MSER and SIFT by normalizing numbers of SIFT points and MSERs across decomposed tiles.

The spatial entropy method (Journel and Deutsch, 1993) is shown in equation (5.1). It extends the traditional entropy model, $H = -\sum_{i=1}^{n} P_i \log_2 P_i$, by normalizing the extent relative to resolution (Batty, 1974). This ensures that smaller areas with higher variance will be decomposed similarly to larger areas with lower variance. P_i is the probability that the difference between two adjacent pixels is equal to *i*. Since the spatial granularity (resolution) is set, the extent will be adjusted to guarantee the same spatial entropy *E* among the tiles.

$$E = (-\sum_{i=1}^{n} P_i \log_2 P_i) * (\log_2 \frac{Extent}{Resolution}) \quad (5.1)$$

In practice, it is difficult to achieve a perfect match of *E* image to image, so we create a tolerance parameter called τ, which is the maximum variance between the two entropy scores. Big data brings the fine spatial granularities that are much smaller than the size of the land use/cover entities, but any big data LUCC will invariably split some objects across multiple tiles.

5.3.2 MSER Extraction and Matching

Second, MSER generates regions and then attempts to match them. We have implemented a colour MSER extraction method (Forssén, 2007) which also integrates spectral information into the feature extraction process. The MSERs are generated from *n* iterations of a "growing-and-merging" approach to segment an image tile into clusters of pixels (Zhu and Yuille, 1996). We systematically evaluated different thresholds in each iteration to test if the region boundaries remained stable (i.e., the boundary changes are smaller than the maximum variation value-*MaxVariation*) (Matas et al., 2004). In each iteration, the difference between the thresholds needs

to be larger then a predefined value m_{min}. Regions are considered to be stable MSERs if they contain pixels larger than the minimum (*MinArea)*. We further refine the matching potential with the RANdom SAmple Consensus (RANSAC) (Fischler and Bolles, 1981) algorithm, which is commonly used in combination with MSER (Cheng, et al., 2012). The parameters are usually tuned with training datasets or determined heuristically (Forssén, 2007).

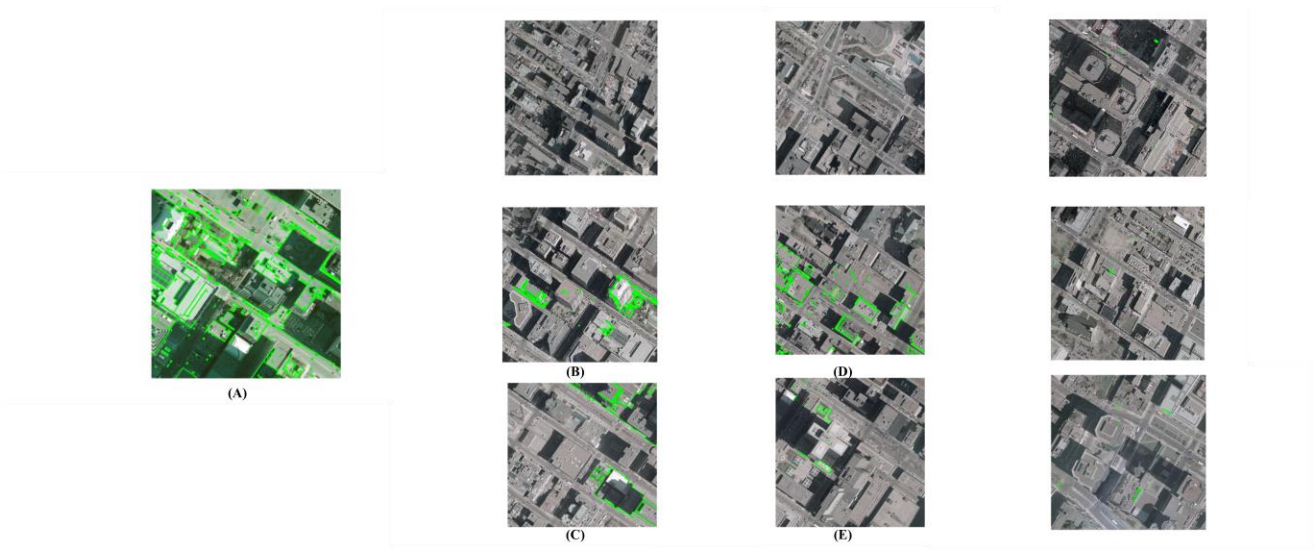


Figure 5-5. MSER matching across image tiles. (A) The unchanged MSERs extracted from Image X_1, 0.11m MMCO image tile acquired in 2005 at downtown Montreal. Figure (B), (C), (D), and (E) are the four of nine coarser granularity tiles with the highest MSER matching scores, using 0.13m MMCO acquired in 2007 (from Image X_2). Unchanged MSER "mask" is depicted with green boundaries.

MSER matching occurs in two steps. First, the thousands of MSERs in the finer granularity set of decomposed images tiles are successively compared to the thousands of MSERs in the coarser granularity set (Figure 5-5). The MSERs in each tile X_1 at T_1 is compared with each set of MSERs in a tile of X_2 at T_2. A likelihood of matching is stored for each MSER comparison. The four highest likely candidates from X_2 are identified. Any unmatched MSERs are preliminarily identified as potential changed regions.

5.3.3 SIFT Change Detection Algorithm

We create a scale invariant method that combines MSER matching with SIFT matching to identify LUCC regions. MSER matching generates candidates but they may contain considerable "noisy" regions that do not represent actual LUCC regions. Relative to one MSER in an image with coarse granularity, a finer granularity image may generate several MSERs at the same geo-referenced location. These MSERs are marked as changed regions because we need to cluster the

fine-granularity MSERs for the matching. SIFT is used inside the changed MSERs to refine LUCC detection. This process is composed of three steps: SIFT extraction, SIFT matching, and spatial regression voting.

The SIFT extraction and feature matching are implemented using the CSIFT (Abdel-Hakim and Farag, 2006) and RANSAC algorithm, respectively. First, the colour invariant gradient orientation histograms are calculated using the Gaussian color model, to generates the CSIFT descriptors (Fritz, Seifert, and Paletta, 2006). Then the RANSAC algorithm refines the Euclidean CSIFT matching.

The spatial regression voting algorithm determines whether changed MSERs represent actual LUCC. This algorithm is inspired by a SIFT voting method proposed by Zamir and Shah (2010). For the *n* changed MSERs $\{M_1, M_2, \dots M_n\}$, the center of gravity for each MSERs, $\{g_1, g, \dots g_n\}$, is calculated. We then separately compute the Euclidean distances from the center of gravity g_i to *p* unchanged SIFT key points and *q* changed points inside the changed MSER M_i. The value of each SIFT key points *S(i)* is defined by:

$$S(i) = \begin{cases} 1, & if\ S(i) \in \{unchanged\ SIFT\} \\ -1, & if\ S(i) \in \{changed\ SIFT\} \end{cases} \tag{5.2}$$

Voting in MSER M_j is expressed as:

$$V(j) = \sum_{i=1}^{p+q} [\frac{D'(j)}{D(i,j)} * S(i)] \tag{5.3}$$

where *D'(j)* stands for the largest distance from the centre of gravity to the edge of MSER *j*. For each M_i, the value of *D'(j)* is constant. *D(i,j)* represents the individual Euclidean distance from

each SIFT point, i , to the center of gravity, g_j. The value of $V(j)$ determines if a changed MSER represents an actual LUCC region ($V(j)<0$) or a false changed region ($V(j)>0$).

The more changed SIFT points there are at the center of a MSER, the more likely a MSER is considered to represent actual LUCC. The center area of MSER tends to be more stable over different levels of thresholding than the edge areas (Forssé and Lowe, 2007). Accordingly, MSER and SIFT are combined for matching to generate change maps. These change maps may contain jagged boundaries, discontinuous edges, isolated changed pixels, and "holes" in the middle of changed areas, which will need to be addressed.

5.3.4 Change Map Smoothing

We employ change map smoothing to remove noisy change information, based on the assumption that LUCC is more likely to occur in connected regions rather than at disjoint points (Ramankutty and Foley, 1999). Change map smoothing also serves to merge the many MSERs we over-generated. For example, we may have numerous tiny grass regions inside one large forest region. Change map smoothing will merge these grass regions into a forest, because the forest occupies the majority of that area.

Change map smoothing is performed here using a Markov Random Field (MRF) grouping-smoothing process (Radke et al., 2005) as follows. According to the Hammersley-Clifford theorem (Frank and Strauss, 1986), the joint probability distribution of any MRF can be written as a Gibbs distribution:

$$P(x) = \frac{1}{Z}\prod_{c\in C} \emptyset_c(x_c) \tag{5.4}$$

where x refers to the particular configuration of the values (intensities) of pixels in the image X_i $\{i=1,2,\ldots,n\}$ (we have n=2 for each image pair comparison) and Z stands for the normalizing

constant. *C* represents all the cliques in the given image. One clique *c* is a group of pixels whose members are mutual neighbours, and $\emptyset_c(x_c)$ is called the clique potential function, which helps define the energy function to be optimized.

The acquired RS image $Y_i(x)$ can be viewed as a combination of a "true" ground image $X_i(x)$ and noise $W_i(x)$:

$$Y_i(x) = X_i(x) + W_i(x) \tag{5.5}$$

Then the noise removal problem can be formulated as the minimization of $W_i(x)$, or $\|Y_i(x) - X_i(x)\|_2^2$ using the Euclidean distance norm. It is widely accepted that $W_i(x)$ follows the Gaussian distribution, so the clique potential function is

$$\emptyset_c(x) = V(x_i) = \exp[-\sum_{i=1}^{m} \frac{\|y_i - x_i\|_2^2}{2\delta^2}] \tag{5.6}$$

δ stands for the deviation of $W_i(x)$, and *m* is the number of pixels. The clique function *V* is presented as

$$V(x_i, x_j) = \gamma \min(\|x_i - x_j\|_2^2, \beta) \tag{5.7}$$

to model the clique neighbourhood, which penalizes the difference between adjacent nodes with threshold β and the weight γ. The total number of possible change map for X_i is $K=2^m$. We define $H_k(x)=1$ to represent change at location *x* in the kth change map ($k \in K$), while $H_k(x)=0$ means no-change at the same location. Given H_k, the change map X_i is encoded as X_i^1 and X_i^0, which represent the change and no-change areas in X_i respectively. The conditional MRF model becomes

$$P(x_i, x_j|H_k) = \frac{1}{Z}\{\frac{\exp[-\sum_{c\in C} V(x_i) - \sum_{c\in C} V(x_j)]}{\sum_{X^1} \exp[-\sum_{c\in C} V(x_i^0, x_j^0)]}\} \quad (5.8)$$

The associated optimization problem as shown in (5.9) results in an optimized change map, with the energy function in (5.10) obtained by merging (5.6) and (5.7) into (5.8).

$$H_k = \arg\{max_{H_l}[\sum_{x_i, x_j \in X} P(y_i|x_i)P(y_j|x_j)P(x_i, x_j|H_l)P(H_l)]\} \quad (5.9)$$

$$E = -log\left\{\sum exp\begin{bmatrix} -\frac{1}{2\delta^2}(y_i - x_i)^T(y_i - x_i) \\ -\frac{1}{2\delta^2}(y_j - x_j)^T(y_j - x_j) \\ -V(x_i, x_j) \end{bmatrix}\right\} \quad (5.10)$$

Since simulated annealing optimization follows naturally from this MRF model (Kasetkasem and Varshney, 2002), it was used to generate the optimized change map. We begin with the original change map. We estimate its initial parameters and set the initial temperature for the simulated annealing. We then obtain a new change map from the previous change map, based on a Gibbs sampling procedure (Gerhard, 1995). Finally, we reduce the temperature with a predetermined schedule and repeat the prior step until there is a convergence or the maximal number of iterations is reached.

Here the temperature is the control parameter of the randomness generator for change area boundaries. More details about this algorithm can be found in the pseudo code in Appendix III. In Figure 5-6, it is easy to notice the MRF-based change map smoothing method removes small vehicles, trees, and shadows. Some large vehicles and shadows still exist after the smoothing. Large shadow areas are difficult to verify without further reference datasets since shadows are very similar to the pavement within RGB colour space. It is possible to add other smoothing algorithms to remove large vehicles and shadows but that runs the risk of eliminating

actual LUCC. Removing shadows and vehicles with minimum impact on LUCC in dense urban area remains an ongoing challenge (Yin et al., 2015).

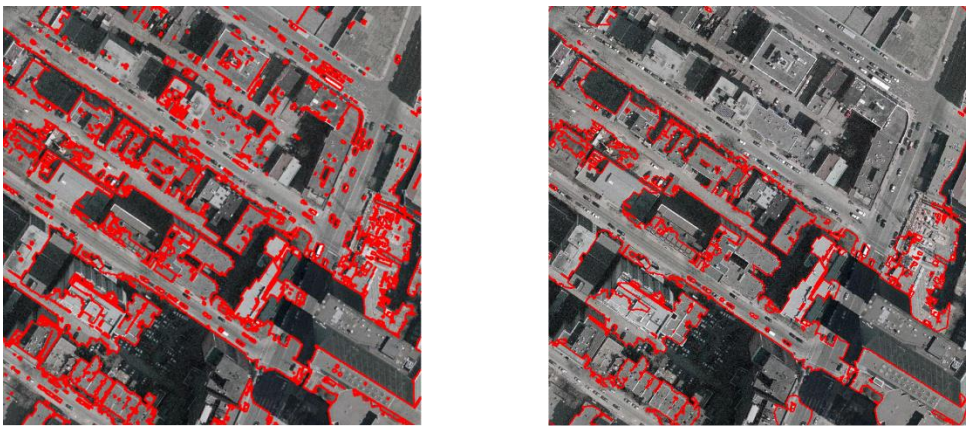


Figure 5-6. Change map smoothing. (A) The change map generated by MSER and SIFT matching, by comparing 2005 0.11m MMCO and 2007 0.13m MMCO collected at downtown Montreal, and overlaid with the MMCO image tile in 2007; (B) The change map after the MRF-based map smoothing process. We note some large vehicles and shadows still exist after smoothing.

5.3.5 LUCC Labelling

Labelling of LUCC is always challenging as it requires coordination between spatial and temporal scales. These methods require significant training data and continuous landscape monitoring. In the following empirical study, RS data within the Greater Montreal Area from 2005-2012 were collected and the images were acquired in early July to avoid seasonal differences. Consistent with practice, standard land use or land cover labels are used (Ridd and Liu, 1998). For image classification, a Support Vector Machine (SVM) classifier is selected due to its high accuracy and low sensitivity to noisy data in RS research (Melgani and Bruzzone, 2004). There are seven labels for the SVM classification: forest, grass, farmland, bare ground, water, roads and buildings. A subset of raw images is used for classifier training and then applied

to the rest of the imagery datasets. Finally, ground truth data is employed to evaluate the accuracy of the scale invariant LUCC detection method.

5.4. Case Study in Montreal LUCC

The scale invariant LUCC detection method was evaluated with the urban-rural LUCC detection in the Greater Montreal Area, Quebec, Canada, covering up to 4,163 km^2 from 2005-2012. Details about the scale heterogeneous data were listed in Table 5-1. The 2005 MMCO data covered most urban areas in the City of Montreal. To obtain a seamless image for 2007, we used MMCO for the most areas of Montreal city and surroundings at 0.13m spatial granularity; some suburban and rural areas of the Greater Montreal Area were acquired at 0.3m spatial granularity. The computing resource provisioning was supplied by a hybrid cloud composed of one local controller (Intel® Core™ i7-6700 Processor, 32GB memory, and 2TB storage) and Virtual Machines (VMs) from the Microsoft Azure cloud computing platform. Four Azure Hadoop clusters were utilized for four cross-scale LUCC processes, with each cluster consisting of six VMs. Most of the code was developed in Java, based on Hadoop, BoofCV, and OpenIMAJ libraries. The detailed implementation of the workflow was illustrated in Figure 5-7.

The scale invariant LUCC detection method was designed for image pair comparisons so there were four separate LUCC comparisons 2005-2006, 2006-2007, 2007-2009, and 2009-2012. Azure D3_V2 VM was chosen for the LUCC 2005-2006 process (4 cores and 14GB memory). Both 2006-2007 and 2007-2009 processes utilized six D5_V2 VMs (16 cores and 56GB memory). The 2009-2012 comparison was deployed on six D4_V2 nodes (8 cores and 28GB memory). The VM configurations were determined by the trade-off between the computing workload and costs (Zhu and Agrawal, 2010). Five hundred GB Azure online file storage (100GB for each year; 0.13m data was selected for most areas, and 0.3m data for the other areas,

for comparison of the two 2007 datasets with different granularities) was utilized for the datasets in Table 1[2]. All the raw datasets were geo-referenced; consistent with computer vision, the tiles were not.

For each LUCC comparison, the scale invariant LUCC detection workflow was deployed as five *MapReduce* steps. The first *map* step extracted the MSERs; whereas the MSER matching was conducted as a *reduce* step (Section 5.3.2). The second *map* step extracted SIFT; whereas the *reduce* computation removed "artificial" SIFT features (e.g., artificial SIFT features can be caused by tile borders, as the artificial border challenge in {Xing et al., 2014}). The third *map* step deployed the SIFT matching and the spatial regression voting algorithm (see Appendix II for the pseudo code for the spatial regression voting). The fourth *map* step handled change map smoothing (Section 5.3.4). There was no reduce steps for the third and fourth *MapReduce* process. The final *map* step scheduled the SVM classification (Section 5.3.5) and recombined the distributed results and output the final results to the local controller in its *reduce* step.

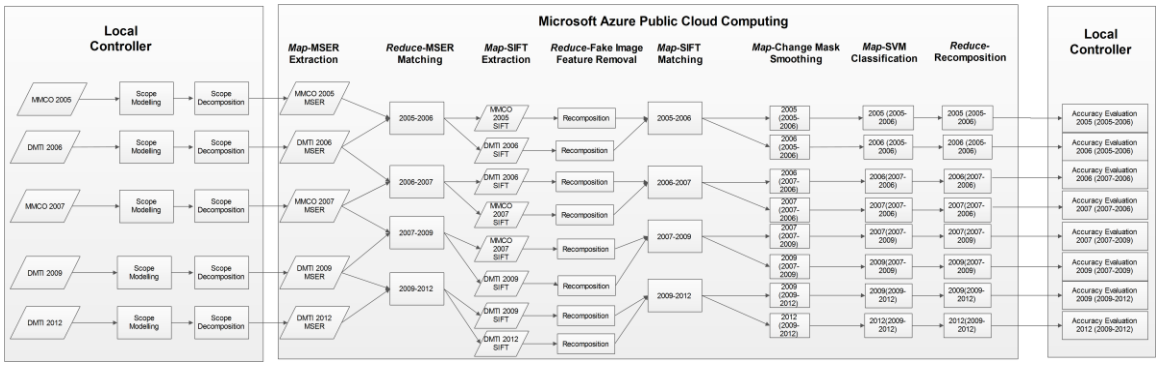


Figure 5-7. Implementation of the scale invariant LUCC workflow for our Montreal urban-rural LUCC case study.

[2] Although the scale invariant LUCC method is designed to solve the big data challenges in LUCC, the data in our case study is not so big (~500GB), due to the limited availability of high-resolution RS imagery data at the Schulich Library of McGill University.

Table 5-1. Details about datasets used in the Montreal urban-rural LUCC

Year	Platform	Spatial Resolution (m)	Spectral Resolution	Spatial Extent (km2)	Number of Image Files
2005	Montreal Metropolitan Community Orthophotos	0.11	RGB (sharpened & fused)	75.02	62
2006	DMTI	0.60	RGB (sharpened & fused)	3528.00	50
2007	Montreal Metropolitan Community Orthophotos	0.13 / 0.30	RGB (sharpened & fused)	3718.75 / 139.73	2380/18
2009	DMTI	0.60	RGB (sharpened & fused)	2257.92	32
2012	DMTI	0.60	RGB (sharpened & fused)	4163.04	59

The entropy-based spatial decomposition can be illustrated using MMCO 2005 and DMTI 2006 datasets. Each MMCO image file covered approximately 1.21 km^2 area; whereas the DMTI image covered nearly 70.56 km^2. The average of the first part of *E*, ($H = -\sum_{i=1}^{n} P_i \log_2 P_i$) of the entropy in (5.1) was 7.51 and 6.74 for MMCO 2005 and DMTI 2006,

respectively. Then $|E_1 - E_2| < \tau$ became $|7.51 * log_2 extent^1 - 6.74 * log_2 extent^2 + 18.64| < 10$. We chose $\tau = 10$. If H was the same for any image pair then the pixel difference between image tiles would be no more than 1024 (32*32), which appeared to provide a satisfactory tolerance for creating tile pairs that generated similar number of SIFT points and MSERs. We obtained 0.24 km^2 and 1.88km^2, as the extent of the decomposed image tiles, respectively. The spatial decomposition is depicted in Figure 8. Since the number of tiles must be an integer, 2*2 and 6*6 decomposition were selected in Figure 8 as the closest solution.

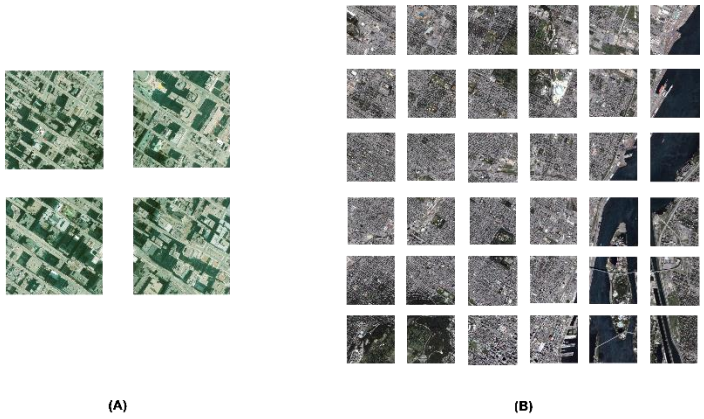


Figure 5-8. The spatial entropy-based spatial decomposition. (A) 2*2 splitting of MMCO imagery data; and (B) 6*6 decomposition of DMTI dataset.

To generate a larger number of smaller MSERs, the *MinArea* was set to 10 and *MaxVariation* equaled 0.2 for the MSER extraction in Section 5.3.2. Following Forssén (2007), the m_{min} parameter was set to 0.003 and the step parameter *n* was heuristically set to 200 (because we prefer more iterations with smaller threshold difference for more but smaller MSERs). We favor generating a large number of small MSERs as opposed to a smaller number but bigger MSERs. This reduces the risk of missing LUCC, but may result in more noise. After MSERs were extracted from the decomposed tiles, we implemented MSER matching for the potential changed MSER identification. The MSER extraction and matching are implemented

with OpenIMAJ library (Hare, Samangooei, and Dupplaw, 2011). The pseudo code for MSER matching is listed in Appendix I.

Tile comparison was a many-to-many process, which would create a problem in serial processing but *MapReduce* implementation turned it into a parallel one-to-many matching, based on the <*key, value*> data structure. Separate change maps were favored over fused change maps that can depict LUCC at a finer granularity but conceal the scale variance. Consequently there were two versions of change maps for the year 2006, 2007, and 2009, generated from the different LUCC comparisons.

For both MSER and SIFT matching, the Euclidean distance ratio method was adopted for initial matching, with 1.5 as the threshold. RANSAC was then implemented by fitting a geometric model— an affine transform model was chosen —to the initial results (Pereira and Pun, 2000) since the matches of image features could be invariant to translations, rotations and scale changes. This process iteratively selected a random set of matches, estimated the geometric model from the selected random set and then tested the remaining matches against the learned model – always eliminating outliers. The method looped until the size of matches reached below 50 percent of the initial match. The RANSAC matching was based on the OpenIMAJ library. The change map smoothing algorithm in Section 5.3.4 removed noisy change information (see Appendix III for the pseudo code). The parameters in the above steps were determined using the sampling strategy from the four LUCC processes. In each comparison, five image pairs (without decomposition) were sampled to calculate the parameters. The classification was developed on libSVM (Chang and Lin, 2011) with seven pre-defined labels in Section 5.3.5. The SVM training process was conducted according to Melgani and Bruzzone (2004) with image features of

MSERs, using brightness, shape, and texture. The SVM classifier training processes employed the same five image pairs.

The results of the LUCC comparisons were verified with 1000 ground truth points. We collected 650 points with purposive sampling in high density areas (i.e., downtown and rapidly developing areas like the City of Laval) because we wanted to verify our method in problem areas (e.g., tall buildings with long shadows and road repaving). The other 350 points were collected via random sampling by gridding the Great Montreal Area. All 1000 points were physically inspected. We note the oversampling of problems likely negatively impacted the accuracy compared with a completely random sampling. The accuracy of the LUCC is shown in Table 5-2.

Table 5-2. LUCC Accuracy Evaluation with Ground-Truthing

		Ground-Truthing			Total Accuracy (%)
		Change (%)	NoChange (%)	Total (%)	
2005-2006 LUCC	Change (%)	6.6	36.1	42.8	
	NoChange (%)	2.1	55.8	57.1	62.4
	Total (%)	8.7	91.3	100.0	
2007-2006 LUCC	Change (%)	4.3	24.6	28.9	
	NoChange (%)	6.6	64.6	71.1	67.9
	Total (%)	10.9	89.2	100.0	
2007-2009 LUCC	Change (%)	12.1	21.3	33.5	
	NoChange (%)	5.8	60.7	66.6	72.9
	Total (%)	17.9	82.0	100.0	
2009-2012 LUCC	Change (%)	3.0	5.7	8.7	
	NoChange (%)	9.2	82.1	91.2	85.1
	Total (%)	12.3	87.7	100.0	

Our lowest overall accuracy occurs in the 2005-2006 LUCC comparison, at 62.4 percent. Most error derives from false change areas (36.1%). Numerous regions in the MMCO 2005 are designated as changed since they fail to match regions in DMTI 2006. Our method can handle scale variance, with some constraints. Subtle view angle differences caused some building windows to be visible only in images at higher resolutions, which generated several false changes. High resolution also renders noisy information (e.g., vehicles, trees, and water on the roads) much harder to remove, which otherwise can be removed with change map smoothing in lower resolution images. Most false changes were found in dense urban areas. Our method improves its accuracy in rural and suburban areas, with a greater than five times granularity difference between the MMCO 2005 and DMTI 2006 datasets. To further improve the accuracy, possible solutions could entail resampling the results to the same granularities, or utilizing image fusion techniques (Li, Manjunath, and Mitra, 1995) to homogenize the scale of the results.

The highest accuracy is achieved from the 2009-2012 comparison because the two datasets are recorded with the same spatial resolution and sensing platform. For the 2007-2006 and 2007-2009 LUCC comparisons, the average accuracy is 68.9 percent and 72.8 percent, respectively. The small difference between the accuracies can be explained by the larger spatial extents covered by the DMTI 2006 dataset. The reason for accuracy differences, we believe, not only lies in the scale variance (i.e., spatial granularities and extents) of data, but also in smaller differences in view angle, shadow, vehicle, trees, and water areas of MMCO and DMTI data. The coarser spatial resolution of DMTI data reduces shadow and vehicle noise to some extent. It is important to remember that high resolution imagery datasets do not guarantee high accuracy in LUCC as high resolution inevitably generates more diverse and noisier information.

We compare our total accuracy to other LUCC studies. An MSER and SURF-based LUCC detection achieved approximately 75 percent total accuracy (Ye et al. 2014). They utilized resampling to hide the scale heterogeneity of 200 aerial photos. Their accuracy resembles our 2007-2009 comparison but our method did not require preprocessing the data with resampling. Raja et al. (2013) achieved 82.5 percent accuracy in their scale-variant LUCC study (IRS-1B at 72.5m compared to IRS-P6 at 5.8m) but they also employed resampling. Pham, Mercier, and Michel (2016) reported 85 percent total accuracy, using a SIFT matching and graph-based LUCC detection method. They conducted their test using a pair of 800 × 400-pixel radar images, each at the same (10m) resolution. Their result was similar to our 2009-2012 comparison. This suggests that our scale-invariant LUCC detection method can handle the scale heterogeneity directly and still achieve good results.

Although portions of our empirical study do not generate very high accuracy, we argue that the scale-invariant LUCC detection method can more effectively extract LUCC from scale heterogeneous datasets without image scaling techniques (e.g., spatial interpolation, down-sampling, and image resizing). Image scaling techniques invariably assign pixel values that are consistent with their neighbours, which increases the risk of missing LUCC (Dai and Khorram, 1998). Again, large scale differences (both granularity and extent) will lower the LUCC detection accuracy. The scale-invariant LUCC detection method can reduce the influence of scale variance in LUCC detection but not eliminate it.

5.5. Conclusion

This paper presents the promises and challenges of handling scale variance in LUCC and proposes a scale invariant LUCC detection method. Our method integrates extent, shape, and spectral information into scale invariant image features, as a workflow composed of entropy

decomposition, MSER, SIFT, spatial regression voting, change map smoothing and LUCC labelling. The method is deployed in a cloud computing and Hadoop framework to address the scale variance challenges in big data. We note the drawbacks of our method. First, not all LUCC regions can be extracted as MSERs (e.g., construction sites and the road repair works) when these regions cover small spatial extents (for us, less than *minArea*) and are unstable across different levels of intensity thresholds. Second, some features (e.g., road re-pavement with darker colors in the image) are difficult to distinguish from shadows, due to similarities in shape and spectral attributes. Third, noisy objects with well-defined borders and sharp contrasts from their neighbouring objects (e.g., large vehicles) produce unmatched MSERs with unmatched SIFT points. As these are not LUCC, the overall accuracy is decreased. These drawbacks worsen as the image granularity difference increases (Haverkamp and Poulsen, 2003).

Big data has significantly changed LUCC research, not only in terms of data management and processing, but also on spatial-temporal scales. Short time period changes can be captured by advances in sensing platforms (e.g., the temporary construction sites). We assume that modelling of both spatial and temporal variance in LUCC will focus increasingly on temporal analysis. Our own research will investigate methods that integrate spatial and temporal variance to build consistent spatial-temporal models.

5.6. Acknowledgement

This research has been funded by a Microsoft Azure Research Award and the Social Sciences and Humanities Research Council, Canada (SSHRC).

5.7. References

Abdel-Hakim, A. E., & Farag, A. A. (2006). CSIFT: A SIFT descriptor with color invariant characteristics. In Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on (Vol. 2, pp. 1978-1983).

Batty, M. (1974). Geospatial entropy. Geographical analysis, 6(1), 1-31.

BoofCV. (2017). Available online: http:// http://boofcv.org/

Bundy, A., & Wallen, L. (1984). Difference of gaussians. In Catalogue of Artificial Intelligence Tools (pp. 30-30). Springer Berlin Heidelberg.

Celik, T. (2009). Multiscale change detection in multitemporal satellite images. IEEE Geoscience and Remote Sensing Letters 6(4), 820-824.

Chang, C. C., & Lin, C. J. (2011). LIBSVM: a library for support vector machines. ACM Transactions on Intelligent Systems and Technology 2(3), 27.

Chen, G., Hay, G. J., Carvalho, L. M., & Wulder, M. A. (2012). Object-based change detection. International Journal of Remote Sensing 33(14), 4434–4457.

Cheng, L., Li, M., Liu, Y., Cai, W., Chen, Y., & Yang, K. (2012). Remote sensing image matching by integrating affine invariant feature extraction and RANSAC. Computers & Electrical Engineering, 38(4), 1023-1032.

Communauté métropolitaine de Montréal, 2005, 2007. Montreal Metropolitan Community Orthophotos / Orthophotographies de la Communauté métropolitaine de Montréal, 2005 and 2007.

Dai, X., & Khorram, S. (1998). The effects of image misregistration on the accuracy of remotely sensed change detection. IEEE Transactions on Geoscience and Remote sensing, 36(5), 1566-1577.

Dellinger, F., Delon, J., Gousseau, Y., Michel, J., & Tupin, F. (2014). Change detection for high resolution satellite images, based on SIFT descriptors and an a contrario approach. In 2014 IEEE Geoscience and Remote Sensing Symposium (pp. 1281-1284).

DMTI, 2006, 2009, 2012. Montreal Satellite StreetView, 3_1- 9_7_MONTREAL-S3XM, Markham ON: DMTI Spatial Inc., 2006, 2009 and 2012.

Fischler, M. A., & Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM 24(6), 381-395.

Forssén, P. E. (2007). Maximally stable colour regions for recognition and matching. In Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on (pp. 1-8).

Forssén, P. E., & Lowe, D. G. (2007). Shape descriptors for maximally stable extremal regions. In Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on (pp. 1-8).

Frank, O., & Strauss, D. (1986). Markov graphs. Journal of the American Statistical Association 81(395), 832-842.

Fritz, G., Seifert, C., & Paletta, L. (2006). A mobile vision system for urban detection with informative local descriptors. In Computer Vision Systems, 2006 ICVS'06. IEEE International Conference on (pp. 30-30).

Gerhard, W. (1995). Image Analysis, Random Fields, and Dynamic Monte Carlo Methods: A Mathematical Introduction. New York: Springer-Verlag.

Goodchild, M. F. (2011). Scale in GIS: An overview. Geomorphology 130(1), 5-9.

Haghani, A., Hamedi, M., Sadabadi, K., Young, S., & Tarnoff, P. (2010). Data collection of freeway travel time ground truth with bluetooth sensors. Transportation Research Record: Journal of the Transportation Research Board, (2160), 60-68.

Hampton, S. E., Strasser, C. A., Tewksbury, J. J., Gram, W. K., Budden, A. E., Batcheller, A. L., ... & Porter, J. H. (2013). Big data and the future of ecology. Frontiers in Ecology and the Environment, 11(3), 156-162.

Hare, J. S., Samangooei, S., & Dupplaw, D. P. (2011). OpenIMAJ and ImageTerrier: Java libraries and tools for scalable multimedia analysis and indexing of images. In Proceedings of the 19th ACM international conference on Multimedia (pp. 691-694).

Huo, C., Zhou, Z., Liu, Q., Cheng, J., Lu, H., & Chen, K. (2008). Urban change detection based on local features and multiscale fusion. In IGARSS 2008-2008 IEEE International Geoscience and Remote Sensing Symposium (Vol. 3, pp. III-1236).

Jönsson, P., & Eklundh, L. (2004). TIMESAT—a program for analyzing time-series of satellite sensor data. Computers & Geosciences, 30(8), 833-845.

Journel, A. G., & Deutsch, C. V. (1993). Entropy and spatial disorder. Mathematical Geology, 25(3), 329-355.

Kasetkasem, T., & Varshney, P. K. (2002). An image change detection algorithm based on Markov random field models. IEEE Transactions on Geoscience and Remote Sensing 40(8), 1815-1823.

Kwok, W., & Sun, H. (1993). Multi-directional interpolation for spatial error concealment. IEEE Transactions on consumer electronics, 39(3), 455-460.

Li, H., Manjunath, B. S., & Mitra, S. K. (1995). Multisensor image fusion using the wavelet transform. Graphical models and image processing, 57(3), 235-245.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60(2), 91-110.

Luo, R., & Min, H. (2010, June). Multi-scale maximally stable extremal regions for object recognition. In Information and Automation (ICIA), 2010 IEEE International Conference on (pp. 1799-1803). IEEE.

Majumdar, A., & Ward, R. K. (2009). Discriminative SIFT features for face recognition. In Electrical and Computer Engineering, 2009. CCECE'09. Canadian Conference on (pp. 27-30).

Matas, J., Chum, O., Urban, M., & Pajdla, T. (2004). Robust wide-baseline stereo from maximally stable extremal regions. Image and Vision Computing 22(10), 761-767.

Merhav, N., & Bhaskaran, V. (1996). A transform domain approach to spatial domain image scaling. In Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference on (Vol. 4, pp. 2403-2406). IEEE.

Melgani, F., & Bruzzone, L. (2004). Classification of hyperspectral remote sensing images with support vector machines. IEEE Transactions on Geoscience and Remote Sensing 42(8), 1778-1790.

Miller, H. J., & Goodchild, M. F. (2015). Data-driven geography. GeoJournal 80(4), 449-461.

Ohn-Bar, E., & Trivedi, M. M. (2014). Hand gesture recognition in real time for automotive interfaces: A multimodal vision-based approach and evaluations. IEEE transactions on intelligent transportation systems, 15(6), 2368-2377.

Olofsson, P., Foody, G. M., Herold, M., Stehman, S. V., Woodcock, C. E., & Wulder, M. A. (2014). Good practices for estimating area and assessing accuracy of land change. Remote Sensing of Environment, 148, 42-57.

Perona, P., & Malik, J. (1990). Scale-space and edge detection using anisotropic diffusion. IEEE Transactions on pattern analysis and machine intelligence, 12(7), 629-639.

Pereira, S., & Pun, T. (2000). Robust template matching for affine resistant image watermarks. IEEE transactions on image Processing, 9(6), 1123-1129.

Pham, M. T., Mercier, G., & Michel, J. (2016). Change detection between SAR images using a pointwise approach and graph theory. IEEE Transactions on Geoscience and Remote Sensing, 54(4), 2020-2032.

Prashanth, H. S., Shashidhara, H. L., & KN, B. M. (2009). Image scaling comparison using universal image quality index. In Advances in Computing, Control, & Telecommunication Technologies, 2009. ACT'09. International Conference on (pp. 859-863). IEEE.

Radke, R. J., Andra, S., Al-Kofahi, O., & Roysam, B. (2005). Image change detection algorithms: a systematic survey. IEEE Transactions on Image Processing 14(3), 294-307.

Raja, R.A.A., Anand, V., Kumar, A.S., Maithani, S., Kumar, V.A. (2013). Wavelet Based Post Classification Change Detection Technique for Urban Growth Monitoring. Journal of the Indian Society of Remote Sensing 41, 35–43.

Ramankutty, N., & Foley, J. A. (1999). Estimating historical changes in global land cover: Croplands from 1700 to 1992. Global biogeochemical cycles, 13(4), 997-1027.

Ridd, M. K., & Liu, J. (1998). A comparison of four algorithms for change detection in an urban environment. Remote sensing of Environment 63(2), 95-100.

Sheikh, H. R., & Bovik, A. C. (2006). Image information and visual quality. IEEE Transactions on image processing, 15(2), 430-444.

Singh, A. (1989). Review article digital change detection techniques using remotely-sensed data. International journal of remote sensing, 10(6), 989-1003.

Sirmacek, B., & Unsalan, C. (2009). Urban-area and building detection using SIFT keypoints and graph theory. IEEE Transactions on Geoscience and Remote Sensing 47(4), 1156-1167.

Tan, C. P., Koay, J. Y., Lim, K. S., Ewe, H. T., & Chuah, H. T. (2007). Classification of multi-temporal SAR images for rice crops using combined entropy decomposition and support vector machine technique. Progress In Electromagnetics Research, 71, 19-39.

Tuermer, S., Kurz, F., Reinartz, P., & Stilla, U. (2013). Airborne vehicle detection in dense urban areas using HoG features and disparity maps. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 6(6), 2327-2337

Uijlings, J. R., Smeulders, A. W., & Scha, R. J. (2009). What is the spatial extent of an object? In Computer Vision and Pattern Recognition. IEEE Conference on (pp. 770-777). IEEE.

Witkin, A. (1984). Scale-space filtering: A new approach to multi-scale description. In Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'84. (Vol. 9, pp. 150-153).

Woodcock, C. E., & Strahler, A. H. (1987). The factor of scale in remote sensing. Remote sensing of Environment, 21(3), 311-332.

Xing, J., Sieber, R., & Kalacska, M. (2014). The challenges of image segmentation in big remotely sensed imagery data. Annals of GIS, 20(4), 233-244.

Xu, B., & Gong, P. (2007). Land-use/land-cover classification with multispectral and hyperspectral EO-1 data. Photogrammetric Engineering & Remote Sensing, 73(8), 955-965.

Yang, L., Xian, G., Klaver, J. M., & Deal, B. (2003). Urban land-cover change detection through sub-pixel imperviousness mapping using remotely sensed data. Photogrammetric Engineering & Remote Sensing 69(9), 1003-1010.

Ye, S., Nourzad, S. H. H., Pradhan, A., Bartoli, I., & Kontsos, A. (2014). Automated Detection of Damaged Areas after Hurricane Sandy using Aerial Color Images. In Computing in Civil and Building Engineering (2014) (pp. 1796-1803).

Yi, Z., Zhiguo, C., & Yang, X. (2008). Multi-spectral remote image registration based on SIFT. Electronics Letters 44(2), 1.

Yin, D., Du, S., Wang, S., & Guo, Z. (2015). A direction-guided ant colony optimization method for extraction of urban road information from very-high-resolution images. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 8(10), 4785-4794.

Zamir, A. R., & Shah, M. (2010). Accurate image localization based on google maps street view. In European Conference on Computer Vision (pp. 255-268). Springer Berlin Heidelberg.

Zhu, Q., & Agrawal, G. (2010). Resource provisioning with budget constraints for adaptive applications in cloud environments. In Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing (pp. 304-307). ACM.

Zhu, S. C., & Yuille, A. (1996). Region competition: Unifying snakes, region growing, and Bayes/MDL for multiband image segmentation. IEEE transactions on pattern analysis and machine intelligence, 18(9), 884-900.

Appendix I: MSER Matching in Recomposition

Algorithm: MSER Matching

Input: the image lists *A* and *B*, the MSER lists *MA* and *MB*, and *threshold* for MSER matching

Output: the list *L* containing the correspondence between *A* and *B*

for each MSER *ma* in *MA*:
  S=zeroes(size(*B*))
  for each MSER *mb* in *MB*:
    s_i=match(m*a.score*,*mb.score*)
    S.add(s_i)
  end for
  S'=descend_sort(*S*)
  S'=sub_list(*S'*,1,4)
  for *i*=1:4
    if(s_i'<=*threshold*)
    S'.remove(*i*)
    end if
  end for
  L.add(*S'*)
end for
return *L*

End

Appendix II: SIFT Change Detection within Voting

Algorithm: SIFT Change Detection with Voting

Input: image tile *I*, unmatched MSER list *UM* for I, and SIFT list *S* for I

Output: the MSER change region list *C*

for each unmatched MSER *u* in *UM*:

u.gravity_center=$(\frac{\max(u.x)+\min(u.x)}{2}, \frac{\max(u.y)+\min(u.y)}{2})$

u.score=Equation (5.3) in Section 5.3.3 using *S* and *u.gravity_center*

if (*u.score*<=0)

C.add(u);

end if

end for

return *C*

End

Appendix III: Change Map Smoothing

Algorithm: Change Map Smoothing

Input: initial image change map *C*, and the original imagery dataset *D*.

Output: smoothed image change map *C*

```
for i = 1 to Max_Iteration
  T = T0/log(1 + i)
    for k =1 to Max_k
      for m = 1 to Max_m
        if(C(k,m) = =0)
        E0= Equation (5.10) in Section 5.3.4
        else
        E1= Equation (5.10) in Section 5.3.4
        end if
        P0 = exp(-E0/T)
        P1 = exp(-E1/T)
        P0 = P0*(P0 + P1)
        R= rand(0,1)
        if (R < P0)
            C(k,m) = 0
        else
            C(k,m) = 1
        end if
      end for
    end for
end for
return C
```

End

Connecting Statement: Integrating the LUCC workflow with Geospatial CyberInfrastructure

Chapter 5 has presented the scale invariant change detection algorithm for LUCC. This method identifies LUCC by comparing scale invariant image features that are extracted separately from scale heterogeneous images. However, the average accuracy of this method is abut seventy-two percent, lower than most LUCC analysis using scale homogeneous data. To improve the accuracy, different spatial optimization techniques are proposed. On the other hand, the temporal models are not integrated to delineate the change trajectories, and most of the LUCC trajectory generation rely on the temporal optimization. Therefore, optimization becomes a necessary component in LUCC especially with GCI.

Chapter 6 formulates LUCC as a spatial-temporal optimization problem. First, the change/no-change areas are modelled as the spatial-temporal atoms using the spatial-temporal object model, and these atoms either change completely or remain the same thorough the study time spans. Second, boundaries of the spatial-temporal atoms are optimized via the branch-and-mincut algorithm. Third, the MapReduce distributed computing model is replaced by Apache Storm, a graph-based parallel computing framework, to avoid unnecessary waiting time in GCI. Finally, all these optimization methods are integrated within GCI. The LUCC evaluation study in the Greater Montreal Area proves the GCI-based optimization method can achieve high accuracy around 90%.

Chapter 6 has been published on the *International Journal of Geographic Information Science*, 2016. The manuscript contained in this chapter was co-authored with my supervisor, Prof. Renée Sieber. I am the primary author and contributed the GCI-based optimization framework and corresponding implementation in the cloud computing for LUCC research. Prof.

Sieber introduced the spatial-temporal object model into the optimization framework and improved the readability of this article.

Chapter 6. A Land Use/Land Cover Change Geospatial CyberInfrastructure to Integrate Big Data and Temporal Topology

Abstract

Big data has shifted spatial optimization from a purely computational-intensive problem to a data-intensive challenge. This is especially the case for spatio-temporal Land Use /Land Cover Change (LUCC) research. In addition to greater variety, for example from sensing platforms, big data offers datasets at higher spatial and temporal resolutions; these new offerings require new methods to optimize data handling and analysis.

We propose a LUCC-based Geospatial CyberInfrastructure (GCI) that optimizes big data handling and analysis, in this case with raster data. The GCI provides three levels of optimization. First, we employ spatial optimization with graph-based image segmentation. Second, we propose ST Atom Model to temporally optimize the image segments for LUCC. Finally, the first two domain spatio-temporal optimization is supported by the computational optimization for big data analysis. The evaluation is conducted using DMTI (DMTI Spatial Inc.) Satellite Streetview imagery datasets acquired for the Greater Montreal area, Canada in 2006, 2009, and 2012 (534 GB, 60cm spatial resolution, RGB image). Our LUCC-based GCI builds an optimization bridge among LUCC, spatio-temporal modelling, and big data.

Keywords: LUCC; Geospatial CyberInfrastructure; Optimization; Spatio-Temporal Object Model.

6.1 Introduction

Geographic Information Science (GIScience) and Remote Sensing (RS) research into big data has been triggered by increasing spatial, spectral, and temporal resolutions of sensing systems and Web 2.0 platforms (McAfee *et al*., 2012). That is, we simply have magnitudes' larger

volumes of data, which are arriving at increasing velocity, and with greater variety in data structures. Big data has forced a rethinking of numerous aspects of GIScience, from spatial data collection and storage to sampling, analysis, and visualization (Hampton *et al*., 2013; Liang *et al.* 2010; Zaslavsky *et al*. 2013). Big data also requires a new architecture for managing those methods.

Big data has the potential to shift research on detection of Land Use/ Land Cover Changes (LUCC). The field of LUCC has been explored for over 50 years (Singh, 1989). LUCC detection addresses three questions: (1) Is there any change of interest when comparing two or more temporally distanced datasets?; (2) What are these changes quantitatively?; and (3) What are the change trajectories and corresponding rates? Because big data affects spatial and temporal domains simultaneously, it impacts all three questions. Higher volumes and velocity may allow us to detect finer grained changes that may have been missed with datasets at coarser spatial resolution and temporal periodicity. Techniques in quantitative detection of LUCC should enable multi-temporal analysis (comparison of more than two raster datasets) and handle heterogeneity in spatial, spectral and temporal resolutions. Big data dramatically increases the number of potential changed objects (since more objects can be extracted from higher spatial and spectral resolutions and more object changes can be detected with improved temporal resolutions). Big data promises greater LUCC but renders the trajectories of those changes—time—more difficult to delineate.

Additionally, big data poses significant computational challenges, such as the need for scalable data storage, flexible computing resource provisioning, and dynamic workflow management. Solutions to these challenges should be integrated with the domain demands of LUCC to increase accuracy of results and shorten computation time.

In this paper, we propose a LUCC-based Geospatial CyberInfrastructure (GCI) that seeks to optimize the Spatio-Temporal (ST) handling and analysis of big data. This optimization is three-fold. A domain-based layer handles spatial optimization through energy cost minimizations of pixel clustering into feature objects. Because LUCC in big data likely requires handling multiple time slices, our GCI temporally optimizes via what we call a ST Atom Model. Third, our GCI optimizes computing resource provisioning, data decomposition, and workflow. Figure 6-1 shows how these optimizations function in our GCI.

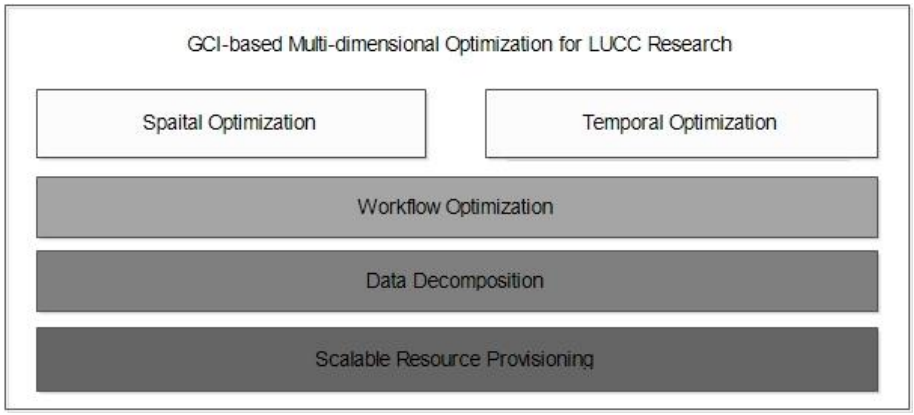


Figure 6-1. GCI-based Multi-dimensional Optimization for LUCC Research

The paper is organized as follows. In Section 6-2, we discuss research to date on LUCC and spatio-temporal modelling. We describe our LUCC-based GCI, which attempts to optimize along space, time, and data handling in Section 6-3. We deploy and evaluate the optimization methods in Section 6-4. We conclude with opportunities for future research.

6.2 LUCC and ST Optimization

In this section, we review the related works about employing spatial optimization in LUCC. Then we delineate the needs of using temporal information to optimize an object-based LUCC. We also discuss literature on computational optimization within GCI for LUCC support.

6.2.1 Optimization Challenges in LUCC

Spatial optimization has been studied in LUCC for a long time (Tong and Murray, 2012). Jenerette and Wu (2001) utilize spatial optimization to simulate the LUCC in the central Arizona – Phoenix region of U.S., while Ligmann-Zielinska *et al.* (2008) optimize generative models for land-use allocation. Most of these works focus on spatial optimization at a given time, but temporal information has rarely been integrated into the optimization process.

A traditional approach in LUCC for identifying change has been pixel-based. The pixel-based approach relies on pixel-level calculation to generate a "difference image" (e.g., the subtraction of two images) to identify the relative amounts of change. This approach is frequently utilized for bi-temporal analysis, and generally requires imagery datasets to match in spatial and spectral resolutions (Singh, 1989). Reviews of LUCC (Singh, 1989; Lu *et al*., 2004; Jianya *et al*., 2008) have revealed a gradual shift in research from pixel-based to object-based approaches, which work with groups of pixels as objects. In part, this is because increasing spatial resolutions afforded by big data allow for pixels that are significantly smaller than objects of interest. Object-based approaches have the additional advantage of moving us beyond traditional raster-vector divides that separate RS from GIScience.

Blaschke (2010) terms the shift to objects as Object Based Image Analysis (OBIA). OBIA groups similar image pixels as objects, calculates object features, and then applies classification algorithms (Congalton, 1991) to label various types of changes. Walter (2004) argues that OBIA is less sensitive than pixel-based analyses to different spatial and spectral resolutions of datasets because comparing object properties (e.g., texture and shape) can identify change. According to Tong and Murray (2012), OBIA is a type of district optimization, which spatially optimize the change areas for multi-temporal LUCC. Also, because pixels are grouped

into and then conceptually handled as objects via spatial optimization (Baatz and Schäpe, 2000), OBIA attempts to represent some degree of spatial topology in each original image with these objects.

OBIA has its drawbacks. First, if parts of the object change from time t_1 to t_2 then OBIA may mark the entire resultant object as "changed". Figure 6-2 illustrates this issue. Second, the temporal topology is not retained in the OBIA process. The lack of temporal topology change information impedes integration of OBIA into multi-temporal LUCC, which tracks the impact of one object's change on its neighbours over several periods (Pijanowski *et al*., 2002). If we choose OBIA then we should find techniques that optimize the integration of change information in both spatial and temporal dimensions, including the between-time topology.

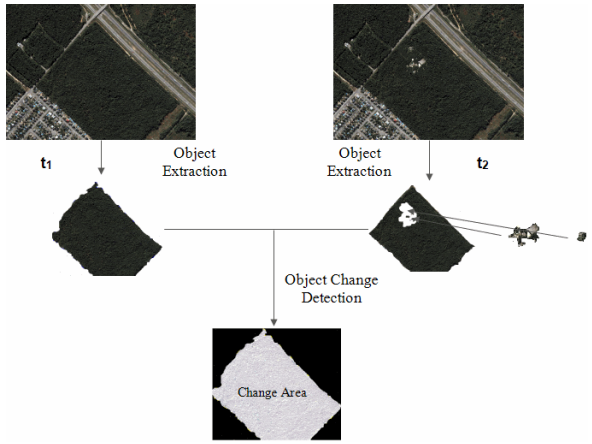


Figure 6-2. Example of drawbacks of OBIA Change Detection. Fine grained changes occurring at t2 will fail to be recorded when compared to change area at t1. Samples are extracted from Montreal Streetview satellite images 2006 and 2009 (DMTI Inc., 2006 and 2009).

Figure 6-2 illustrates the problems in OBIA with two images of Montreal, Canada, at 2009 (t_1) and 2012 (t_2). We show sample objects generated from clustering: at t_1, a forest object, and at t_2, a forest-"donut hole" object and a building object. The building object compared to the spatial extent of the forest object t_1 results in the whole object being labelled as "changed". From Figure 6-2, we can see only using spatial optimization for image segmentation cannot solve the

partial object challenge, so we need to use the temporal information to further optimize the OBIA.

Before we arrive at objects, we need to group pixels according to their similarities. This is called image segmentation and follows three general types: spatial segmentation, feature-based clustering, and graph-based methods. Spatial segmentation extracts regional entities from imagery datasets based on the spatial structure information, while feature-based clustering algorithms rely on similarities of image features to group pixels. Graph-based image segmentation method combines elements of spatial- and feature- based image segmentation methods (Shi and Malik, 2000). The core idea of the graph based method lies in constructing a weighted graph, where each vertex represents pixels (regions) in the image and the weight of each edge connecting two pixels represents the likelihood of segmentation. The weight, which is usually calculated by combining feature and spatial information, forms a cost energy function. Minimization of the energy cost is considered a traditional spatial optimization, where the image is cut into several segments (Tong and Murray, 2012). Graph-based segmentation algorithms have been studied extensively for object extraction (Sumengen and Manjunath, 2006; Jermyn and Ishikawa, 2001; Wu and Leahy, 1993).

The spatial optimization found in energy cost minimization suffers from difficulties in determining initial values and is easy to trap with the local optimal solutions (Celik and Yetgin, 2011). Various parametric learning and optimization approaches have been applied with graph-based segmentation methods to address these problems. For example, parametric maxflow (Gallo *et al.*, 1989) integrates non-local features into the optimization process; Kolmogorov *et al.* (2007) present case studies using maxflow approach. Lempitsky *et al.* (2012) propose a global optimization method called “branch-and-mincut” for graph-based image segmentation with

segmentation mask and non-local parameters. To handle multi-label image segmentation, Boykov and Funka-Lea (2006) supply α-expansion and αβ-swap-move-based algorithms.

Building the graph and the minimization of the energy cost function are highly computation-intensive, and may take a long processing time. Since most research projects have time constraints, it is very tempting to consider High Performance Computing (HPC). With big data, this process becomes both data-intensive and computationally-intensive, so it becomes more difficult to consider this optimization process separately from the computation. We conclude that optimization in big data LUCC requires a combination of spatial, temporal, and computational optimization methods.

6.2.2 ST Modelling and Temporal Optimization in LUCC

Different ST models have been integrated with LUCC (Radke *et al.*, 2005). These include statistical distribution modelling of the change and non-change areas (Bazi *et al*., 2005), predictive models (Veldkamp and Lambin, 2001), and cellular automata simulations (Li and Yeh, 2002). These applications advance temporal and spatial components, but not the two components equally (Deng *et al*., 2009; Pan *et al*., 1999).

Over the years, GIScience researchers have proposed various methods to effectively and elegantly integrate temporal and spatial dynamics. Yuan (1996) provided the first survey of ST models, which illustrated their pros and cons in representing LUCC. Abraham and Roddick (1999) surveyed the most widely utilized ST database systems. More recently, Nandal (2013) reviewed ST models, and categorized them into ten types: snapshot model (Armstrong 1988), space-time composite data model (Langran and Chrisman, 1988), data models based on simple time-stamping (Allen, 1991), event-oriented model (Peuquet and Duan, 1995), three domain model (Yuan, 1999), history graph model (Van Der Wal and Pye, 2003), Spatio-Temporal

Entity-Relationship (STER) model (Parent *et al*., 1999), Object-Relationship (O-R) model (Coppin *et al.*, 2004), ST object model (Huang and Chandramouli, 2009), and moving object data model (Erwig et al., 1999). Multiple ST models may be combined, such as the Hybrid ST Data Model which merges the event oriented model and space-time composite data model (Sengupta and Yan, 2004).

A predominant reason why so many models have been created is that it is difficult to determine how best to model and store the changes. For example, missing state information creates difficulty in applying event or process-based temporal modelling (i.e., event-oriented model, O-R model, STER Model, and moving object data model). A challenge in applying, for example, the space-time composites model occurs when attempting to compare imagery datasets with heterogeneous resolutions, which prevents the direct overlay of temporal snapshots of land surface (Nadi and Delavar, 2003). Likewise, the difficulty of extracting semantics from imagery datasets impedes the employment of the three domain model. Most ST models are vector based, at least in their deployment; whereas, RS imagery datasets are generally raster data. ST models need to provide interfaces to ease the vectorization process. The simple time stamping and the history graph model method present difficulties in vectorizing RS datasets. ST models, we argue, pose a significant optimization problem that will only get worse with big data.

6.2.3 GCI Related to LUCC

GCIs have been designed to handle challenges found in big data research and in computation-intensive jobs found in GIScience and RS (Wang, 2010). For example, Liang *et al.* (2010) used GCI to enable sharing and visualization of big environmental sensing datasets. Yue *et al.* (2010) proposed a semantic web based GCI to provide on-demand RS big data products. A

GCI was also developed to perform data mining from volunteered geographic information harvested over the Internet (Gao *et al.*, 2014).

GCI can provide the integration of domain specific optimization and computing techniques for GIScience. Consequently GCI research is intertwined with specific hardware and software platforms for distributed data handling. An example of this intertwining is Xia *et al.*'s (2010) hardware solution—a Compute Unified Device Architecture (CUDA) based GCI to accelerate inverse distance weighting and viewshed analysis. CUDA exacts a cost in host-device data transfer, which cannot be neglected in large volume transfers (Yang *et al*., 2008).

Compared to hardware (e.g., CUDA, grid computing), Yang *et al*. (2011) conclude that cloud computing affords the best platform for geospatial big data. Specific cloud solutions include Google's development of MapReduce, which is a software platform to distribute computing tasks over multiple machines. Hadoop, an open source implementation of MapReduce, is highlighted by Yang *et al.* (2010) for its capacity to process big spatial data.

Hadoop is the preferred choice for GCIs due to its scalability and flexibility (Nurian et al., 2012). Nonetheless, it has problems. Lee *et al.* (2012) highlight the weakness of dataflow management in MapReduce. They also note the low input-output efficiency of MapReduce. Some researchers have begun to explore data streaming, which is defined as a continuous sequence of datasets. Researchers have implemented data streaming to analyze radar datasets (Plale *et al*., 2006). Another study utilized data streaming for environmental observation analysis in cluster computing (Tilak *et al*., 2007). Neither study calls their work GCI; however, they resemble GCIs in that geospatial analysis is conducted with distributed computing environment and the emphasis is on the underlying architecture.

GCIs have not been widely applied for big data analysis in LUCC. First, domain specific optimization challenges in LUCC require scalable and flexible computing resource provisioning. Second, the massive data exchanges among different computation process which shape LUCC much more complicated than a collection of batch processing. Third, a LUCC workflow also needs to be optimized for better data transfer and less computation time. Therefore, the optimization of computing resource provisioning and workflow management need to be twisted together with LUCC studies. In this paper, we propose LUCC-based GCI, to provide the integrated GCI-based optimization.

6.3 LUCC-based GCI

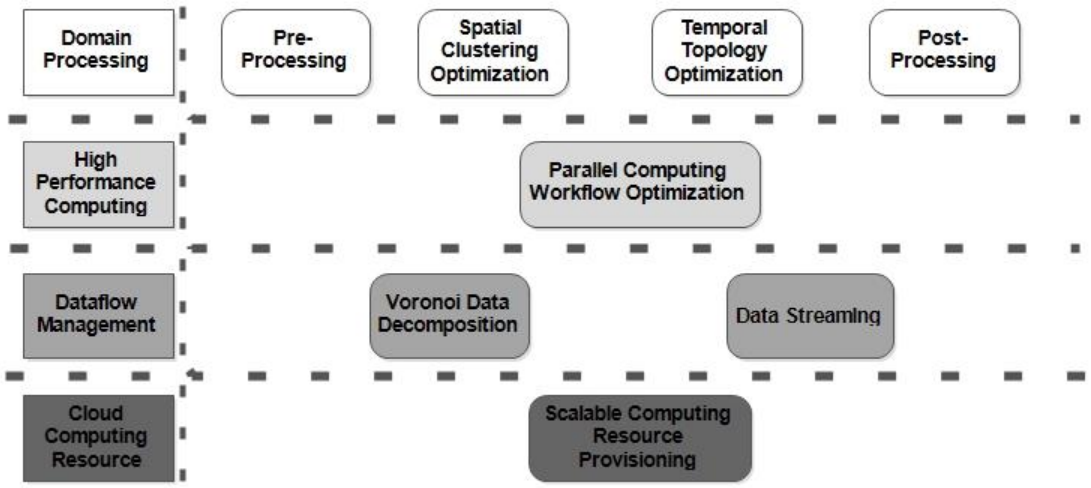


Figure 6-3. LUCC-based GCI and the Multi-dimensional Optimization

Figure 6-3 illustrates the architecture of our LUCC-based GCI. This GCI provides the integration of domain specific optimization methods with computation optimization techniques. Specifically, our spatial and temporal optimization methods extract what we call ST atoms from multi-temporal images to detect any changes, where the ST atoms stand for image pixel groups that either remain or completely change across the time span. This ST atom model is similar to

Worboys' (2005) ST atom concept, which stands for the homogeneous areas have constant properties over space and time. Our ST atom is designed to study change/no-change and has higher tolerance on property difference. The whole process is supported by data streaming, Voronoi image decomposition, workflow optimization and scalable cloud computing resources. In the era of big data, we argue that we should consider optimization as a combination of domain knowledge and computation (Wang, 2010). Otherwise, excessively long processing times and errors incurred, for example by "oversplitting" of big data due to insufficient understanding of a domain like LUCC, can hinder the knowledge discovery in big data.

6.3.1 Optimization in the Domain Layer

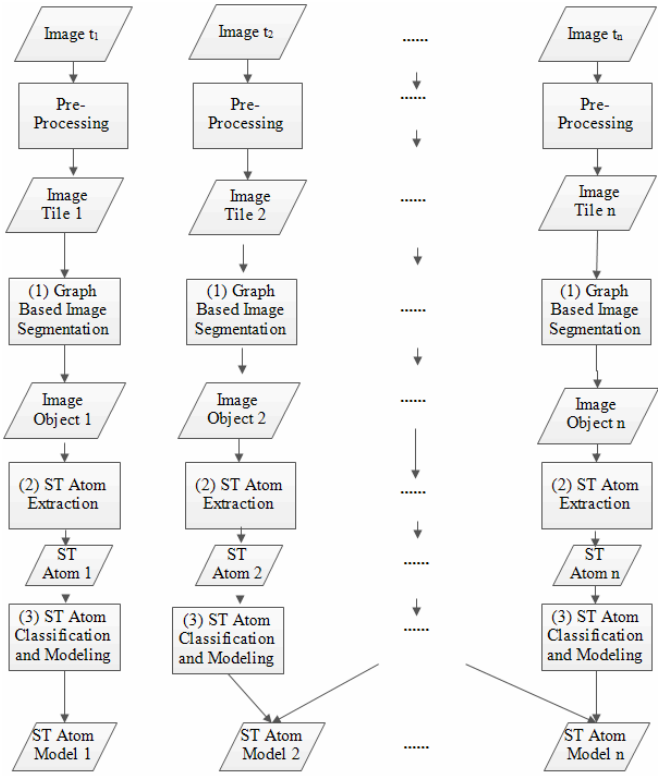


Figure 6-4. Workflow of our LUCC Framework, with input/output illustration of each step.

Figure 6-4 shows the workflow of our domain layer, focusing on our optimization methods. First, we implement a graph-based image segmentation process to extract objects with spatial and spectral similarities from multi-temporal RS imagery datasets. Second, we use temporal topology rules to find the ST atoms with the image segments. Finally, we generate the change trajectories based on the ST Atom Models by applying classification and ST object modelling.

6.3.1.1 Graph Based Image Segmentation with Spatial Optimization

We use $\mathbf{X}_{t1}=\{x_{1,1}, x_{1,2}, \ldots, x_{I,J}\}$to denote an image that is recorded in time t_1 with I×J pixels and *b* bands. $\mathbf{X}_{t1}$ is modelled as an undirected graph ***G: (V, E)***, where the pixel in spatial position ***(i,j)*** is linked with a vertex $v_{i,j} \in \boldsymbol{V}$, and $e_{i,j;t,u} \in \boldsymbol{E}$ is the edge that connects $v_{i,j}$ to its neighboring pixel $v_{t,u}$. In this paper, we consider the neighbouring system N as a 4 connected grid, which consists of ordered pixel pairs ($x_{i,j}$, $x_{t,u}$). We introduce $\boldsymbol{L}=\{1, 2, \ldots, \boldsymbol{K}\}$ and K labels. K labels are defined for the given image, for multi-object segmentation. Let $f = \{f_{v_{i,j}} | v_{i,j} \epsilon \boldsymbol{X_{t1}}\}$ ($f_{v_{i,j}} \epsilon \boldsymbol{L}$) be the collection of all the pixel-label assignment. The spatial optimization is found via an energy cost function. The energy function of our graph based image segmentation is formulated, according to Boykov *et al.* (2001), as:

$$E(f) = \lambda \sum_{v_{i,j} \epsilon \boldsymbol{V}} D(f_{v_{i,j}}) + \sum_{v_{i,j}, v_{t,u} \epsilon N} V(f_{v_{i,j}}, f_{t,u}) \quad (6.1)$$

The term $D(f_{v_{i,j}})$ is called the data term and $V(f_{v_{i,j}}, f_{v_{t,u}})$ is named the smoothness term. $D(f_{v_{i,j}})$ represents the cost of assigning label $f_{v_{i,j}}$ to pixel $v_{i,j}$; whereas $V(f_{v_{i,j}}, f_{v_{t,u}})$ penalizes spatial inconsistency and tends to assign the same label to neighbouring pixels. Minimizing $E(f)$ will optimize segmentation in the image graph. A solution to the multi-labelling problem is achieved by the α-expansion algorithm (Boykov *et al*., 2001). Given a labelling f and a label α, a

move from f to f^{α} is called an α-expansion if $f_{v_{i,j}} \neq f^{\alpha}_{v_{i,j}} \rightarrow f^{\alpha}_{v_{i,j}} = \alpha$. The α-expansion algorithm iterates over all labels α to find the best α-expansion until convergence. The drawback is that α-expansion might trap the local optima. The trapping problem means that a local optima (either maximum or minimum) within a neighbouring set of candidate solutions is mistakenly considered as the global optima of all the candidates. To overcome this problem, we utilize the global optimization branch-and-mincut algorithm (Lempitsky *et al*., 2012).

To achieve the minimal value in the energy cost function, we use the branch-and-mincut spatial optimization method. This method tends to find the global optimal solution by a top-down search in the feature space, which is organized as a binary tree. This technique is built on top of graph cut and branch-and-bound algorithm (Quesada and Grossmann, 1992). Lempitsky *et al.* (2012) has proved its effectiveness in different image segmentation studies.

We turn to the data term and smoothness term in equation (6-1). It is quite difficult to calculate the data term $D\left(f_{v_{i,j}}\right) = -\log \Pr(f_{v_{i,j}}|F_{v_{i,j}})$ directly, where $F_{v_{i,j}}$is the observed geometric feature vector for pixel $v_{i,j}$. Liu *et al*. (2008) propose a super-pixel and SVM (Supporter Vector Machine) classification-based method to approximate the distribution of $\Pr(f_{v_{i,j}}|F_{v_{i,j}})$. We adopt Liu *et al*.'s (*ibid*.) approach in this paper.

$V(f_x, f_y)$ reflects the weight among pixels in the graph. The weight exists to penalize assigning different labels to adjacent pixels. The RBF (Radial Basis Function) kernel (Camps-Valls *et al.*, 2008) is used for its (relative) simplicity:

$$\omega_{i,j;t,u} = e^{\frac{-\left\|I_{i,j}-I_{t,u}\right\|_2^2}{2\sigma^2 dist(I_{i,j},I_{t,u})}} \tag{6.2}$$

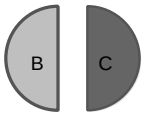
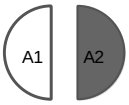
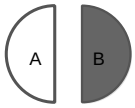
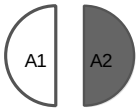
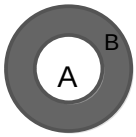
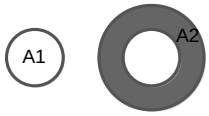
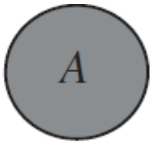
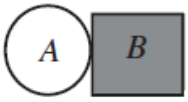
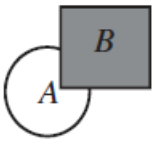
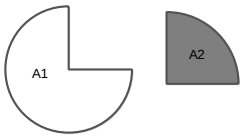
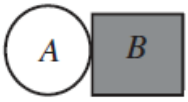
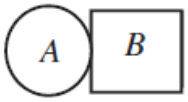
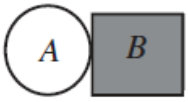
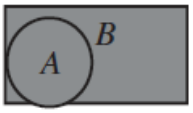
where $I_{i,j}$ is the intensity of $x_{i,j}$, dist $(I_{i,j}, I_{t,u})$ stands for the spatial distance, and σ is the Gaussian width. We illustrate image segmentation using RS image $\mathbf{X}_{t1}$. The equations will be similar when comparing multiple images. Using the graph-based optimization method, we can obtain a collection of image segmentations for multi-temporal RS imagery datasets.

6.3.1.2 Spatio-Temporal Atom Extraction and Labelling

We now need to extract ST atoms using temporal optimization, which guarantees the ST atoms either remain the same or completely change through the study time span. The ST Atom Model is proposed to handle the partial object changes in Figure 6-2, and amend the lack of temporal topology in LUCC. To implement the ST atoms in LUCC we use the largest homogeneous units that hold their spatial and temporal features. We find that the object concept in ST model and OBIA can most easily be applied to the ST atoms. ST atoms can be viewed as hybrid vector-objects, which become the lingua franca between OBIA and the ST object model in LUCC.

For pre-processing, we need to first "snap" the image segments into objects. The image segmentation boundary pixels obtained in Section 6.3.1.1 might be discrete and out of order. For example, the boundary of image segments may not form neat lines and may contain numerous small variations. To facilitate the employment of ST atom extraction, we utilize a chain-code (Li *et al*., 1995) to connect the boundary pixels and then we implement the Douglas-Peucker algorithm to turn the region boundary into polygons (Saalfeld, 1999). These processing steps turn the image segments into objects. Then object matching is conducted using coordinates of image registration to find the corresponding objects in the other time periods.

Table 6-1. Temporal Topology Rules Chosen for ST Atoms Extraction

Temporal Topology Relationship	Study Time t_1	Change Reference Time t_2	Explanation	ST Object Atom
Equal			The object A has changed completely to another object in t2.	
Split			Object A has changed to two or more objects in t_2.	
Partial Change			One or more parts of object A has changed to another objects in t_2.	
Contain			One or more changes happen within object A in t_2.	
Expand			Although object A become larger in t_2, the ST object atom is till its original size.	
Shrink			Object A become smaller in t_2.	
Overlap			Objects B changes its position and overlaps with A in t_2.	
Merge			The adjacent object or parts have changed to A in t_2.	
Covered-By			Object A is totally covered by its neighbor object B in t_2.	

The most important part of our temporal optimization is retaining the temporal topology. The temporal optimization process can be defined as finding the largest ST atoms with the constraints of temporal topology rules. Egenhofer and Al-Taha (1992), and Müller and Zeshen (1992) provide a set of spatial rules that explain spatial relations among objects (Table 6-1). Concepts of "equal", "contain", "split", "overlap", "merge", and "covered-by" rules are useful to model the interactions between two ST atoms with similar spatial extents. (Other topology change rules, like, "disjoint" in Egenhofer and Al-Taha (1992) are not included as they have little impact on the ST atom extraction.) In addition to considering the spatial distance between objects (Egenhofer and Al-Taha, 1992), we highlight the temporal topological changes from t_1 to t_2. We introduce the state of "partial change" to describe situations in which Object A cannot be considered having completely changed into one or more objects. An example of this would be parts of a roof on a hypothetical Building A that are re-painted. The "expand" and "shrink" topology change rules describe changes that involve all neighboring objects (Müller and Zeshen, 1992). The key idea is to keep ST atoms as the largest temporally homogeneous object. An object is split into ST atoms, which will be entirely changed or unchanged across the LUCC study time span. There is no partially changed ST atom after the employment of these rules. We borrow the concept of vector objects in vector analysis and apply it to RS image analysis for LUCC detection. Therefore, ST atom could be viewed as a combination of vector and raster object.

Nine rules for bi-temporal topology change are listed. More complicated topologies can be represented by combining two or more of these nine basic rules. For example, we can obtain three objects from Figure 6-2 t_1 by simply applying the "contain" rule twice with a bi-temporal image pair. The ST atom extraction also can be parallelized to fit the distributed computing

environment in our LUCC-based GCI. The ST atom extraction process thus becomes the temporal optimization process for OBIA.

We first implement change masks. Change masks serve as templates for any interpolation, for results of temporal rule application, and for difference images that will then be overlaid onto the original objects to extract the atoms. Image interpolation is applied if the two candidate objects do not have the same spatial resolution (Lam, 1983). We perform interpolation at this stage, instead of at the pre-processing stage, because the graph based image segmentation and classification parameters are very sensitive to the noise generated by overall image interpolation (Lempitsky *et al*., 2012). Then we utilize the univariate image differencing technique (Singh, 1989) to generate the "difference image". By employing *k-means* clustering algorithms (Rui and Turi, 1999) and a thresholding technique (Lu *et al*., 2004), the difference image is clustered as several change areas. Temporal topology rules are applied to each change mask, which are overlaid onto the original objects to generate ST atoms. Atom extraction processes are applied in an iterative bi-temporal way. Each time we apply one entry in Table 1 to extract ST atoms with the bi-temporal image pairs. An iterative application of temporal topology rules transforms the objects into ST atoms (more details in Section 6.4).

The final process employs a classification algorithm, where the ST atoms are given labels to represent the actual LUCC types (e.g., forest, buildings, roads, and grassland). Because we use OBIA, the label is not ascribed to individual pixels but to ST atoms. A broad range of classification algorithms can be used to generate the labels (Walter, 2004). We use the SVM classification algorithm due to its high classification accuracy and low sensitivity to noisy data in RS analysis (Melgani and Bruzzone, 2004).

We create a ST LUCC optimization that can be widely applied with RS imagery datasets that possess a high level of variety. The increased variety of big data suggests that LUCC must be adjusted according to imagery datasets with heterogeneous spectral and spatial resolution. To some extent, our OBIA with optimization technique eliminates spatial and spectral heterogeneity (Chen *et al*., 2010); whereas temporal topology rules addresses the temporal heterogeneity. We also note if the spatial and spectral resolutions are very different (e.g., one with 0.6m spatial resolution and the other one with 600m), our LUCC optimization may fail.

6.3.2 HPC and Workflow Management Optimization

Workflow management is a large portion of any GCI. The LUCC-based GCI dataflow management layer partitions the big datasets as shown in Figure 6-4. We use the Voronoi diagram and Fortune's Sweepline algorithm as described by Xing and Sieber (2014) to better decompose large datasets. This method provides rough load balancing and minimizes the influence of the splitting borders in LUCC. It serves as a way to optimize splitting tiles so that features are retained as much as possible.

Our Voronoi-based partitioning method uses data streams afforded by Apache Storm, which also serves the HPC. Storm is a free and open source software project of the Apache Software Foundation. Storm relies on data streaming as part of real-time job scheduling to improve parallel computing support for big data analysis. Data streams can be fed into any number of processing nodes with minimum couplings in parallel. Storm characterizes streams as an unbounded sequence of data tuples. Data tuples may contain image tiles, object or ST atoms. Tuples are continuously pushed into processing nodes in parallel. Storm manages the network of these data streaming communications, which is called *topology* management. The Storm

framework is the development tool to build the data stream topology and provide fault-tolerance functionalities in our LUCC-based GCI.

The optimization of LUCC detection computation can be formulated as the minimization of computation time with respect to limited computing resource. Graph image segmentation and ST atom extraction illustrate how Storm, the software, and data streaming, the concept, achieve computational optimization. Given equal computing resources (i.e., virtual machines {VMs} with identical configuration), it is unlikely that all nodes running the image segmentation tasks will finish at the same time. There will be a moment when some nodes finish their tasks while others are still running. Our LUCC-based GCI streams image segments to ST atoms extraction VM, while other image segmentation jobs are still running.

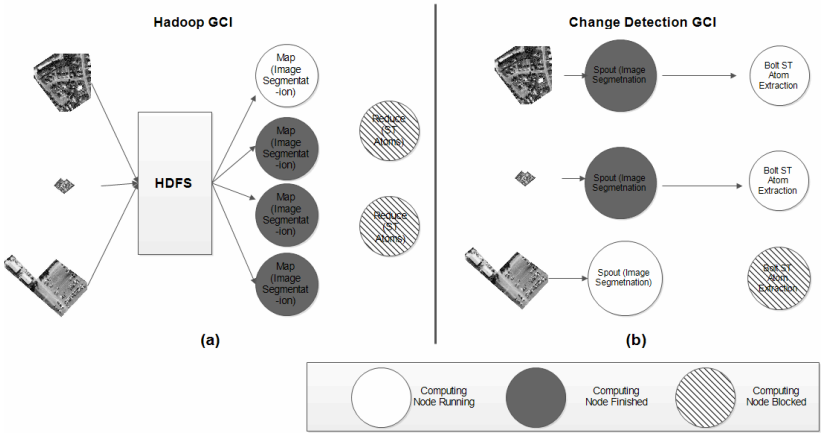


Figure 6-5. Comparison Between Hadoop GCI and Apache Storm in Change Detection GCI

We choose Storm as opposed to Hadoop. Compared with a Hadoop-based GCI (Li *et al.*, 2011), we argue that Storm in our GCI offers better flexibility and efficiency in job scheduling and dataflow management. Figure 6-5 shows a comparison between Storm and Hadoop. A Hadoop based GCI must wait until all VMs finish their image segmentation tasks. ST atom extraction can be scheduled in parallel with image segmentation on Storm. Another advantage of

a Storm based GCI is the use of data streams instead of intermediate data storage (e.g., HDFS {Hadoop Distributed File System} in Hadoop) for image segmentation results. This reduces input-output costs. At last, the cloud computing is employed to provide the scalable and flexible computing resource provisioning, as another part of the computation optimization.

6.4 Results

We designed two case studies to test our LUCC-based GCI. The first case tested the performance of our LUCC-based GCI to optimize the LUCC detection. The second case compared two different spatial optimization algorithms in multi-temporal LUCC detection.

Figure 6-6 and Table 6-2 show details of the implementation. The test bed was deployed on the Amazon Elastic Cloud Computing platform (Amazon EC2, 2017). Seventy-one VMs were utilized, with one local GCI controller and 70 nodes in the EC2 cloud. Sixty data streams were configured, with 30 streams connecting graph image segmentation VMs to ST atom extraction VMs. Twenty were utilized for the iterative ST atom extraction communication. The ST atom extraction VMs were connected to 10 classification and ST modelling VMs via 10 streams.

We used the Storm framework to map connections between VMs. During the implementation on Amazon, we utilized Amazon Kinesis to perform the actual data streaming. Kinesis is a data streaming service to manage real-time communication among Amazon cloud computing components (e.g., VMs) (Amazon Kinesis, 2014). Kinesis replaces the default stream of Storm and has been found to exert no impact on Storm's functionality (Bhartia, 2014). Then we utilized the topology interface of Storm to specify the topology of computing nodes and data streams.

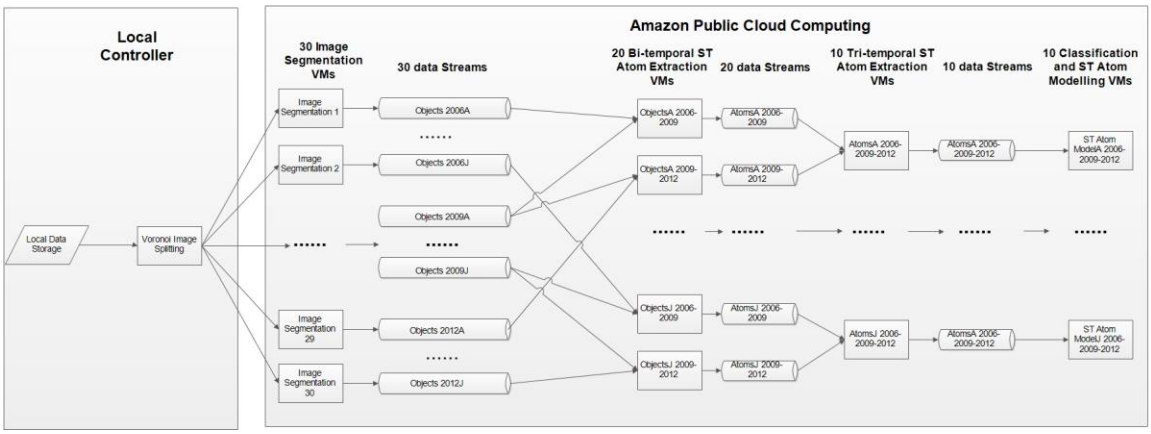


Figure 6-6. Implementation Details of LUCC-based GCI for Montreal 2006-2012 case study. The test bed includes 70 VMs in Amazon EC2 and 60 Amazon Kinesis data streams.

The evaluation imagery datasets were collected in 2006, 2009, and 2012 for the Greater Montreal area, Canada (DMTI, 2006, 2009, 2012), with total size of approximately 534GB. The DMTI StreetView RGB satellite images are recorded at 0.6m spatial resolution; we assume that no object is smaller than one pixel. We employ radiometric correction, geometric rectification, image registration to each of the testing datasets, and Voronoi image decomposition on the local controller node. Figure 6-6 shows the three steps of our ST Atom Modelling are hosted on Amazon EC2 cloud computing.

Table 6-2. Testbed Configurations

	Number of VMs	Instance Type	VCPU	VMemory	Local Storage
Pre-Processing and Image Decomposition	1	Private Cloud	8	32.0	1TB
Image Segmentation	30	m3.large	2	7.5	32 GB SSD
ST Atom Extraction	30	m3.large	2	7.5	32 GB SSD
Classification and ST Atom Modelling	10	m3.large	2	7.5	32 GB SSD

6.4.1 ST Optimization

We first tested our ST Atom Modelling in the LUCC-based GCI. After pre-processing, image tiles (i.e., Voronoi polygons) were uploaded onto Amazon cloud. A week was required to upload image tiles onto Amazon S3 cloud storage (Amazon S3, 2017), which highlights the input-output challenges in big data. The second column of Table 6-3 lists the computation time of each step in our LUCC-based GCI.

We first needed to parameterize the graph cut. We used a subset of the tiles (i.e., 5 image tiles sampled from different areas of Montreal) to extract parameters for the cost energy function. Parameters were applied to the whole dataset for image segmentation. To avoid over-fitting, a cross validation technique was applied (Hall and Koch, 1992). One hundred ground truth points were selected from other image tiles to fine-tune the parameters. Then graph cut image

segmentation was executed with the branch-and-mincut optimization. Finally, we followed Blaschke *et al*. (2000) and filtered objects unrelated to LUCC (e.g., vehicles).

The graph based image segmentation method generated 42,628 objects in 2006, 49,894 objects in 2009, and 47,742 objects in 2012. This represents relatively small differences (+17% and -4%) and reflects a depressed retail and residential market relative to other North American cities (e.g., CMHC, 2012). The Greater Montreal area has regions of dense urbanism but also is composed of agriculture and forest. We found a large number of objects located in urban portions of our study area, due to a high mixture of land uses.

ST atom extraction began with pre-processing, which included spatial interpolation, chain-code, and Douglas-Peucker algorithm. We did not employ spatial interpolation since all datasets matched in spatial and spectral resolutions. We found some instances of non-contiguous potential atoms so we utilized chain code together with a regression technique (Esbensen et al., 1992) and Douglas-Peucker to “snap” potential atoms in objects.

Temporal topology rules were applied to the change masks, which were then applied to extract the ST atoms from objects. The most applied temporal topology rules were “overlap” and “split”, which resulted in the large number of ST atoms. For example, the forest area in Figure 6-7 for 2006 was split as finer ST atoms, which either changed into buildings or remained forest across the whole study time span. Because there is some overlapping of the temporal topology rules (i.e., an object change can be classified as both "contain" and "partial change") and occasionally multiple rules can apply to one ST atom extraction. Building 4 in Figure 6-7 could utilize both “split” and “contain” rules. We chose the “split” rule because we utilized the 2006 objects as the “Object A” in Table 6-1. When multiple temporal topology rules are applicable, it is the implementation of the rules that determines the order of the rules’ application. So care

must be taken in coding. The bi-temporal ST atom extraction output 228,683 ST atoms. The tri-temporal process, which compared 2006-2009 and 2009-2012, generated 750,402 ST atoms.

Finally, we used the SVM classification algorithm to label seven different types of atoms (i.e., forest, grass, farmland, bare ground, water, roads and buildings). The SVM training process was conducted according to Melgani and Bruzzone (2004) with ST atom features, like brightness, shape, and texture. SVM classification and ST modelling labelled 750,402 ST atoms with the pre-defined seven classes, time-stamp the classified atoms, and linked them in time sequential order as ST Atom Models. "Forest" occupied the largest areas in Greater Montreal; whereas "building" class dominated in number.

One thousand ground truth sampling points were randomly chosen to evaluate the performance of our ST optimization. These 1000 points were visually inspected and assigned labels. The highest accuracy was achieved in forest areas, approximately 97 percent. The lowest accuracy is found in complex urban areas, approximately 85 percent. The reason for this reduced accuracy is that ST atom extraction generates excessive numbers of atoms when there are complex object mixtures and numerous iterations of temporal topology rules. This results in poorer classification performance. For example, temporal topology rules may split one road into small blocks due to road repair in one study time period. Small roadblocks may possess very similar geometric and texture attributes and may be misclassified as buildings.

Figure 6-7 shows an example of new buildings that were constructed in a forest in 2009. Our ST optimization was able to detect these changes and presented them as ST atom models. Our ST Atom Model assures no partial ST atom changes across the study period (the "forest" ST atoms either remain as "forest" {ST Atom1 and 2} or completely into "buildings" {ST Atom 3 and 4}), because we optimize change information in both spatial and temporal domains. This

evaluation proves the LUCC-based GCI can address partial object changes using ST Atom Modelling for big data analysis.

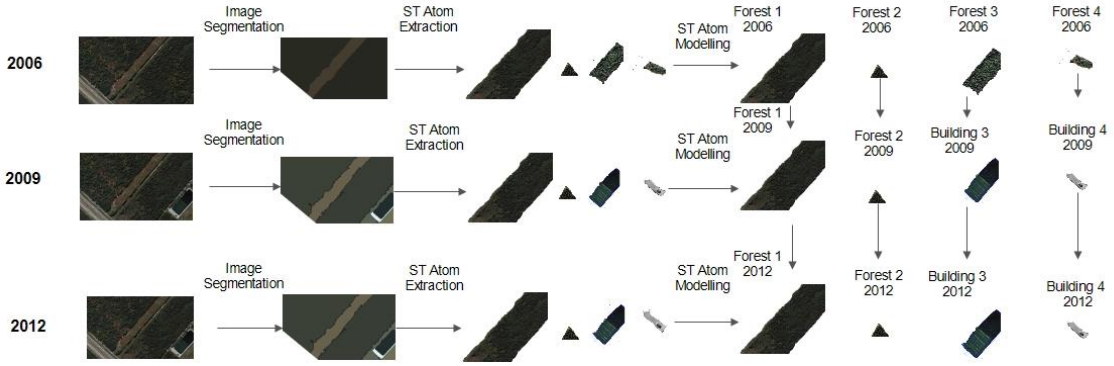


Figure 6-7. Examples of forest changing to buildings in suburban area with results of image segmentation, ST atom extraction, and ST atom classification and modelling. Streetview image collected in 2006, 2009, and 2012 from Greater Montreal Area (Source: DMTI Inc.)

Table 6-3. Computing Time for Steps in LUCC-based GCI. Then compared to a second implementation in which Hadoop replaces Storm.

Steps	LUCC-based GCI (Hours)	Hadoop-based GCI (Hours)
Pre-Processing and Image Decomposition	91.4	91.4
Image Segmentation with Optimization	19.7	26.5
ST Atom Extraction	4.6	19.3
Classification and Modeling	14.4	31.5
Total	130.1	168.7

6.4.2 Spatial Optimization Comparison

Image segmentation optimization plays a pivotal role in our LUCC-based GCI. We assessed its effect by comparing the min-cut/max-flow and branch-and-mincut algorithm. Another 1000 ground truth points were selected to evaluate two instances of LUCC from 2006 to 2009, and 2009 to 2012 in Greater Montreal area, Canada. We used a simple atom change/no-change error matrix (Macleod and Congalton, 1998) with reference data to test the performance of the two optimization algorithms. We used average accuracy from the change of seven predefined classes, and merged them as "change" and "no-change" super-classes. Overall accuracy can be calculated by adding the true change (change/change) and true no-change (no-change/no-change) percentage in Table 6-4. For 2006-2009 LUCC detection, min-cut/max-flow achieved 97.2 percent in overall accuracy; whereas branch-and-mincut was 98.0 percent. Min-cut/max-flow optimization produced 96.6 percent in 2009-2012 LUCC detection, but branch-and-mincut slightly outperformed with 97.3 percent. Both algorithms generated satisfying results, but the performance of branch-and-mincut was slightly better than min-cut/max-flow. The reason could be the best-first branch-and-bound search mechanism of branch-and-mincut, which determines the global optima with searching tree techniques (Lempitsky *et al*., 2012). Nonetheless, this result does not guarantee branch-and-mincut will always outperform the min-cut/max-flow algorithm in graph based image segmentation. Further study is needed to find a suitable global optimization technique for LUCC detection.

Table 6-4. Comparison between min-cut/max-flow and branch-and-mincut optimization algorithms

		2006-2009 Reference			2009-2012 Reference		
		Change (%)	No-Change (%)	Total (%)	Change (%)	No-Change (%)	Total (%)
min-cut/max-flow	Change (%)	6.7	1.4	8.1	5.2	0.8	6.0
	No-Change (%)	1.4	90.5	91.9	2.6	91.4	94.0
	Total (%)	8.7	91.3	100.0	7.8	92.2	100.0
branch-and-mincut	Change (%)	6.9	1.2	8.1	5.3	0.7	6.0
	No-Change (%)	0.8	91.1	91.9	2.0	92.0	94.0
	Total (%)	7.7	92.3	100.0	7.5	92.7	100.0

To evaluate our temporal optimization, we use the same 1000 ground truth points to compare the performance between the ST Atom Model and a standard OBIA change detection method (Chen *et al*. 2010). The overall accuracy of OBIA was 77.4 percent, which was 19.9 percent less than our ST Atom Model. We visually inspected the points and found OBIA mismarked 146 unchanged points as “change”, which were caused by the partial object changes (see Figure 6-2). Additionally, the standard OBIA output temporally isolated objects, which prevented the generation of change trajectories.

Finally, we tested the computing optimization induced by Storm with a Hadoop version. Results are shown in Table 6-3. Hadoop required 168.7 hours, 29.6 percent more than the streaming implementation. Extra time was induced by HDFS based data exchange and unnecessary waiting time (Figure 6-5). Delays from the previous step accumulated in later steps, which explained the increasing delay in ST atom extraction, and classification and modelling steps in the Hadoop implementation.

6.5 Conclusion

In this paper, we presented and evaluated a GCI-based ST optimization for LUCC. Optimization techniques play important roles in the GCI, including the spatial, temporal and computational optimization techniques. With GIScience becoming data-driven (Miller and Goodchild, 2015), GCI shifts as an important knowledge discovery approach. Thus the combination of domain optimization and computation optimization will become much stronger in the future GCI research.

With the ever-increasing amounts and speed of data, GCIs should integrate new methods for improving input-output and computing task scheduling. Despite new algorithms, more prosaic optimization of data handling is likely to constrain usage of GCIs. Advanced optimization algorithms also should be explored to improve the accuracy of image segmentation and LUCC detection. For example, algebraic geometry optimization (Wang, 2014) can improve image segmentation, and swarm optimization can optimize change/no-change thresholds (Liu *et al.*, 2014). This paper provides a preliminary step in re-shaping optimization as a combination of domain knowledge and computation. More work can be done to optimize ST models for LUCC. The ST model cannot express change rates explicitly and has limitations in describing changes semantically. This challenge might be solved by using Yuan's (1999) three domain model to

include additional semantic information in the form of description tags for changes. On the other hand, the relationship between domain specific optimization and different computation optimization techniques also calls for further exploration. Hopefully, new methodologies of ST optimization within GCI will remain a focus in future research.

6.6 Disclosure statement

No potential conflict of interest was reported by the authors.

6.7 Funding

This research is supported by an Amazon AWS Research Grants from Amazon.com, Inc.

6.8 Reference

Aggarwal, C. C. (2007). Data streams: models and algorithms (Vol. 31). Springer.

Allen, J. F. (1991). Time and time again: The many ways to represent time. International Journal of Intelligent Systems, 6(4), 341-355.

Amazon Elastic Cloud Computing (EC2) (2014). Online: http://aws.amazon.com/ec2/

Amazon Kinesis (2014). Online: http://aws.amazon.com/kinesis/

Amazon Simple Storage Service (S3) (2014). Online: http://aws.amazon.com/s3/

Apache, Apache Storm (2014). Online: https://storm.incubator.apache.org/

Abraham, T. and Roddick, J.F., 1999. Survey of spatio-temporal databases. GeoInformatica 3(1): 61-99. Pay attention to page 76.

Armstrong, M. P., (1988). Temporality in spatial databases. Proceedings: GIS/LIS'88, 2:880-889.

Baatz, M., & Schäpe, A. (2000). Multiresolution segmentation: an optimization approach for high quality multi-scale image segmentation. Angewandte Geographische Informationsverarbeitung XII, 12-23.

Bazi, Y., Bruzzone, L., & Melgani, F. (2005). An unsupervised approach based on the generalized Gaussian model to automatic change detection in multitemporal SAR images. Geoscience and Remote Sensing, IEEE Transactions on, 43(4), 874-887.

Bhartia, R., (2014). Implement a Real-time, Sliding-Window Application Using Amazon Kinesis and Apache Storm. (2014). Retrieved December 4, 2014, from

http://blogs.aws.amazon.com/bigdata/post/Tx36LYSCY2R0A9B/Implement-a-Real-time-Sliding-Window-Application-Using-Amazon-Kinesis-and-Apache

Blaschke, T. (2010). Object based image analysis for remote sensing. ISPRS Journal of Photogrammetry and Remote Sensing, 65(1), 2-16.

Blaschke, T., Lang, S., Lorup, E., Strobl, J., & Zeil, P. (2000). Object-oriented image processing in an integrated GIS/remote sensing environment and perspectives for environmental applications. Environmental information for planning, politics and the public, 2, 555-570.

Boykov, Y., & Funka-Lea, G. (2006). Graph cuts and efficient ND image segmentation. *International Journal of Computer Vision*, *70*(2), 109-131.

Boykov, Y., Veksler, O., & Zabih, R. (2001). Fast approximate energy minimization via graph cuts. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 23(11), 1222-1239.

Camps-Valls, G., Gómez-Chova, L., Muñoz-Marí, J., Rojo-Álvarez, J. L., & Martínez-Ramón, M. (2008). Kernel-based framework for multitemporal and multisource remote sensing data classification and change detection. Geoscience and Remote Sensing, IEEE Transactions on, 46(6), 1822-1835.

Canada Mortgage and Housing Corporation, (2014). Housing Market outlook, Montréal CMA. Retrieved December 4, 2014, from http://www.cmhc-schl.gc.ca/odpub/esub/64291/64291_2012_B01.pdf?fr=1345862116234

Celik, T., & Yetgin, Z. (2011). Change detection without difference image computation based on multiobjective cost function optimization. Turk. J. of Elec. Eng. & Comp. Sci, 19(6), 941-956.

Chen, G., Hay, G. J., Carvalho, L. M., & Wulder, M. A. (2012). Object-based change detection. *International Journal of Remote Sensing*, 33(14), 4434-4457.

Chen, J., Pappas, T. N., Mojsilovic, A., & Rogowitz, B. (2005). Adaptive perceptual color-texture image segmentation. *Image Processing, IEEE Transactions on*, *14*(10), 1524-1536.

Congalton, R. G. (1991). A review of assessing the accuracy of classifications of remotely sensed data. Remote sensing of environment, 37(1), 35-46.

Coppin, P., Jonckheere, I., Nackaerts, K., Muys, B., & Lambin, E. (2004). Review ArticleDigital change detection methods in ecosystem monitoring: a review. International journal of remote sensing, 25(9), 1565-1596.

Deng, J. S., Wang, K., Hong, Y., & Qi, J. G. (2009). Spatio-temporal dynamics and evolution of land use change and landscape pattern in response to rapid urbanization. Landscape and Urban Planning, 92(3), 187-198.

DMTI, 2006, 2009 , 2012. Montreal Satellite StreetView, 3_1- 9_7_MONTREAL-S3XM, Markham ON: DMTI Spatial Inc., 2006, 2009 and 2012.

Egenhofer, M. J., & Al-Taha, K. K. (1992). Reasoning about gradual changes of topological relationships. In Theories and methods of spatio-temporal reasoning in geographic space (pp. 196-219). Springer Berlin Heidelberg.

Erwig, M., Gu, R. H., Schneider, M., & Vazirgiannis, M. (1999). Spatio-temporal data types: An approach to modeling and querying moving objects in databases. GeoInformatica, 3(3), 269-296.

Esbensen, K. H., Geladi, P. L., & Grahn, H. F. (1992). Strategies for multivariate image regression. Chemometrics and intelligent laboratory systems, 14(1), 357-374.

Gallo, G., Grigoriadis, M. D., & Tarjan, R. E. (1989). A fast parametric maximum flow algorithm and applications. SIAM Journal on Computing, 18(1), 30-55.

Gao, S., Li, L., Li, W., Janowicz, K., & Zhang, Y. (2014). Constructing gazetteers from volunteered big geo-data based on Hadoop. Computers, Environment and Urban Systems, doi:10.1016, in press.

Hall, P., & Koch, I. (1992). On the feasibility of cross-validation in image analysis. SIAM Journal on Applied Mathematics, 52(1), 292-313.

Hampton, S. E., Strasser, C. A., Tewksbury, J. J., Gram, W. K., Budden, A. E., Batcheller, A. L., ... & Porter, J. H. (2013). Big data and the future of ecology. Frontiers in Ecology and the Environment, 11(3), 156-162.

Huang, B., & Chandramouli, M. (2009). Spatio-Temporal Object Modeling. Handbook of Research on Geoinformatics.

Jenerette, G. D., & Wu, J. (2001). Analysis and simulation of land-use change in the central Arizona–Phoenix region, USA. Landscape ecology, 16(7), 611-626.

Jermyn, I. H., & Ishikawa, H. (2001). Globally optimal regions and boundaries as minimum ratio weight cycles. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 23*(10), 1075-1088.

Jianya, G., Haigang, S., Guorui, M., & Qiming, Z. (2008). A review of multi-temporal remote sensing data change detection algorithms. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 37(B7), 757-762.

Kolmogorov, V., Boykov, Y., & Rother, C. (2007). Applications of parametric maxflow in computer vision. In Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on (pp. 1-8). IEEE.

Lam, N. S. N. (1983). Spatial interpolation methods: a review. The American Cartographer, 10(2), 129-150.

Langran, G., & Chrisman, N. R. (1988). A framework for temporal geographic information. Cartographica: The International Journal for Geographic Information and Geovisualization, 25(3), 1-14.

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(4), 11-20.

Lempitsky, V., Blake, A., & Rother, C. (2012). Branch-and-mincut: global optimization for image segmentation with high-level priors. *Journal of Mathematical Imaging and Vision*, *44*(3), 315-329.

Li, H., Manjunath, B. S., & Mitra, S. K. (1995). A contour-based approach to multisensor image registration. Image Processing, IEEE Transactions on, 4(3), 320-334.

Li, J., Bioucas-Dias, J. M., & Plaza, A. (2010). Semisupervised hyperspectral image segmentation using multinomial logistic regression with active learning. Geoscience and Remote Sensing, IEEE Transactions on, 48(11), 4085-4098.

Li, Q., Zhang, T., & Yu, Y. (2011). Using cloud computing to process intensive floating car data for urban traffic surveillance. International Journal of Geographical Information Science, 25(8), 1303-1322.

Li, X., & Yeh, A. G. O. (2002). Neural-network-based cellular automata for simulating multiple land use changes using GIS. International Journal of Geographical Information Science, 16(4), 323-343.

Liang, S., Chen, S., Huang, C., Li, R., Chang, Y., Badger, J., & Rezel, R. (2010, September). Geocens: Geospatial cyberinfrastructure for environmental sensing. In Proceedings of GIScience 2010—Sixth international conference on Geographic Information Science (Vol. 6292). Zurich, Switzerland: Springer.

Ligmann‐Zielinska, A., Church, R. L., & Jankowski, P. (2008). Spatial optimization as a generative technique for sustainable multiobjective land‐use allocation. International Journal of Geographical Information Science, 22(6), 601-622.

Lillesand, T. M., Kiefer, R. W., & Chipman, J. W. (2004). Remote sensing and image interpretation (No. Ed. 5). John Wiley & Sons Ltd.

Liu, X., Veksler, O., & Samarabandu, J. (2008). Graph cut with ordering constraints on labels and its applications. In Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on (pp. 1-8). IEEE.

Liu, Y., Hu, K., Zhu, Y., & Chen, H. (2014). A Novel Method for Image Segmentation Based on Nature Inspired Algorithm. In Intelligent Computing in Bioinformatics (pp. 390-402). Springer International Publishing.

Lu, D., Mausel, P., Brondizio, E., & Moran, E. (2004). Change detection techniques. *International journal of remote sensing*, *25*(12), 2365-2401.

Macleod, R. D., & Congalton, R. G. (1998). A quantitative comparison of change-detection algorithms for monitoring eelgrass from remotely sensed data. Photogrammetric Engineering and Remote Sensing, 64(3), 207-216.

Madden, S., & Franklin, M. J. (2002). Fjording the stream: An architecture for queries over streaming sensor data. In Data Engineering, 2002. Proceedings. 18th International Conference on Data Engineering (pp. 555-566). IEEE.

Malik, J., Belongie, S., Leung, T., & Shi, J. (2001). Contour and texture analysis for image segmentation. International Journal of Computer Vision, 43(1), 7-27.

McAfee, A., Brynjolfsson, E., Davenport, T. H., Patil, D. J., & Barton, D. (2012). Big data. The management revolution. Harvard Bus Rev, 90(10), 61-67.

Melgani, F., & Bruzzone, L. (2004). Classification of hyperspectral remote sensing images with support vector machines. *IEEE Transactions on Geoscience and Remote Sensing* 42(8), 1778-1790.

Miller, H. J., & Goodchild, M. F. (2014). Data-driven geography. GeoJournal, 1-13.

Müller, J. C., & Zeshen, W. (1992). Area-patch generalisation: a competitive approach. The Cartographic Journal, 29(2), 137-144.

Myint, S. W., Gober, P., Brazel, A., Grossman-Clarke, S., & Weng, Q. (2011). Per-pixel vs. object-based classification of urban land cover extraction using high spatial resolution imagery. Remote Sensing of Environment, 115(5), 1145-1161.

Nadi, S., & Delavar, M. R. (2003). Spatio-Temporal Modeling of Dynamic Phenomena in GIS. In ScanGIS (pp. 215-225).

Nandal, R. (2013). Spatio-Temporal Database and Its Models: A Review.Journal of Computer Engineering (IOSR-JCE), 2278-0661, p-ISSN: 2278-8727,Volume 11, Issue 2 (May. - Jun. 2013), PP 91-100.

Nurain, N., Sarwar, H., Sajjad, M. P., & Mostakim, M. (2012). An In-depth Study of Map Reduce in Cloud Environment. In Advanced Computer Science Applications and Technologies (ACSAT), 2012 International Conference on (pp. 263-268). IEEE.

Pan, D., Domon, G., De Blois, S., & Bouchard, A. (1999). Temporal (1958–1993) and spatial patterns of land use changes in Haut-Saint-Laurent (Quebec, Canada) and their relation to landscape physical attributes. Landscape Ecology, 14(1), 35-52.

Parent, C., Spaccapietra, S., & Zimányi, E. (1999). Spatio-temporal conceptual models: data structures+ space+ time. In Proceedings of the 7th ACM international symposium on Advances in geographic information systems (pp. 26-33). ACM.

Peuquet, D. J., & Duan, N. (1995). An event-based spatiotemporal data model (ESTDM) for temporal analysis of geographical data. International journal of geographical information systems, 9(1), 7-24.

Pijanowski, B. C., Brown, D. G., Shellito, B. A., & Manik, G. A. (2002). Using neural networks and GIS to forecast land use changes: a land transformation model. Computers, environment and urban systems, 26(6), 553-575.

Quesada, I., & Grossmann, I. E. (1992). An LP/NLP based branch and bound algorithm for convex MINLP optimization problems. Computers & Chemical Engineering, 16(10), 937-947.

Radke, R. J., Andra, S., Al-Kofahi, O., & Roysam, B. (2005). Image change detection algorithms: a systematic survey. Image Processing, IEEE Transactions on, 14(3), 294-307.

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).

Saalfeld, A. (1999). Topologically consistent line simplification with the Douglas-Peucker algorithm. Cartography and Geographic Information Science, 26(1), 7-18.

Sengupta, R. and Yan, C., 2004. A Hybrid Spatio-Temporal Data Model and Structure for Efficient Storage and Retrieval of Land Use Information. Transactions in GIS 8(3): 351-366.

Shi, J., & Malik, J. (2000). Normalized cuts and image segmentation. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 22(8), 888-905.

Singh, A. (1989). Review article digital change detection techniques using remotely-sensed data. International journal of remote sensing, 10(6), 989-1003.

Sumengen, B., & Manjunath, B. S. (2006). Graph partitioning active contours (GPAC) for image segmentation. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 28(4), 509-521.

Tong, D., & Murray, A. T. (2012). Spatial optimization in geography. Annals of the Association of American Geographers, 102(6), 1290-1309.

Van Der Wal, D., & Pye, K. (2003). The use of historical bathymetric charts in a GIS to assess morphological change in estuaries. The Geographical Journal, 169(1), 21-31.

Veldkamp, A., & Lambin, E. F. (2001). Predicting land-use change. Agriculture, ecosystems & environment, 85(1), 1-6.

Vincent, L., & Soille, P. (1991). Watersheds in digital spaces: an efficient algorithm based on immersion simulations. IEEE transactions on pattern analysis and *machine intelligence*, *13*(6), 583-598.

Walter, V. (2004). Object-based classification of remote sensing data for change detection. ISPRS Journal of photogrammetry and remote sensing, 58(3), 225-238.

Wang, M. (2014). An Improved Image Segmentation Algorithm Based on Principal Component Analysis. In Proceedings of the 9th International Symposium on Linear Drives for Industry Applications, Volume 4 (pp. 811-819). Springer Berlin Heidelberg.

Wang, S. (2010). A CyberGIS framework for the synthesis of cyberinfrastructure, GIS, and spatial analysis. Annals of the Association of American Geographers, 100(3), 535-557.

Winn, J., & Shotton, J. (2006). The layout consistent random field for recognizing and segmenting partially occluded objects. In Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on (Vol. 1, pp. 37-44). IEEE.

Worboys, M., 2005. Event-oriented approaches to geographic phenomena. International Journal of Geographical Information Science 19(1): 1-28.

Wu, Z., & Leahy, R. (1993). An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, *15*(11), 1101-1113.

Xia, Y. J., Kuang, L., & Li, X. M. (2011). Accelerating geospatial analysis on GPUs using CUDA. Journal of Zhejiang University SCIENCE C, 12(12), 990-999.

Xing, J., & Sieber, R. (2014). Sampling Based Image Splitting In Large Scale Distributed Computing Of Earth Observation Data, 2014. Proceedings 35th IEEE Geoscience and Remote Sensing Symposium. IEEE.

Xing, J., Sieber, R., & Kalacska, M. (2014). The challenges of image segmentation in big remotely sensed imagery data. Annals of GIS, 20(4), 233-244.

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

Yang, C., Raskin, R., Goodchild, M., & Gahegan, M. (2010). Geospatial cyberinfrastructure: past, present and future. Computers, Environment and Urban Systems, 34(4), 264-277.

Yang, Z., Zhu, Y., & Pu, Y. (2008). Parallel image processing based on CUDA. In Computer Science and Software Engineering, 2008 International Conference on (Vol. 3, pp. 198-201). IEEE.

Yuan, M. (1996, January). Temporal GIS and spatio-temporal modeling. In Proceedings of Third International Conference Workshop on Integrating GIS and Environment Modeling, Santa Fe, NM.

Yuan, M. (1999). Use of a Three Domain Repesentation to Enhance GIS Support for Complex Spatiotemporal Queries. Transactions in GIS, 3(2), 137-159.

Yue, P., Gong, J., & Di, L. (2010). Augmenting geospatial data provenance through metadata tracking in geospatial service chaining. Computers & Geosciences, 36(3), 270-281.

Zaslavsky, A., Perera, C., & Georgakopoulos, D. (2013). Sensing as a service and big data. arXiv preprint arXiv:1301.0159.

Chapter 7. Conclusion

7.1 General Summary

Big data has brought different challenges into Remote Sensing (RS)-based Land Use/Cover Change (LUCC) research, and my dissertation focuses mainly on addressing high volume and variety of geographical information. In Chapter 2, I review most recent key publications in RS-based LUCC, together with big data processing techniques. I also highlight the scale challenges in RS-based LUCC which originate from the ever increasing spatial, spectral, and temporal resolutions and extents of the sensing platforms. In Chapter 3, I propose the concept of Scope, which models scale with the spatial granularity, extent, time, and property. I present the decomposition/recomposition framework in Chapter 4, which manages the workflow of big data in the distributed computing environment. I illustrate the scale invariant LUCC detection algorithm in Chapter 5, by merging Scale Invariant Feature Transformation (SIFT) and Maximally Stable Extremal Region (MSER) for LUCC identification. In Chapter 6, I utilize Geospatial CyberInfrastructure (GCI) to spatial-temporally optimize the segmentation-based LUCC method with cloud computing and Apache Strom, and name the methodology as LUCC-GCI.

My dissertation provides a holistic solution for big data analysis in RS-based LUCC. The main contributions of my dissertation include the Scope methodology for scale modelling, the decomposition/recomposition dataflow management framework, the new scale invariant LUCC detection algorithm, and the methodology to integrate LUCC workflow with advanced high performance computing techniques as LUCC-GCI.

Meanwhile, I also acknowledge the limitation of my RS-based LUCC detection methodologies. First, several LUCC regions might be missed if the RS platforms fail to record

them. Second, it is difficult to generate long time-span urban change patterns due to the limited RS data availability without integrating urban growth models (Moghadam and Helbich, 2013). Third, I highlight errors in the sensing technologies, such as the errors in sensor calibration, the maneuverability, and the signal processing. The recent advancement in satellite constellation can be a potential solution to minimize the compound errors in RS data collection (Sierawski et al., 2017) by fusing images from constellation members. Finally, each LUCC steps may inevitably bring additional errors, as the prorogation of errors. For example, the change map smoothing method in my scale invariant LUCC detection method may remove small changed regions by mistake. How to detect these errors and evaluate the uncertainty in the workflow of LUCC remains an open research question (Olofsson et al., 2013).

7.2 Discussion

Scope covers a very important topic in Geographic Information Science (GIScience)— the Modifiable Area Unit Problem (MAUP). MAUP describes the problems when the smaller areal units are grouped into larger but fewer area units, features and attributes of the original data change accordingly (Openshaw and Taylor, 1979). The areal unit can take any size or shape, which brings greater scale complexity. MAUP provides a great opportunity to employ Scope, in which we can model these areal units as Scope with different granularities and extents, and rely on various properties to represent the features and object attributes. On the other hand, Alvanides, Openshaw, and Macgill (2001) introduced zone design as a solution for MAUP, which captured the variation of both properties and scales with units aggregation. The Scope quadruple projection turns out to be an abstraction of the zone design method, by tracing the projection of the original data into a series of Scope quadruples with different granularities and extents, where aggregation algorithms could be encoded as the *Algorithm* in Scope quadruple projections. Therefore, Scope can serve as an efficient solution for MAUP.

The decomposition/recomposition framework plays a pivotal role in big data handling. But there is no guarantee that recomposition can remove all the problems aroused by the artificial border challenge. Another solution is to take advanced data decomposition techniques to avoid cutting geographic objects. For example, Xing and Sieber (2014) have proposed a Voronoi diagram-based approach to minimize the artificial border challenge in data decomposition. However, the additional computation costs incurred by advanced data decomposition methods need to be taken into account, especially working with public cloud computing.

The scale invariant LUCC detection algorithm presents high accuracy in rural and suburban areas, but lower accuracy in dense urban regions. The main reason is the high complexity of geographic entities in urban areas. Big data provides more LUCC details with increasing granularities and extents, but the noise and errors also get augmented. Therefore, finer granularity and larger extents do not guarantee higher LUCC accuracy. Another challenge is the similarity of urban entities, which cannot be fully distinguished from the SIFT descriptors. Therefore, future research in LUCC detection needs to investigate new image feature descriptors to address the representation of geographic entities

The classification of RS-based LUCC becomes a semantic problem in the era of big data. Classification does not only attach labels to the image regions, but also help the cognition of the LUCC trajectories. For example, rainfall in a given area is not labelled as LUCC, but the flooding caused by the rainfall is a type of widely accepted LUCC (Sanyal and Lu, 2004). It is the ontology and temporal models that help us distinguish “rainfall” from “flooding” in remotely sensed images. Although the Stommel diagram-based method (Stommel, 1963) can be utilized to select the appropriate spatial-temporal labels for LUCC, temporal models are still necessary in

LUCC identification, especially for the change trajectory generation and investigation of the LUCC speed. To summarize, LUCC study needs to investigate both temporal and semantic models to improve the classification of RS-based LUCC.

Finally, LUCC-GCI proves the success of merging domain knowledge and high performance computing as a big data solution. On the one hand, new high performance computing techniques, such as Apache Storm (Apache, 2017), Apache Hama (Apache, 2017), and Apache Nifi (Apache, 2017), will continue improve the computational efficiency of GCI. On the other hand, new methodology are being invented to integrate various domain knowledge into GCI (Zhuge, 2015). GCI has already become a systematic methodology for knowledge discovery and decision-making as a new branch of GIScience, and GCI will be employed more frequently for big data analysis.

To summarize, my dissertation provides a holistic big data solution for GIScience and RS research, with a focus on scale. First, Scope model clarifies and integrates the complex meanings of scale in GIScience and RS, which also provides the traceability of scaling operations. It could be very useful for research involving different types of scaling operations. Second, the decomposition/recomposition dataflow management framework solves the artificial border challenge and proposes a general solution to handle large data volume with distributed computing environment. Third, the scale invariant LUCC detection method identifies LUCC from scale invariant image features, without resorting to resampling for scale heterogeneity handling. It illustrates a general geospatial data analysis framework to incorporate compute vision algorithms to address scale heterogeneity. Fourth, LUCC-GCI demonsttrates that the spatial optimization does not only depend on algorithms, but also on computation, especially within GCI. Finally, I emphasize that big data does not mean better knowledge discovery in

LUCC study, since the topological and geometric information shape it much more complicated than a simple aggregation of small data.

7.3 Future Directions

There are three main directions for my future research. The first one is the advanced scale invariant LUCC detection algorithm with new deep learning techniques (LeCun, Bengio, and Hinton, 2015). Since Yi et al. (2016) have obtained enhanced SIFT matching through deep learning, I plan to use deep learning to improve my scale invariant LUCC detection algorithm in Section 5. I also want to implement the SIFT flow field into LUCC (Liu, Yuen, and Torralba, 2011), to enhance the performance of the SIFT-based LUCC detection.

The second direction is the investigation of advanced computing techniques in LUCC-GCI. I plan to build the LUCC workflow with Hama (Apache, 2017) in my future research, and compare its performance with the Hadoop-based LUCC-GCI. Apache Hama utilizes the Bulk Synchronous Parallel computation model on Hadoop to achieve higher speedups (Golghate and Shende, 2014). Moreover, I also plan to integrate Apache Nifi for the job deployment and management in the new LUCC-GCI.

The last direction for future work is the Stommel diagram-based LUCC classification (Stommel, 1963). This approach will explore the correlations among multiple spatial-temporal granularities and extents in the RS-based LUCC to determine the appropriate class labels. This study will require huge amounts of RS data collected in longer time spans for the training in the Stommel diagram modelling. Fortunately, big data can continue provide increasing remotely sensed datasets for the training process, because of the rapid development of sensing platforms. Therefore, big data does not only bring unprecedented challenges, but also considerable new

opportunities. In additional to these three directions, I also plan to investigate the social and economic influence on LUCC, to get better understanding of the human-landscape interaction.

7.4 Reference

Alvanides, S., Openshaw, S., & Macgill, J. (2001). Zone design as a spatial analysis tool. Modelling Scale in Geographical Information Science, London, 141-157.

Apache, Apache Hama (2017). Online: https://hama.apache.org/

Apache, Apache Nifi (2017). Online: https://nifi.apache.org/

Apache, Apache Storm (2017). Online: http://storm.apache.org/

Golghate, A. A., & Shende, S. W. (2014). Parallel k-means clustering based on hadoop and hama. International Journal of Computing and Technology,1(2014).

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature,521(7553), 436-444.

Liu, C., Yuen, J., & Torralba, A. (2011). Sift flow: Dense correspondence across scenes and its applications. IEEE transactions on pattern analysis and machine intelligence, 33(5), 978-994.

Moghadam, H. S., & Helbich, M. (2013). Spatiotemporal urbanization processes in the megacity of Mumbai, India: A Markov chains-cellular automata urban growth model. Applied Geography, 40, 140-149.

Olofsson, P., Foody, G. M., Stehman, S. V., & Woodcock, C. E. (2013). Making better use of accuracy data in land change studies: Estimating accuracy and area and quantifying uncertainty using stratified estimation. Remote Sensing of Environment, 129, 122-131.

Openshaw, S. and Taylor, P.J. (1979). A million or so correlation coefficients: three experiments on the modifiable areal unit problem. In Wrigley, N.,editor, Statistical applications in spatial sciences, London: Pion, 127–44.

Sanyal, J., & Lu, X. X. (2004). Application of remote sensing in flood management with special reference to monsoon Asia: a review. Natural Hazards, 33(2), 283-301.

Sierawski, B. D., Reed, R. A., Warren, K. M., Sternberg, A. L., Austin, R. A., Trippe, J. M., ... & Fleetwood, D. M. (2017). CubeSat: Real-time soft error measurements at low earth orbits. In Reliability Physics Symposium (IRPS), 2017 IEEE International (pp. 3D-1). IEEE.

Stommel, H. (1963). Varieties of oceanographic experience. Science,139(3555), 572-576.

Xing, J., & Sieber, R. (2014). Sampling Based Image Splitting In Large Scale Distributed Computing Of Earth Observation Data, 2014. Proceedings 35th IEEE Geoscience and Remote Sensing Symposium. IEEE.

Yi, K. M., Trulls, E., Lepetit, V., & Fua, P. (2016). Lift: Learned invariant feature transform. In European Conference on Computer Vision (pp. 467-483). Springer International Publishing.

Zhuge, H. (2015). Mapping Big Data into Knowledge Space with Cognitive Cyber-Infrastructure. arXiv preprint arXiv:1507.06500.