

**Parallel Geospatial Data Management for Multi-Scale Environmental Data  
Analysis on GPUs**

**2013 DOE Visiting Faculty Program Project Report**

**By**

**Jianting Zhang** (Visiting Faculty)

(Department of Computer Science, the City College of New York)

**Dali Wang** (DOE Lab Host)

(Climate Change Science Institute, Oak Ridge National Laboratory)

08/09/2013

## Abstract

In-situ and/or post- processing large amount of geospatial data increasingly becomes a bottleneck in scientific inquires of Earth systems and their human impacts as spatial and temporal resolutions increase and more environmental factors and their physical processes are being incorporated. In this study, we have developed a set of parallel data structures and algorithms that are capable of utilizing massively data parallel computing power available on commodity Graphics Processing Units (GPUs) for a popular geospatial technique called Zonal Statistics. Given a raster and a polygon input layers, our technique has four steps in computing the histograms of raster cells that fall within polygons. Each of the following four steps are mapped to GPU hardware by identifying its inherent data parallelisms (1) dividing an input raster into blocks and compute per-block histograms, (2) pairing raster blocks with polygons and determining inside/intersect raster blocks for each polygon, (3) aggregating per-block histograms to per-polygon histograms for inside raster blocks (4) updating polygon histograms for raster cells that are inside respective polygon through point-in-polygon test by treating raster cells in intersecting raster blocks as points. In addition, we have utilized a Bitplane Quadtree (BQ-Tree) based technique to decode encoded rasters to significantly reduce disk I/O and CPU-GPU data transfer times.

Experiment results have shown that our GPU-based parallel Zonal Statistic technique on 3000+ US counties (polygonal input) over 20+ billion NASA SRTM (Shuttle Radar Topography Mission) 30 meter resolution Digital Elevation (DEM) raster cells has achieved impressive end-to-end runtimes: 101 seconds and 46 seconds on a low-end workstation equipped with a Nvidia GTX Titan GPU using cold and hot cache, respectively; and, 60-70 seconds using a single OLCF Titan computing node and 10-15 seconds using 8 nodes. The results clearly show the potentials of using high-end computing facilities for large-scale geospatial processing and can serve as a concrete example of designing and implementing frequently used geospatial data processing techniques on new parallel hardware to achieve desired performance. The project outcome can be used to support DOE BER's overarching mission to "understand complex biological and environmental systems across many spatial and temporal scales"

## Introduction

As spatial and temporal resolutions of Earth observatory data and Earth system simulation outputs are getting higher, in-situ and/or post-processing such large amount of geospatial data increasingly becomes a bottleneck in scientific inquires of Earth systems and their human impacts. Existing geospatial techniques that are based on outdated computing models (e.g., serial algorithms and disk-resident systems), as have been implemented in many commercial and open source packages, are incapable of processing large-scale geospatial data and achieve desired level of performance.

One of the most significant technical trends in parallel computing is the increasingly popular General-purpose computing on Graphics Processing Units (GPGPU) technologies. High-end GPGPU devices, such as Nvidia GPUs based on its Kepler architecture [1], have thousands of cores and provide more than 1 teraflops double precision floating point computing power and hundreds of GB/s memory bandwidth. GPU clusters that are made of multiple GPU-equipped computing nodes, such as OLCF's Titan supercomputer [2], are enormously powerful for scientific investigations. Unfortunately, while single GPU devices and GPU clusters have been extensively used in many computing intensive disciplines [3], they have yet been widely utilized for geospatial computing although their potentials are well recognized [4]. GPUs typically adopt a shared-memory architecture, however, they require a different set of techniques from the traditional ones in order to be fully utilized. While this generally makes it difficult to adapt traditional geospatial techniques for GPUs, it also brings an opportunity to develop new high-performance techniques by synergized exploiting modern hardware features, such as large CPU memory capacity, parallel data structures and algorithms and GPU hardware accelerations.

Among various geospatial techniques that are required by multi-scale environmental data analysis, a frequently used one is called Zonal Statistics [5]. Given two input datasets with one representing measurements (e.g., temperature or precipitation) and the other one representing polygonal zones (e.g., ecological or administrative zones), Zonal Statistics computes major statistics and/or complete distribution histograms of the measurements in all polygonal zones. The geospatial computing task is both computing intensive and data intensive, especially when the number of measurements is large and the zonal polygons are complex. While several commercial and open source implementations of Zonal Statistics are currently available, such as ESRI ArcGIS [6], very few of them have been parallelized and used in cluster computers. To the best of our knowledge, there is no previous work on parallelizing Zonal Statistics on GPUs. In this study, we have developed a set of parallel data structures and algorithms that are capable of utilizing massively data parallel computing power available on commodity GPUs and are scale well on cluster computers.

Results have shown that our GPU-based parallel Zonal Statistic technique on 3000+ US counties over 20+ billion NASA SRTM (Shuttle Radar Topography Mission) 30 meter resolution Digital Elevation (DEM) raster cells [7], which would take hours to compute in a traditional Geographical Information System (GIS) environment, has achieved impressive end-to-end runtimes: 101 seconds and 46 seconds on a low-end workstation equipped with a Nvidia GTX Titan GPU using cold and hot cache, respectively; and, 60-70 seconds using a single OLCF Titan computing node and 10-15 seconds using 8 nodes. Our experiment results clearly show the potentials of using high-end computing facilities for large-scale geospatial processing.

## Progress

Our technique has four steps and each step can be mapped to GPU hardware by identifying its inherent data parallelisms. First, a raster is divided into blocks and per-block histograms are derived. Second, the Minimum Bounding Boxes (MBRs) of polygons are computed and are spatially matched with raster blocks; matched polygon-block pairs are tested and blocks that are either inside or intersect with polygons are identified [8]. Third, per-block histograms are aggregated to polygons for blocks that are completely within polygons. Finally, for blocks that intersect with polygon boundaries, all the raster cells within the blocks are examined using point-in-polygon-test [9]. Raster cells that are within polygons are used to update corresponding histograms. The overall framework is illustrated in Fig. 1.

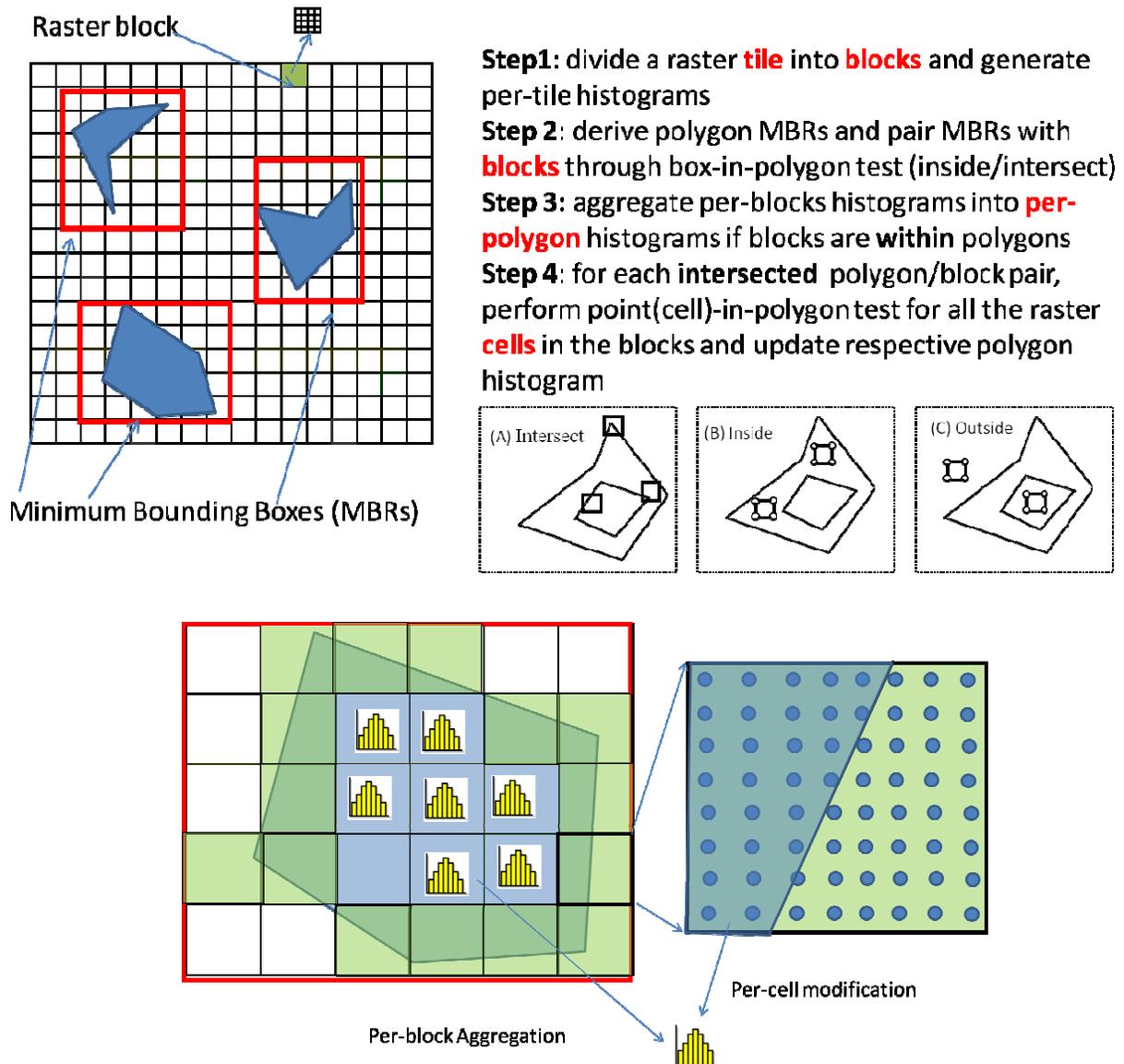


Fig. 1 Illustration of the Framework of the Proposed GPU-based Parallel Zonal Statistics Technique

Paring raster blocks with polygons essentially applies a grid-based spatial indexing technique which significantly reduces the number of required point-in-polygon tests that is needed to assign raster cells to polygons. The four steps are highly data parallelizable and can be efficiently implemented on modern GPUs. Our results have shown that, after applying spatial indexing and GPU hardware acceleration, Parallel Zonal statistics using the NASA SRTM raster data and Continental United States(CON-US) county data becomes I/O bound. Reading the 38GB data from disk took more than 400 seconds which is several times larger than computing time using the new technique. To further reduce end-to-end runtimes, we have applied our Bitplane Quadtree (BQ-Tree) technique [10] that was initially developed for coding raster data. The idea of the technique is to separate an M-bit raster into M binary bitmaps and then use a modified quadtree technique to encode the bitmaps (Fig. 2). As many neighboring raster cells are similar, there is a high chance that quadrants of the M binary bitmaps are uniform and can be efficiently encoded using quadtrees. Our experiments have shown that the data volume of the encoded NASA SRTM raster data is about 7.3 GB, which is only about 1/5 of the original data volume. As a result, the BQ-Tree coding technique has successfully reduced I/O times by 5 times. As a comparison, the data volumes of TIFF-based and gzip-based compression are 15 GB and 8.3 GB, respectively. Interestingly, when applying gzip-based compression on BQ-Tree encoded NASA SRTM raster data, the data volume is further reduced to 5.5 GB.

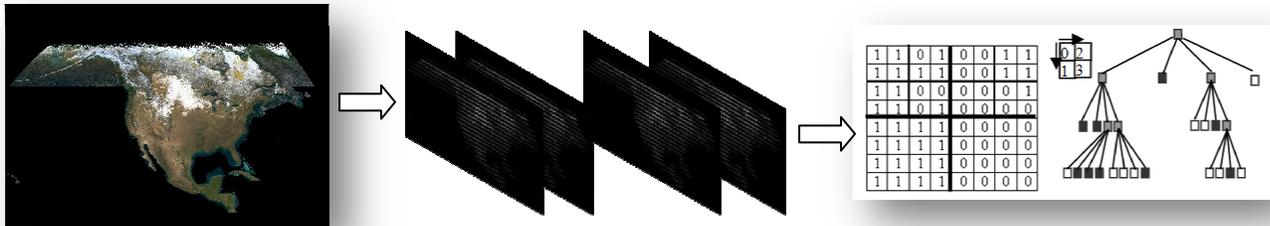


Fig. 2 Illustration of BQ-Tree Coding of Rasters

We have set up three experiment environments to test the efficiency and scalability of the proposed technique. The first two environments are single-node configurations that use a Fermi-based (Quadro 6000) and a Kepler (GTX Titan) GPU device, respectively. Note that both devices have 6 GB GPU device memory. The third experiment environment is the OLCF Titan GPU cluster and we have varied the numbers of computing nodes from 1 to 16 after chunking the input rasters into 36 tiles. The results of the single-node and the cluster configurations are shown in Table 1 and Table 2, respectively. From Table 1, we can see that the Kepler-based GPU device delivers significantly higher throughputs (lower runtimes) than the Fermi-based one. When looking into the runtimes of individual components, including raster decoding and the four steps in Fig. 1, the Kepler-based GPU device is about 2X faster than the Fermi-based one, especially for Step 4 that is mostly computing intensive. This is expected as Kepler-based GPU devices have larger number of GPU cores and higher memory bandwidth. From Table 2 we can see that our technique scales well when the number of computing nodes varies from 1 to 16 on OLCF Titan supercomputer. This indicates that our technique can be used for large-scale geospatial computing applications that are increasingly required in environmental data analysis.

Table 1 End-to-End Runtimes of Single Node Configurations

	Cold Cache	Hot cache
Single Node Config1 (Quadro 6000)	180s	78s
Single Node Config2 (GTX Titan)	101s	46s

Table 2 Scalability Test on OLCF Titan GPU Cluster

# of computing nodes	1	2	4	8	16
end-to-end runtime(s)	60.7	31.3	17.9	10.2	7.6

While it is typical that analyzing climate simulation outputs relies on GIS software in a post-processing manner, which is neither efficient nor scalable, our study has shown that it is quite feasible to develop new parallel geospatial processing techniques that can be embedded into large-scale climate simulation modeling process and executed efficiently on high-end GPU clusters, such as OLCF Titan. The project outcome can be used to support DOE BER’s overarching mission to “understand complex biological and environmental systems across many spatial and temporal scales” [11].

## Future Work

The pilot study opens many promising future work directions. First of all, we plan to further optimize the implementations of both raster coding and the four steps in Zonal Statistics and push the performance limit on a single GPU device. Second, as raster sizes can significantly impact load balancing among multiple computing nodes, we plan to utilize domain-specific metadata to achieve better load balancing and improve the overall performance. Finally, we observe that several techniques utilized in this study, such as BQ-Tree based raster coding and paring raster blocks with polygons, can be equally applied to realize other geospatial processing tasks. We would like to prioritize such tasks based on their relevance to large-scale climate and Earth system modeling, and, implement them on GPU clusters (such as Titan) to significantly boost modeling performance and support interactive data exploration and decision making.

## Conclusions

Partially supported by the DOE VFP program, we have successfully developed a set of parallel data structures and algorithms and implemented a high-performance Zonal Statistics technique on OLCF’s Titan GPU cluster. Using NASA SRTM 30m DEM data, our technique is capable of computing elevation distribution histograms for all continental United States counties in about 100 seconds (end-to-end runtime including disk I/O) using a single GPU device and about 10 seconds using 8 computing nodes on Titan. Our experiments have demonstrated the level of achievable high-performance on high-end computing facilities when compared to traditional geospatial processing pipelines. The pilot study may suggest that higher throughputs

of large-scale Earth system modeling are quite possible by incorporating advanced data management techniques and fully utilizing parallel hardware capabilities.

## References

1. <http://www.nvidia.com/object/nvidia-kepler.html>
2. <http://www.olcf.ornl.gov/titan/>
3. Hwu, W.-M. W (eds.), 2011. GPU Computing Gems: Emerald & Jade Editions. Morgan Kaufmann
4. Zhang, J., 2010. Towards Personal High-Performance Geospatial Computing (HPC-G): Perspectives and a Case Study. Proceedings of the ACM SIGSPATIAL International Workshop on High Performance and Distributed Geographic Information Systems (HPDGIS).
5. Theobald, D. M., 2005. GIS Concepts and ArcGIS Methods, 2nd Ed., Conservation Planning Technologies, Inc
6. <http://www.esri.com/software/arcgis>
7. <http://www2.jpl.nasa.gov/srtm/>
8. Zhang, J. and You, S., 2013. Parallel Zonal Summations of Large-Scale Species Occurrence Data on Hybrid CPU-GPU Systems. Technical Report available at [http://www-cs.ccny.cuny.edu/~jzhang/zs\\_gbif.html](http://www-cs.ccny.cuny.edu/~jzhang/zs_gbif.html).
9. Zhang, J. and You, S., 2012. Speeding up Large-Scale Point-in-Polygon Test Based Spatial Join on GPUs. Proceedings of ACM SIGSPATIAL International Workshop on Analytics for Big Geospatial Data Workshop (BigSpatial).
10. Zhang, J., You, S. and Gruenwald, L., 2011. Parallel Quadtree Coding of Large-Scale Raster Geospatial Data on GPGPUs. Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (GIS).
11. <http://science.energy.gov/ber/>