SAMPLING BASED IMAGE SPLITTING IN LARGE SCALE DISTRIBUTED COMPUTING OF EARTH OBSERVATION DATA

Jin Xing and Renée Sieber

Department of Geography
McGill University

ABSTRACT

With increasing amounts of spatial, spectral and temporal remote sensing data and heterogeneity of platforms, we have entered an era of big data in remote sensing research. Imagery now routinely exceeds the memory size of personal computers so splitting/distributing big remote sensing data becomes a necessary pre-processing step. Standard rectangle based splitting methods can distort existing geometric and topological information and lose features as images are split into tiles.

To address these challenges, we propose a sampling based image splitting method, which models the dataset as a streaming service and splits the dataset with a Voronoi diagram. The streaming data is systematically sampled to initially select the seeds of a Voronoi diagram. Voronoi regions are then generated according to spatial and spectral distances using Fortune's sweepline algorithm [1]. We test the splitting method with AVIRIS imagery of North America in 2013 (courtesy of NASA/JPL-Caltech) to evaluate the ability to detect objects of our splitting method. For evaluation we employ the object-based classification method of Hay and Castilla [2]. In contrast to rectangle based splitting approaches, most polygon borders generated by our method are found to converge with object borders (e.g., trees, building, and roads). When deployed with MapReduce, our sampling based splitting method also helps balance the computation intensity between each computing node.

Index Terms— Big Data, Splitting, Sreaming, Voronoi

1. INTRODUCTION

Big data has attracted considerable interest in a broad range of research fields, including economy [3], soci- ology [4], biology [5], Geographic Information Science (GIScience) [6], and Remote Sensing (RS) [7]. Big data is often described by the "4Vs": Volume, Variety, Velocity, and Veracity. Advancements in sensing platforms, the spatial, spectral, and temporal resolutions of remotely sensed imagery datasets mean that image processing represents an ideal type of big data. Platform growth and increasing resolutions result in larger data volumes, and RS datasets are increasing at petabyte-level rate [8]. Variety can be interpreted as images with different spatial and spectral resolutions, stored in different data formats (e.g., GeoTIFF and ecw) [9]. Even tiles with the same file size can express variety in terms of object complexity within the file [10]. Velocity comes from higher temporal resolutions of RS datasets, resulting in shorter data collection intervals. This means that data might be changed in the process of analysis (i.e., new data being added and old data being removed). Big data poses significant challenges in data management and also in corresponding analysis and computing. We may not have the accurate data attributes (e.g., spectral statistics and number of objects) available before we analysis the big RS datasets and these datasets may change in the flow of analysis. Therefore, we need new computing models for big RS data analysis.

Large volumes of big RS datasets exceed the memory and storage of most desktops so imagery datasets must be split into smaller tiles for distributed computing [11]. The most widely used RS data splitting method is rectangle based partitioning [12], which simply assumes the spectral and special information in RS datasets is homogeneous. Rectangle based partitioning method can incur errors in object-based RS analysis [2] because it changes the original object attributes in the RS imagery datasets by cutting objects. In this paper, we propose a sampling based Voronoi splitting method to address these drawbacks.

We use a sampling based Voronoi splitting method to decompose big RS imagery datasets into Voronoi regions. The Voronoi diagram can efficiently decompose a space into a collection of disjoint polygons (regions) [1], which then can be analyzed with existing computing tools. By integrating spatial and spectral information in the distance calculation (see below), our algorithm can split large imagery datasets by object boundaries. We also use a duplicate sweepline algorithm to reduce the cutting of objects.

This paper is organized as follows. Section 2 introduces related research and describes a central challenge: splitting big RS imagery datasets. In Section 3, we present our sampling based Voronoi splitting method, based on systematic sampling, streaming, and Fortune's algorithm. The comparison with rectangle based partitioning algorithm using AVIRIS imagery datasets in object-based classification is shown in Section 4. We conclude in Section 5.

978-1-4799-5775-0/14/$31.00 ©2014 IEEE

2. PROBLEM STATEMENT

Current research for big data in RS concentrates on high performance computing tools for big RS data processing but not necessarily the special features of big RS data. For example, Liang et al. [13] utilize social networks and hybrid P2P (Peer-to-Peer) techniques to enable sharing and visualization of big environmental sensing datasets. Wang et al. begin to tackle the challenges in modifying geostatistics for big RS imagery by introducing cloud computing [14] for rapid clustering analysis [15]. However, big RS imagery datasets are much more than a collection of independent small image tiles, as objects may span several tiles.

To handle big RS imagery datasets, scientists tend to use a divide and conquer method—partition into smaller tiles and then process in individual computing nodes in a distributed environment [16]. Generally, the tiles are numerous equal-size rectangular blocks to balance the load on each computing node [17]. This method neglects the spectral and spatial information in the data and can fail to balance objects in each tile. Figure 2-1 exemplifies the inability of current techniques to consider the spatial and spectral information of the imagery datasets and the lack of load balancing of computing intensity in object-based analysis. Figure 2.1 (a) and (b) show rectangle partitioned DMIT satellite images taken from two different locations in Montreal 2006 [18]. Their corresponding object extraction [19] in (c) and (d) produce quite different results. Moreover, using eCognition [20] to obtain (c) takes 7 seconds on a Dell laptop while (d) takes 21 seconds with the same computing configurations.

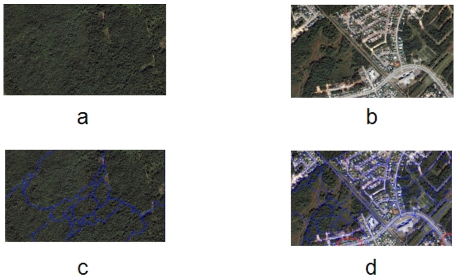

Figure 2-1 (a) and (b) are tiles from [18] (60 cm RGB); (c) and (d) are objects extracted using eCognition (object-based image analysis software) from (a) and (b), with color=0.3 and scale=150

A particular problem in this simple partitioning is that the objects are cut and therefore the object's attributes are altered. In Figure 2-1 (c) and (d), the cutting borders do not actually exist but they have to be analyzed as the real boundaries of objects. In (d), rectangle base partitioning creates two "false ends" to the highway (red circles), which changes the original characteristics of the road. To address these challenges, we propose a sampling based Voronoi splitting method.

3. SAMPLING BASED VORONOI SPLITTING METHOD

3.1. Big Data as Streaming Service

Considerable research is being conducted into data streaming in big data, for example, mining Twitter data for sentiment analysis [21]. Whereas, most of the big data research is on social media like Twitter, we use stream in the multimedia sense [22], which represents data as a sequence of chunks delimited by spatial extent and time stamps. In RS, data streaming has previously been utilized to visualize big RS datasets [23] or collect RS imagery datasets in real-time [24]. Large volumes can be modeled as several data streams, where different types of streams can be used for various purposes (e.g., vegetation and transportation). In this way, streaming services encapsulate the volume, variety, and velocity features of big RS data.

For us, the most straightforward reason for using data streaming modelling is to provide an efficient way to load big RS imagery datasets into distributed computing memories. Our streaming modelling provides such a mechanism. We split the data in two phases. First, we stream the data as "chunks" and then pass Fortune's algorithm through them in multiple passes. We do this in sequential passes to limit the effect of cut borders or edges (Section 3.2). The streaming process allows us to generate the second splitting into Voronoi regions [25], which are also distributed.

3.2. Sampling Based Voronoi Splitting Method

A Voronoi diagram partitions space into a collection of regions, using a set of seeds. The Voronoi region of seed *s* is all the points in the image for which *s* is the closest seed. Voronoi diagrams are widely employed in GIS data analysis. Akdogan et al. [26] use Voronoi diagram for geospatial query processing, and Li et al. [27] utilize Voronoi diagram for image segmentation. We employ the Voronoi diagram for big RS imagery dataset splitting. We define distance *d* of a pixel (x,y) from seed *s* as a combination of spatial and spectral distance:

$$d = \alpha \left\| I - I_s \right\|_2 + (1-\alpha)\sqrt{(x-x_s)^2 + (y-y_s)^2}$$

$\|I-I_s\|_2$ is the spectral distance and α is the weight parameter with range from 0 to 1, which can be tuned according to different requirements.

We use a systematic sampling method to select the seeds for Voronoi diagram generation [28], in which every (*ith,jth*) pixel in a 2D sequence is selected. This guarantees Voronoi seeds are evenly distributed in the imagery dataset. Although the Voronoi seeds can be selected through optimization techniques [29], we choose systematic sampling due to its simplicity.

Figure 3-1 shows how we use Fortune's sweepline algorithm to generate the Voronoi regions. In Figure 3-1 (a), the sweepline moves in the opposite direction of the incoming data stream (if they move in the same direction, it will always be the same pixels on the sweepline). We use a duplicate sweepline method to reduce the influence of cut borders (Figure 3-1 (b)). The duplicate sweepline takes 50% of each neighboring stream to forge a new input stream. We then remove duplicated tiles and partial ones. The algorithm does not change the generated Voronoi image tiles, since a Voronoi diagram is unique given the set of seeds [25]. Sample Voronoi tiles are shown in Figure 3-1 (c) and we can see that the cutting borders coincide with some object boundaries. Although a tile size may differ, Voronoi regions allow us to maintain a similarity of object level complexity.

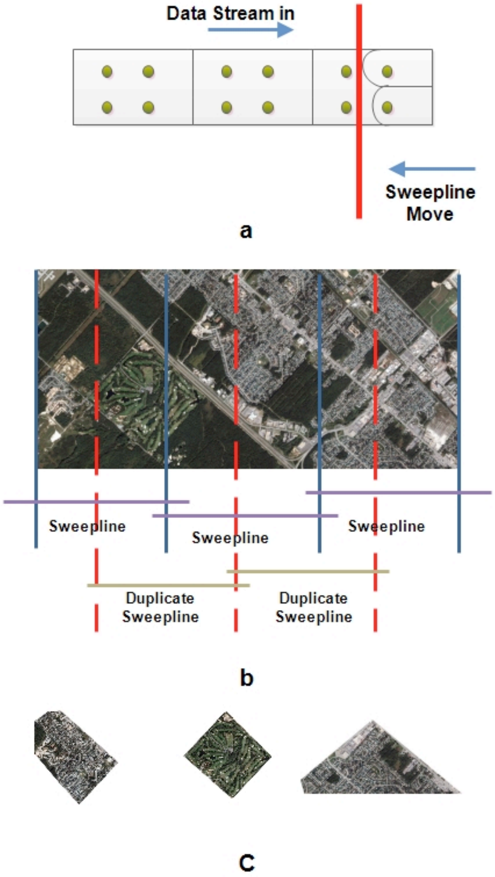

Figure 3-1 (a) Fortune's algorithm with streaming data modeling; (b) duplicate sweepline algorithm; (c) sample tiles obtained using our Voronoi splitting method

We implement the system with the MapReduce framework [30]. MapReduce offers scalability and reliability for parallel computing support. The *map* phase reads data via the streaming services and then splits large RS imagery datasets using our sampling based Voronoi splitting algorithm with the duplicate sweepline method. The *reduce* phase removes duplicated tiles and merges partial ones. Our Voronoi splitting method minimizes the cutting border influence and balance the object level complexity with MapReduce implementation.

4. PERFORMANCE EVALUATION

We use object-based classification to compare the performance of our Voronoi splitting method and the rectangle based partitioning method. (Our performance evaluation, where the analysis would normally occur, also utilizes MapReduce in a distributed environment.) Approximately 1 TB of AVIRIS imagery datasets of North America collected in 2013 (courtesy of NASA/JPL-Caltech) are utilized as the testing data, with 15m spatial resolution and 224 bands (365nm to 2496nm) spectral resolution. We set different tile sizes (from 50 MB to 1GB) to evaluate the influence of different splitting method on classification accuracy. The accuracy is represented as the average accuracy [31] of different classes. We use seven classes: forest, grass, farmland, bare ground, water, roads and buildings.

We use the distributed computing environment in Roger Tomlinson Laboratory of McGill University as the test bed. We initialized 10 *map* nodes and 2 *reduce* nodes for our Voronoi splitting method (α =0.4); whereas 10 identical *map* nodes (the same number of splittings) are used for the rectangle based partitioning. All the image tiles are stored on a 2 TB network hard disk for classification evaluation on one Dell laptop running eCognition® [20].

Table 4-1 Performance Comparison of Voronoi Splitting and Rectangle based Splitting Method

Tile Size	Rectangle based Splitting	Voronoi Splitting
1GB	92.1%	92.3%
500MB	89.3%	90.7%
200MB	87.9%	89.6%
100 MB	84.6%	89.0%
50 Mb	79.2%	87.7%

Table 4-1 shows the evaluation results. With the increase of tile numbers (or decrease of tile size), the classification accuracy of rectangle based partitioning method decreases faster than our Voronoi based splitting method. When the tile size becomes 50MB, the rectangle splitting method generates considerable noisy geometric and topological information due to cut borders. Evaluation shows Voronoi splitting method outperforms traditional rectangle partitioning method by combining spectral and spatial distance.

5. CONCLUSION

We have presented a sampling based Voronoi splitting method for big RS imagery dataset object-based analysis, which also models datasets as a stream. Compared to the rectangle based partitioning method, our Voronoi splitting method better balances object level complexity and lessens distortions due to cutting borders. The streaming modeling of big data can represent objects in big imagery data and is compatible with the sweepline based Voronoi algorithm and MapReduce framework. Evaluation with different tile size configuration demonstrates that our method outperforms the rectangle based partitioning method.

We plan to improve the performance of our method with three approaches. First, the seeds of a Voronoi diagram can be selected using optimization method as centroids of objects [32]. Second, we plan to explore Apache Storm [33] to replace MapReduce, which provides better real-time parallel computing support for data streaming. Finally, our work presents a big data solution for data decomposition. Recomposing distributed analyses also demands research.

6. REFERENCES

[1] S. Fortune, "A sweepline algorithm for Voronoi diagrams," Algorithmica, vol. 2, no. 1–4, pp. 153–174, 1987.

[2] G. J. Hay and G. Castilla, "Geographic Object-Based Image Analysis (GEOBIA): A new name for a new discipline," in Object-based image analysis, Springer, 2008, pp. 75–89.

[3] S. LaValle, E. Lesser, R. Shockley, M. S. Hopkins, and N. Kruschwitz, "Big data, analytics and the path from insights to value," MIT Sloan Management Review, vol. 21, 2014.

[4] A. Silberstein, A. Machanavajjhala, and R. Ramakrishnan, "Feed following: the big data challenge in social applications," in Databases and Social Networks, 2011, pp. 1–6.

[5] D. Howe, M. Costanzo, P. Fey, T. Gojobori, L. Hannick, W. Hide, D. P. Hill, R. Kania, M. Schaeffer, and S. St Pierre, "Big data: The future of biocuration," Nature, vol. 455, no. 7209, pp. 47–50, 2008.

[6] D. Sui and M. Goodchild, "The convergence of GIS and social media: challenges for GIScience," International Journal of Geographical Information Science, vol. 25,no.11, pp.1737–1748, 2011.

[7] A. Zaslavsky, C. Perera, and D. Georgakopoulos, "Sensing as a service and big data," arXiv preprint arXiv:1301.0159, 2013.

[8] Z. Chen, N. Chen, C. Yang, and L. Di, "Cloud Computing Enabled Web Processing Service for Earth Observation Data Processing," Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal of, vol. 5, no. 6, pp. 1637–1649, 2012.

[9] R. G. Congalton, "Remote Sensing and Geographic Information System Data Integration: Error Sources and Research Issues," Photogrammetric Engineering & Remote Sensing, vol. 57, no. 6, pp. 677–687, 1991.

[10] S. Lang, "Object-based image analysis for remote sensing applications: modeling reality–dealing with complexity," in Object-based image analysis, Springer, 2008, pp. 3–27.

[11] S. Wang and M. P. Armstrong, "A quadtree approach to domain decomposition for spatial interpolation in grid computing environments," Parallel Computing, vol. 29, no. 10, pp. 1481–1504, 2003.

[12] B. Li, H. Zhao, and Z. Lv, "Parallel ISODATA Clustering of Remote Sensing Images Based on MapReduce," in 2010 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), 2010, pp. 380–383.

[13] S. Liang, S. Chen, C. Huang, R. Li, Y. Chang, J. Badger, and R. Rezel, "Geocens: Geospatial cyberinfrastructure for environmental sensing," in Proceedings of GIScience 2010—Sixth international conference on Geographic Information Science, 2010, vol. 6292.

[14] C. Yang, M. Goodchild, Q. Huang, D. Nebert, R. Raskin, Y. Xu, M. Bambacus, and D. Fay, "Spatial cloud computing: how can the geospatial sciences use and help shape cloud computing?," International Journal of Digital Earth,vol.4,no. 4,pp.305–329,2011.

[15] P. Wang, J. Wang, Y. Chen, and G. Ni, "Rapid processing of remote sensing images based on cloud computing," Future Generation Computer Systems, vol. 29, no. 8, pp. 1963–1968, Oct. 2013.

[16] X. Huang and J. R. Jensen, "A machine-learning approach to automated knowledge-base building for remote sensing image analysis with GIS data," Photogrammetric Engineering and Remote Sensing, vol. 63, no. 10, pp. 1185–1193, 1997.

[17] A. Plaza, D. Valencia, J. Plaza, and P. Martinez, "Commodity cluster-based parallel processing of hyperspectral imagery," Journal of Parallel and Distributed Computing, vol. 66, no. 3, pp. 345–358, 2006.

[18] Montreal Satellite StreetView, 3_305AUG_02APR-MONTREAL_1-S3XM. Markham ON: DMTI Spatial Inc., 2006.

[19] T. Blaschke, "Object based image analysis for remote sensing," ISPRS journal of photogrammetry and remote sensing, vol. 65, no. 1, pp. 2–16, 2010.

[20] M. Baatz, U. Benz, S. Dehghani, M. Heynen, A. Höltje, P. Hofmann, I. Lingenfelder, M. Mimler, M. Sohlbach, and M. Weber, "eCognition user guide," Munich, Definiens, 2000.

[21] M. Russell, Mining the Social Web, O'Reilly Media, 2013

[22] J. Ahrens, K. Brislawn, K. Martin, B. Geveci, C. C. Law, and M. Papka, "Large-scale data visualization using parallel data streaming," IEEE Computer Graphics and Applications, vol. 21, no. 4, pp. 34–41, Jul. 2001.

[23] C. Hu, J. Tian, D. Ming, and D. Shen, "Remote visualization for large terrain surfaces based on parallel streaming pipeline architecture," in Proceedings of SPIE - The International Society for Optical Engineering, 2007, vol. 6789.

[24] T. Fountain, S. Tilak, P. Shin, P. Hubbard, and L. Freudinger, "The Open Source DataTurbine Initiative: Streaming data middleware for environmental observing systems," in Proceedings, 33rd International Symposium on Remote Sensing of Environment, ISRSE 2009, 2009, pp. 1124–1129.

[25] F. Aurenhammer, "Voronoi diagrams—a survey of a fundamental geometric data structure," ACM Computing Surveys (CSUR), vol. 23, no. 3, pp. 345–405, 1991.

[26] A. Akdogan, U. Demiryurek, F. Banaei-Kashani, and C. Shahabi, "Voronoi-based geospatial query processing with mapreduce," in Cloud Computing Technology and Science (CloudCom), 2010 IEEE Second International Conference on, 2010, pp. 9–16.

[27] Y. Li, J. Li, and M. A. Chapman, "Segmentation of SAR intensity imagery with a Voronoi tessellation, Bayesian inference, and reversible jump MCMC algorithm," Geoscience and Remote Sensing, IEEE Transactions, vol. 48, no. 4, pp. 1872–1881, 2010.

[28] S. V. Stehman, "Comparison of systematic and random sampling for estimating the accuracy of maps generated from remotely sensed data," Photogrammetric Engineering and Remote Sensing, vol. 58, no. 9, pp. 1343–1350, 1992.

[29] N. Aziz, A. W. Mohemmed, and B. D. Sagar, "Particle swarm optimization and Voronoi diagram for wireless sensor networks coverage optimization," in Intelligent and Advanced Systems, 2007. ICIAS 2007. International Conference, 2007, pp.961–965.

[30] J. Dean and S. Ghemawat, "MapReduce: simplified data processing on large clusters," Communications of the ACM, vol. 51, no. 1, pp. 107–113, 2008.

[31] R. G. Congalton, "A review of assessing the accuracy of classifications of remotely sensed data," Remote sensing of environment, vol. 37, no. 1, pp. 35–46, 1991.

[32] S. S. Khan and A. Ahmad, "Cluster center initialization algorithm for K-means clustering," Pattern recognition letters, vol. 25, no. 11, pp. 1293–1302, 2004.

[33] Apache Storm, http://storm.incubator.apache.org/