Parallel Processing over Spatial-Temporal Datasets from Geo, Bio, Climate and Social Science Communities: A Research Roadmap

Sushil K. Prasad, Danial Aghajarian, Michael McDermott and Dhara Shah
Department of Computer Science
Georgia State University
sprasad@gsu.edu

Mohamed Mokbel
Computer Science and Engineering
University of Minnesota
mokbel@umn.edu

Satish Puri
Mathematics, Statistics and Computer Science
Marquette University
satish.puri@marquette.edu

Sergio J. Rey
Center for Spatial Sciences
University of California Riverside
sergio.rey@ucr.edu

Shashi Shekhar and Yiqun Xe
Department of Computer Science
University of Minnesota
shekhar@cs.umn.edu

Ranga Raju Vatsavai
Department of Computer Science
North Carolina State University
rvatsavai@ncsu.edu

Fusheng Wang, Yanhui Liang and Hoang Vo
Biomedical Informatics Department
Stony Brook University
fusheng.wang@stonybrook.edu

Shaowen Wang
Geography & Geographic Information Science
University of Illinois at Urbana-Champaign
shaowen@illinois.edu

***Abstract*—This vision paper reviews the current state-of-art and lays out emerging research challenges in parallel processing of spatial-temporal large datasets relevant to a variety of scientific communities. The spatio-temporal data, whether captured through remote sensors (global earth observations), ground and ocean sensors (e.g., soil moisture sensors, buoys), social media and hand-held, traffic-related sensors and cameras, medical imaging (e.g., MRI), or large scale simulations (e.g., climate) have always been "big." A common thread among all these big collections of datasets is that they are spatial and temporal. Processing and analyzing these datasets requires high-performance computing (HPC) infrastructures. Various agencies, scientific communities and increasingly the society at large rely on spatial data management, analysis, and spatial data mining to gain insights and produce actionable plans. Therefore, an ecosystem of integrated and reliable software infrastructure is required for spatial-temporal big data management and analysis that will serve as crucial tools for solving a wide set of research problems from different scientific and engineering areas and to empower users with next-generation tools. This vision requires a multidisciplinary effort to significantly advance domain research and have a broad impact on the society. The areas of research discussed in this paper include (i) spatial data mining, (ii) data analytics over remote sensing data, (iii) processing medical images, (iv) spatial econometrics analyses, (v) Map-Reduce-based systems for spatial computation and visualization, (vi) CyberGIS systems, and (vii) foundational parallel algorithms and data structures for polygonal datasets, and why HPC infrastructures, including harnessing graphics accelerators, are needed for time-critical applications.**

***Keywords*-High performance computing, Spatial data mining, Remote sensing data, Medical images, Spatial econometrics, Map-reduce systems, CyberGIS, Parallel algorithms and data structures.**

I. Introduction

Public and private sector agencies rely on spatial data management, analysis, and spatial data mining to gain insights and produce actionable plan. Some of the application domains include public health, climate and environment science, transportation, urban planning and engineering. Some of the agencies that use spatial analysis in decision making include National Institute of Health (NIH), US Department of Transportation, US Department of Agriculture, NASA, and National Oceanic and Atmospheric Administration [79].

Exemplar applications include forest fire or hurricane simulation, where multiple layers of spatial data needs to be joined and overlaid to predict the affected areas and rescue shelters. These disaster response scenarios call for leveraging high performance computing techniques to yield real-time results. In the biomedical domain, spatial query and join algorithms are used to analyze digital pathology images containing millions of cells [100]. The derived data from these images measured are in terabytes and its efficient analysis enables more effective diagnosis and prediction of cancer and other important diseases. In the public health domain, spatial statistics is used to find significant hotspots

978-1-5386-1996-4/17 $31.00 © 2017 IEEE
DOI 10.1109/BigDataCongress.2017.39

related to disease outbreak. Similarly, law enforcement use hotspot maps to find areas with high criminal activities. In transportation, it is used to find sections of highway with higher rates of accidents reported [79].

State of Art and Limitations

The computational context facing researchers and savvy tool users in different scientific domains that deal with geo, bio, social and other Spatio-Temporal (**ST**) data is one of rapidly growing sources of data and traditional computational and analytical tools that were designed for the desktop era. As a result, much of the software stack in spatial analysis is rather ill-suited to the emerging realities of big data. Simply put, research is currently tool-constrained in two ways. First, the problem size that can be addressed is severely limited by the existing computational capacity in ST analysis. Second, the new types of problems, research, and decisions that are afforded by big data are not the type envisaged by the designers of desktop based scientific software and are thus currently beyond reach.

A highly-integrated and reliable software infrastructure ecosystem is required for ST big data management and analysis that will serve as crucial tools for solving a wide set of research problems from different scientific and engineering areas and to empower users with next generation tools. Such an infrastructure is valuable in two ways: first by speeding up desired data management, mining, and analysis projects on the scale and data granularity never available before, and second by enabling new discoveries by the researchers and new ways of planning and decision making by society at large with the novel big-data-oriented tools not available in the standard software on the market. With this vision in mind, a multidisciplinary effort is necessary to enable a broad community of scientists and users to employ high-performance, scalable and open-source software tools for spatial data analytics and computations over massive geo, bio, climate, social and other ST datasets to significantly advance domain research and have broad impact on the society.

Like the paper's vision, this paper itself is a result of an multidisciplinary, multi-institutional collaboration. Our attempt is to capture the current state of art in processing a variety of ST datasets from disparate problem domains and the future research directions. Section II starts us off with an introduction to the broader area of spatial computing and delves into ST data mining techniques. The next three sections are based on three different domains of remote sensing, medical imaging and econometrics. Section III presents work and research challenges in ST data analytics over remote sensing data. Section IV describes a high performance computing (HPC) system over medical image ST datasets. Then, we consider spatial econometrics. Section V examines an open source library for spatial econometrics analyses. Section VII presents the Map-Reduce-based algorithms and systems for spatial computation and visualization. Section VI describes the ongoing CyberGIS system project for GIS community and its frontiers. Finally, in Section VIII, we end with examining algorithms, data structures and system issues and future roadmap, specially for polygonal ST datasets, and why HPC infrastructures, including harnessing graphics accelerators, are needed for time-critical applications.

II. Spatial and Spatio-temporal Data Mining Methods and Their Parallel Formulations

Spatial and spatio-temporal data mining [79, 80] is concerned with the quantification and discovery of interesting, useful and non-trivial patterns families from datasets such as maps, trajectories, remote-sensing images, and the decennial census. Applications of spatial data mining include public health (e.g., cancer cluster detection, predict spread of infectious diseases), public safety (e.g., crime hotspot detection, hurricane trajectory and landfall location prediction), location-based services (e.g., co-location of different service-types), precision agriculture, transportation (e.g., spatial outlier sensors whose traffic measurements are very different from those of neighboring sensors), etc.

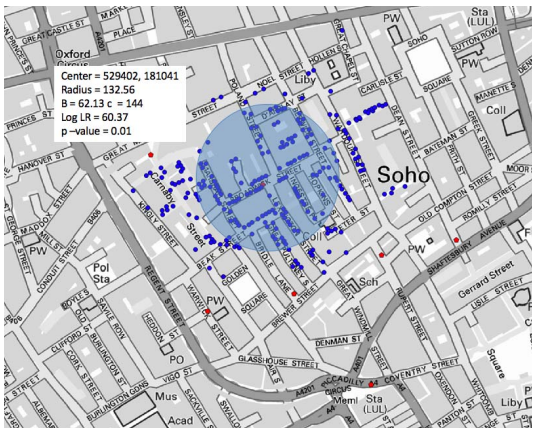


Figure 1. 1854 London Cholera map of mortality (blue dots) and water pumps (red dots). Hotspot analysis identifies the blue circle around the Broad Street water pump. Blue and red dots show locations of mortalities and water pumps respectively. Blue circle shows an hotspot of mortalities

Differences between classical and spatial data mining are similar to the difference between classical and spatial statistics. First, spatial data is embedded in a continuous space, whereas classical datasets are often discrete. Second, the cost of a false positive are often very high for many use cases of spatial data mining. Third, spatial patterns are often local, whereas classical data mining techniques often focus on global patterns. Finally, one of the common assumptions in classical statistical analysis is that data samples are independently generated. When it comes to the analysis of spatial data, however, the assumption about the independence of samples is generally false because spatial data tends to

attern family	GPU	MPI	OpenMP	UPC	MapReduce	Spark
Spatial auto-regression		[48, 106]	[48]	-	-	-
Kriging	-	[64]	-	-	-	-
Prediction	-	-	[75]	[35]	-	-
Colocation	-	-	-	-	[113]	-
Hotspot	[63]	-	-	-	-	-
Outlier	-	[13]	-	-	-	-
Change detection	[66, 67]	-	-	-	-	-

Table I
PATTERN FAMILY AND PARALLEL COMPUTING PLATFORMS

be highly self-correlated. For example, people with similar characteristics, occupation and background tend to cluster together in the same neighborhoods. In spatial statistics, this tendency is called spatial autocorrelation. Ignoring spatial autocorrelation when analyzing data with spatial characteristics may produce hypotheses or models that are inaccurate or inconsistent with the data set. Consequently, classical data mining algorithms often perform poorly when applied to spatial data sets and more powerful methods such as spatial statistics and spatial data mining are needed. Spatial statistics [20, 21] provide measures for spatial auto-correlation (e.g., spatial auto-regressive models), methods for spatial interpolation (e.g., Kriging), theories (e.g., spatial point process), etc. leveraging the neighbor relationship between location-aware data items. Parallel formulations have been explored for computing spatial auto-correlation (e.g., Morans I [11] Getis-Ord [106] and spatial interpolation methods (e.g.Kriging [64]) using MPI, OpenMP [36, 48, 64, 84] as summarized in Table I. In addition, parameter estimation procedures for many spatial statistical models (e.g., spatial auto-regression) use matrix operations and may benefit from parallel formulations of linear algebra algorithms. However, the matrices representing neighbor relationships among spatial data-items are sparse but not necessarily banded, which may require new parallel formulations for operations such as determinant computation.

Spatial data mining explores patterns families such as hotspots, co-locations, location prediction models, spatial outliers, teleconnections, etc. Hotspots represent geographic areas of unusually high concentration of an event such as trees, retail stores, disease, crimes, etc. The 1854 Cholera epidemic in London is a well-known example of hotspot. Figure 1 shows the Cholera mortality locations using blue dots using the data collected by Dr. John Snow. It also shows a hotspot using a blue circle, within which the mortality density is significantly higher than that outside. It passes the statistical significance test with a p-value of 0.01 using a popular hotspot detection software, namely SatScan [2] from the US National Cancer Institute. Hotspot analysis is used widely in public health and public safety to identify hotspots of diseases and crimes respectively. It is important to note the high cost of false positives and true negatives for hotspot detection. Incorrectly labelling a neighborhood to be a hotspot of disease or crime may lead to stigmatization and significant economic loss. On the other hand, missing true hotspots of disease may lead to preventable mortalities and disease burden. To reduce the number of false positives, hotspot detection techniques [2] use statistical significance tests and follow-up manual verifications. Parallel computing algorithms based on GPU have been proposed in [63].

Co-location pattern discovery process finds frequently co-located subsets of spatial event types given a map of their locations. For example, analysis of habitats of animals and plants may identify co-location of predator-prey species, symbiotic species, and fire events with fuel, ignition sources, etc. Readers may find it interesting to analyze the map in Figure 1 to find co-location patterns. The 1854 London Cholera hotspot was collocated with the Broad Street water pump. This generated a new hypothesis that Cholera was spread via water challenging the then-prevalent Miasma (i.e., bad air) theory. Scientists started examining the water from Broad Street water pump using micro-scopes, which subsequently led the Germ theory, a major turning point in modern science. It influenced design of modern cities by introduction of sewer systems and other public health innovations. A MapReduce-based parallel formulation for co-location pattern discovery is explored in [113].

Spatial outliers [81] are significantly different from their neighborhood even though they may not be significantly different from the entire population. For example, a new house in an old neighborhood of a growing metropolitan area is a spatial outlier, even though it is not a global outlier in a metropolitan area. Another example is an Interstate highway traffic sensor, whose measurements were within normal range, however are often very different than those from its upstream and downstream neighbors. A case study [81] with 4000 sensors on highway network in the Minneapolis St. Paul area found that spatial outlier identified malfunctioning sensors. A MPI-based parallel algorithm was proposed to scale up spatial outlier detection [13].

Location prediction is concerned with discovering a model to infer locations of a spatial phenomenon from the maps of other spatial features. For example, climate scientists make land-cover classification maps from remote sensing images.

ecologist build models to predict habitats for endangered species using maps of vegetation, water bodies, climate and other related species. Classical data mining techniques yield weak prediction models as they do not capture the auto-correlation in spatial datasets. In addition, the land-cover classification maps derived from high-resolution imagery using classical methods (e.g., decision trees, random forest) often exhibit salt-and-pepper noise [46], i.e., locations whose predicted land-cover class is very different from the predicted land-cover classes of its neighboring locations. Such problems are reduced significantly by spatial auto-correlation aware location prediction methods such as the spatial auto-regression, Markov random field based Bayesian Classifiers, and Spatial Decision Trees [46]. Both GPU-based and OpenMP-based parallel algorithms were explored for spatial prediction and classification [35, 75].

There are many other interesting, useful and non-trivial patterns of interest in spatial data mining. For example, change detection patterns identify geographic areas where the phenomena of interest have differed significantly over the time-interval of interest. The efficiency of change detection can be accelerated via GPU-based methods [66, 67]. Emerging hotspots aim at detecting disease outbreaks well before it results in a large number of cases. Tele-connection patterns represent interaction across far- away locations. For example, the El Nino (warming of Pacific) affects weather thousands of miles away in mid-western and eastern United States. Interested readers are referred to survey papers on spatial data mining [79, 80] and parallel computing algorithms for GIS [43, 82, 83, 120] for additional details.

III. Big Data Analytics over Remote Sensing Data

Usage of remote sensing data for tactical and reconnaissance dates back to the 2^{nd} World War. Since then multi-spectral remote sensing imagery has been widely used in civilian applications, such as mapping settlements, forests, crops and other natural and man-made objects on the Earth. Remote sensing instruments and sensors have made significant progress over several decades in terms of spatial, spectral and temporal resolutions (Figure 2). As shown in the figure, spatial resolution has improved from 1 KM to sub-meter, and spectral resolution has improved from multi-spectral (4-bands) to hyperspectral (224 bands). These improvements have led to the collection of global scale very high resolution (VHR) data. For example, improvements in temporal resolution allow monitoring biomass [17, 18] on a daily basis. Improvements in spatial resolution allows fine-grained classification of urban settlements [37, 90, 91], damage assessments [65, 89], and critical infrastructure monitoring [88, 92]. Although these improvements are leading to new applications, dealing with increased spatial and temporal resolutions require high-end computing and parallel I/O infrastructures (e.g, GMIL and Citation-KNN [90], and GP-Learning [17, 18]).

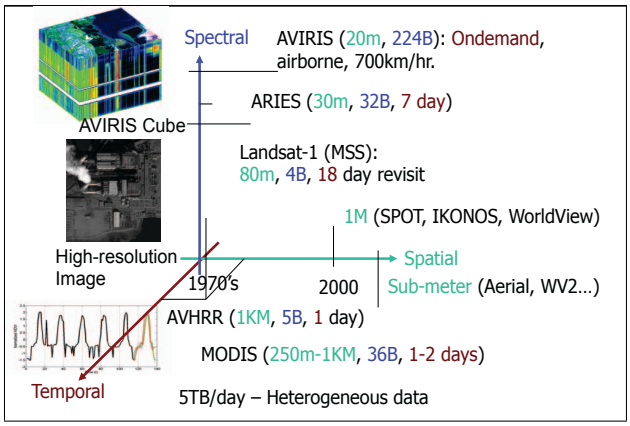


Figure 2. Advances in remote sensing data products (1970 to present)

A concrete example of one of the big data challenges are the MODIS instruments on NASA Terra satellite which acquires two snapshots of the Earth every day. This data is organized in tiles of $10^o \times 10^o$ and each tile consist of 4800×4800 pixels. Biomass monitoring [17, 18] requires processing over 300+ tiles (corresponding to the land coverage). Given that this data is available for the last 16 years on a daily basis, the length of the time series at each location contains over 5700 observations. At a global scale, one would require to process a 7+ billion time series. On the other hand, improved spatial resolution (e.g., 0.5 m) requires processing over 600 trillion pixels for global scale settlement mapping application. The Gaussian Process based monitoring technique [17, 18] shows that these algorithms require $O(T^2)$ memory and $O(T^3)$ time complexity for T data points per pixel (i.e., computation requires inverting 7+ billion covariance matrices, where each matrix is approximately 5700×5700). For settlement mapping, VHR imagery is being used with pixel resolution of 1 meter (m), therefore a MODIS pixel of 250 m^2 is now represented by $250 \times 250 = 62500m^2$ pixels. To process just 1 km^2 image (NY City is roughly 800 km^2), Citation-KNN [90] algorithm requires approximately 27 hours on a standard desktop. Therefore, efficient approaches without sacrificing accuracy are important for these kind of applications.

Global Applications

With the recent launch of satellites by private companies such as Digital Globe (WorldView-2), Planet Labs (Flock of Doves), and SkyBox (recently acquired by Google), applications around very high-resolution (VHR) imagery (sub-meter) are emerging fast. Such imagery provides new opportunities to monitor and map both natural and man made structure across the globe. For past several years, Vatsavai et al. are engaged in developing new approaches to efficiently process these imagery to support applications

of national importance, such as biomass monitoring [17, 18], nuclear proliferation monitoring [88, 92], and settlement mapping [37] at finer spatial and temporal scales. Here is brief description of two applications that captures improved temporal and spatial resolutions.

Biomass Monitoring Using Gaussian Process Learning: Monitoring biomass over large geographic regions for identifying changes is an important task in many applications. With the recent emphasis on biofuel development for reducing dependency on fossil fuels and reducing carbon emissions from energy production and consumption, the landscape of many countries is going to change dramatically in coming years. With the launch of NASA's Terra satellite in December of 1999, with the MODIS instruments aboard, a new opportunity for continuous monitoring of biomass over large geographic regions has emerged. The availability of multi-temporal MODIS imagery has made it possible to study plant phenology, quantitatively describe NPP patterns in time and space, and monitor and map natural resources at regional and global scales. MODIS allows users to identify vegetation changes over time across a region and estimate quantitative biophysical parameters which can be incorporated into global climate models. Even though several cumulative vegetation indices can be found in the literature, MODIS NDVI temporal profiles are widely used in studying plant phenology.

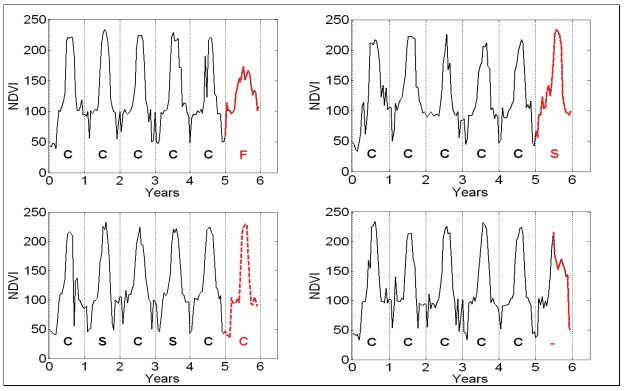


Figure 3. GP-Change Detection Results Using MODIS Time Series Imagery

Vatsavai et al. developed a novel Gaussian Process (GP) based change detection technique [17, 18] that uses MODIS NDVI time series signals to identify changes. As compared to widely used bi-temporal change detection techniques, their change detection technique continuously monitors the biomass using biweekly MODIS NDVI data and updates the change map as soon as new NDVI image is inducted into the system. Though their GP based change detection technique showed improved accuracy over other well-known techniques, the computational complexity (time complexity is $O(n^3)$ and memory is $O(n^2)$) of this technique makes it infeasible for large-scale biomass monitoring studies. However, Vatsavai et al. developed efficient and parallel techniques using shared and distributed memory models which made it possible to apply this technique for continuous monitoring of biomass at continental scales. As an example, GP-based change detection technique was able to identify accurately different types of changes as shown in Figure 3. The first change indicates a corn (C) field is converted into fallow (F) land, the second change indicates a corn field converted into a soybean (S) field, the third change indicates corn and soybean rotation is converted into continuous corn, and finally the fourth change indicates some kind of damage to the corn fields [17, 18].

Settlement Mapping Using Gaussian Multiple Instance Learning: Mapping informal settlements is an important task both from national security and as well as humanitarian grounds. The high rate of urbanization, political conflicts and ensuing internal displacement of population, and increased poverty in the 20^{th} century has resulted in rapid increase of informal settlements. These unplanned, unauthorized, and/or unstructured homes, known as informal settlements, shantytowns, barrios, or slums, pose several challenges to nations as these settlements are often located in the most hazardous regions and lack basic services. Though several World Bank and United Nations sponsored studies stress the importance of poverty maps in designing better policies and interventions, mapping the slums of the world is a daunting and challenging task. Vatsavai et al. developed a computationally efficient and automated framework that is capable of detecting new settlements (especially slums) across the globe.

Most machine learning approaches used for analyzing remote sensing imagery for thematic mapping are single-instance learners (e.g., Bayesian classifiers, Support Vector Machines, Neural Networks, Random Forests). However, with increasing spatial resolution (sub-meter) current satellite images contain much more spatial heterogeneity (rich spatial information). As a result, it is possible to extract more complex classes, such as informal (slums, shanty towns, burrows) settlements from these very high-resolution images. Single instance learning (non-spatial or spatial) is ineffective in such cases due to the fact that the size of the pixel (less than one m^2) is much smaller than the size of the objects (for example, average building size in the US is 250 m^2).

Multi-instance (or Multiple instance) learning (MIL) methods have been developed to overcome some of the limitations of single instance learning schemes. Notable approaches include the seminal work of Dietterich et. al. [23], Diverse Density [60], and Citation-KNN [98]. Recently, MIL algorithms have also been applied to remote sensing image classification as well. For example, in [87] MIL approach is explored for sub-surface land-mine detection using hyperspectral (HS) imagery. In [14], authors have

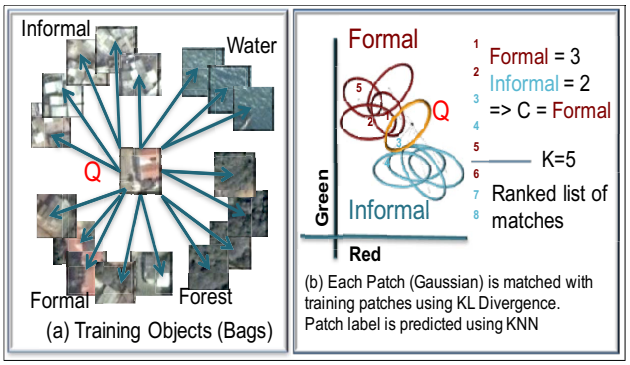


Figure 4. GMIL Framework. (a) Image patch training examples, (b) Prediction of labels for each new patch

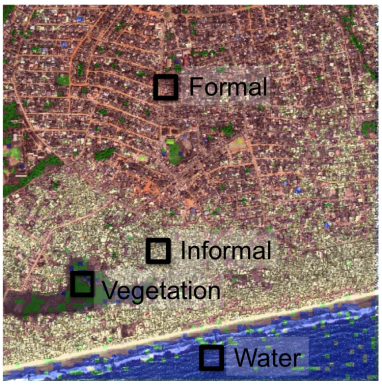


Figure 5. GMIL Classified Image Overlaid on Raw Image

developed MIL based binary classification scheme for identifying targets (landmines) in HS imagery. Gaussian Multiple Instance Learning has also been developed (GMIL) [90] wherein a larger spatial context (image patch) is modeled by a Gaussian distribution, and KL-Divergence is used as the similarity measure for predicting a class label for each patch. This process is shown in Figure 4 and example results are shown in Figure 5.

Outlook

With the advent of Unmanned Aircraft Systems (UAS) and lightweight imaging sensors, acquiring even higher resolution multispectral and hyperspectral imagery as and when required is becoming a reality. As a result, many novel applications at the intersection of food, energy, and water systems can be developed. However, the computational and I/O challenges need to be overcome in order to process the terabytes of remote sensing data generated per day.

IV. High Performance Spatial Queries for 3D Digital Pathology Imaging

The rapid and significant advancement in large-throughput tissue scanning technologies has enabled the production of

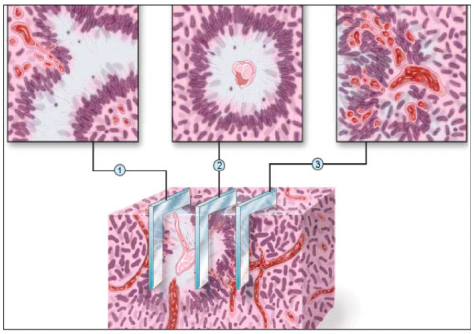


Figure 6. 2D projections of 3D tissue space

pathology imaging big data at cellular and subcellular levels with unprecedented rich information. Digital pathology provides extreme scale quantitative data with high potential for basic research in a wide scope [39, 49, 93], and becomes an emerging technology promising to support computer-aided diagnosis. Importantly, this new field presents salient merits that can help researchers better understand the underlying biological mechanisms of pathological evolutions and disease progressions through quantitative pathology analysis and spatial analytics.

However, almost all prevalent machine-based tissue investigations are bounded by Two-Dimensional phenotypic structure analysis. As 2D projected appearances and spatial profiles of 3D pathologic objects highly depend on the locations and angles of the cutting planes randomly selected during tissue slide preparation process, 3D spatial relationships could be misrepresented and morphological features could be inaccurate after such projection to 2D focal planes.

Recently, three-dimension (3D) digital pathology is made possible through slicing tissues into serial thin sections [53]. The information-lossless 3D tissue space represented by microscopy imaging volumes holds significant potential to enhance biomedical studies through high-performance 3D image analysis and spatial analytics.

Quantitative analyses of 3D pathology images involve both deriving 3D micro-anatomic objects and their features and exploring spatial relationships among a massive number of pathology objects. However, this is challenged by the overwhelming data scale and 3D pathology complexity. Specifically, problems of 3D data explosion (both voxel and structural data), complex histology structures (such as blood vessels), multiple levels of detail for representations, and high computational complexity for both image analysis and spatial queries/geometric computations need to be addressed. There is also a major gap to make the software tools readily executing on large computing platforms such as commodity clusters or public clouds.

Spatial Database Management Systems have been developed for managing and querying 3D spatial data in industrial applications, such as OracleSpatial, MapInfo Discover 3D,

and ESRI 3D GIS. However, they mainly support simple 3D spatial objects like landmarks and buildings with primitive 3D geometry types (solid and its variations). Data loading is also a major bottleneck for SDBMS based solution, especially for large-scale datasets. Moreover, traditional spatial indexing and querying methods have limited support for efficient spatial queries on 3D complex structured objects. Recently, several systems have been proposed to support large-scale spatial queries with distributed computing resources using MapReduce [7, 12, 32, 115]. However, they mainly focus on 2D data and lack critical components for 3D support.

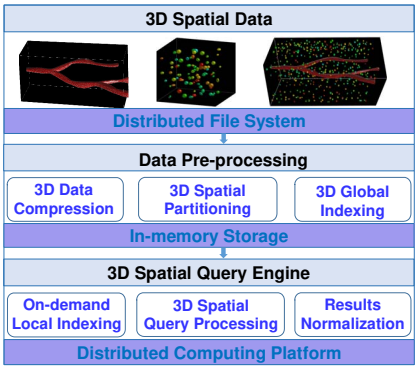


Figure 7. Architecture overview of in-memory based 3D spatial queries

This motivates the need to create a scalable and effective 3D digital pathology analytics framework for large-scale 3D pathology imaging data, and make the software available in multiple distributed computing platforms for convenient use. With derived 3D spatial data from image analysis, the goal is to provide a highly efficient and scalable 3D spatial data management and querying system to discover spatial relationships and patterns, which can be readily run on Apache big data platforms such as Hadoop and Spark. Due to the extreme data scale, one major goal is to mitigate potential high I/O and communication cost, exploit indexing techniques for complex objects to accelerate queries, and provide scalability to run on clusters or computing clouds. Fusheng Wang's group has proposed *iSPEED* [51, 52], an efficient and scalable in-memory based spatial query processing system for large-scale 3D data. To achieve low latency, iSPEED stores data in memory in a highly compressed form using an effective progressive compression approach that compresses each 3D object individually with successive levels of detail. To minimize search space and computation cost, iSPEED provides global spatial indexing in memory through partitioning at subspace level and partitioned cuboid level. iSPEED provides an in-memory 3D spatial query engine *INTENSE*, which can be invoked on-demand for running many instances in parallel. The parallelization of queries is implemented in, but not limited to, Map-Reduce. At run time, iSPEED dynamically decompresses only needed 3D objects at the specified level of detail and creates necessary spatial indexes in-memory to accelerate query processing, such as on-demand object-level indexing and structural indexing on complex structured objects. The salient features of the iSPEED system are summarized as follows.

- The 3D data compression approach makes it possible to significantly reduce data size to have them in memory at very low memory footprint with effective compression and on-demand decompression, which leads to much reduced I/O and communication cost for query processing.
- It models 3D objects with multiple levels of detail for spatial queries, which provides options for users to decide their goals for faster queries or higher accuracy to meet application specific requirements.
- It provides Multi-level in-memory spatial indexing to reduce search space and accelerate queries. In particular, it has unique structural indexing for searching with complex structured objects, which significantly improves query performance compared to traditional MBB based indexing.
- iSPEED has on-demand in-memory based 3D spatial query engine that fully takes advantage of multi-level indexing and data decompression for processing multiple types of spatial queries, which can be implemented with Map-Reduce or other distributed computing paradigms.
- Their preliminary study demonstrates that iSPEED achieves significant benefits on efficiency and scalability of spatial queries over traditional Map-Reduce based query systems.

Future Roadmap

With continuing development of 3D pathology imaging technologies, there will be much increased complexity of spatial analytics due to increasing scale of data with improved resolutions, the heterogeneity of 3D pathology object structures and relationships, and complex spatial patterns to explore. One future direction is to explore parallelization of complex spatial analytics methods such as 3D spatial clustering and comparison of spatial point patterns [15]. We also expect that GPU accelerated spatial querying methods [101] could significantly improve the throughput. Integrating GPU based 3D spatial processing into our framework will be among our future work.

V. Spatial Econometrics and Exploratory Spatial Data Analysis

Spatial econometrics concerns the confirmatory modeling of spatially referenced data employing econometric methods. The closely related field of exploratory spatial data analysis (ESDA) consists of complimentary statistical methods designed to support model free interrogation of geographical

data. In both cases, the geographically referenced nature of the data poses challenges for the application of traditional econometric and exploratory data analysis (EDA) techniques. These largely relate to the notions of spatial heterogeneity and spatial dependence. The former pertains to instability in the model or process over space, while the latter reflects the lack of independence between nearby observations.

PySAL is an open source library of methods for spatial econometrics and exploratory spatial data analysis developed by Rey's group [74]. Written in the Python and released under the BSD License, PySAL is organized as a suite of modules that target different levels of the spatial data analysis research stack as shown in Figure 8. Modular by design, PySAL supports flexible reuse and combinations of its components for specialized applications. The development of the individual modules themselves represents the implementation of advanced spatial analytic and econometric methods that have been developed by Rey's group as well as the broader spatial analysis community.

Figure 8. PySAL: Python Spatial Analysis Library [76]

The main emphasis of PySAL is on vector based geospatial data. A common use case here is the statistical analysis of attributes distributed over areal units, such as census tracts. To enable a consideration of spatial dependence a formal representation of neighbor relations is afforded by the *spatial weights* module. For a given set of n polygons, the spatial weight $w_{i,j}$ expresses the neighbor relation between observations (polygons) i and j. Three broad classes of spatial weights $w_{i,j}$ are supported and include contiguity based weights, distance based weights, and weights formed as a hybrid of distance and contiguity criteria. Given that potentially there are $n^2 - n$ neighbor relations to consider, computational challenges quickly arise in large data sets.

A key challenge can be seen in the case of non-topological spatial data formats, such as ESRI's shapefile, where the contiguity relations between the areal units have to be derived to construct the spatial weights and then carry out any statistical analysis. A similar challenge arises for distance-based weights where KNN criterion are often employed to define neighbor relationships. In the context of big spatial data sets, efficient parallel approaches are required for the construction of the spatial weights.

While the construction and storage of spatial weights require high-performance techniques in large cross-sectional data settings, PySAL also has a *spatial dynamics* module which implements a suite of statistical measures for the analysis of spatio-temporal data. Here the concept of neighbor is expanded to space-time and the computational demands increase in their complexity.

Two broad sets of approaches are implemented, with the first treating spatial panel data where observations on fixed spatial units are taken over regular time steps. In this case, the development of the weights is similar to the approach used in the cross-sectional case as the topology of the units remains fixed over time. There are particular methods in the module that do require the consideration of all pairs of attribute value comparisons, as in the case of measures for local space-time concordance [72], and these add significantly to the computational burden. The second class of methods in the space-time analytics module contains methods for event data where now the spatial locations of the observations are no longer fixed over time but are now random. This necessitates a reconstruction of the spatial neighbor relations for each time step in the analysis. Additionally, analysts typically explore a range of time-windows in which case space-time neighbors are constructed for a large number of cases.

For both the cross-sectional and space-time cases, the spatial weights are used to form various statistics for local space, and space-time, association. The computational demands are not limited to the construction of the relevant spatial weights, but computationally based inference is often relied upon to assess the statistical significance of the resulting association measures. Because analytical results are unavailable for most of the local statistics, random spatial permutations are used to construct realizations of the process under the null hypothesis of complete spatial randomness. This is repeated for a large number of realizations for each observed data set. Depending on the precise form of the null and the spatial domain, this approach to inference can be extremely costly from a compute perspective and the search for efficient and parallel approaches is in its infancy [41].

PySAL also implements state-of-the-science spatial econometric methods in its *spreg* module. Spatial econometrics consists of a subset of econometric methods that

is concerned with spatial aspects present in cross-sectional and space-time observations [9]. As is the case for ESDA, spatial weights play a central role in spatial econometrics, however, the computational demands of spatial econometrics are not limited to weight construction and manipulation. As discussed more fully by [10], there are numerous computationally expensive estimators and required matrix operations in applied spatial econometrics. For example, nonlinear optimization problems are nested in various popular econometric estimation methods such as maximum likelihood and generalized methods of moments. Increasingly, spatial econometricians are employing Bayesian approaches that require the evaluation of high-dimensional integrals via Markov Chain Monte Carlo and Gibbs sampling. While much work in mainstream econometrics has focused on these estimators, their translation to the domain of spatial econometrics has necessitated careful optimization with an eye towards the particularities of spatially referenced data.

The issues examined thus far focus on computational demands of various spatial econometric estimation methods as well as ESDA, yet these methods are often combined with various geovisualization techniques in empirical work. Geovisualization can also pose daunting computational challenges in its own right. For example, choropleth mapping of the residuals from a certain spatial econometric model using the optimal classifier Fisher-Jenks has complexity $O(n^2k)$, where n is the number of observations and k the number of classes, and this is just to construct the classification [77]. The effective rendering of the resulting classification must also be taken into consideration. Parallel workflows that integrate both the implementation of various spatial econometric and ESDA measures together with their visualization remains an underdeveloped area of spatial computing [74].

A final area of spatial analysis with PySAL is the case of the regionalization module. A typical regionalization problem is to aggregate n polygons into p regions or zones, with $p \leq n$. Applications include the study of regional economic dynamics [26, 76] and geodemographics [73], among others. Formally, regionalization problems of this sort can be classified as NP-Hard. Optimal solutions for such problems are only feasible for small n cases, and even these can require long compute times. For example, [25] report a run time of 3 hours for $n = 50$.

Alternative heuristics have been suggested for these types of regionalization problems with prime examples being the max-p algorithm [24] and the p-regions model [25]. A key challenge for these algorithms is to provide an extensive exploration of the solution space as is possible in order to obtain good solutions. The original implementations of these algorithms were serial by design, and work exploiting parallelization to enhance these algorithms has only just begun [50].

Space limitations preclude a fuller examination of the PySAL library and the additional computational considerations. Because PySAL is designed to cover all layers of the spatial data analysis stack, from data integration, geoprocessing, statistical computation, and visualization, computational challenges can be found at each step. Consequently, the library offers an interesting case study for the use of parallel and high-performance advances to enhance the spatial analysis toolset.

VI. Map-Reduce-based Spatial Big Data Computation and Visualization

Map-Reduce [22] was proposed as a simple and flexible programming model to run on large shared-nothing clusters. A program is expressed as a *map* and a *reduce* functions, where the map function takes every input record and generates a set of intermediate key-value pairs, while the reduce function takes all values with the same key and processes them to produce the final answer. This model is used in several open source big data systems such as Apache Hadoop [40], Apache Hive [86], Microsoft Dryad [45], and Apache Pig [62]. Since Map-Reduce was one of the early modern big data systems, its open source counterpart Hadoop is the most widely used system for big spatial data so far. This includes support for range queries [7, 32, 59, 111, 118], *k*-nearest neighbor (KNN) queries [7, 8, 32, 111, 118], reverse nearest neighbor (RNN) queries, MaxRNN [8], spatial index bulk loading [7, 16, 32, 56, 111], spatial join [7, 32, 38, 119], all nearest neighbor (ANN) [99], KNN join [57, 116], and 3D rendering [95]. Table II outlines existing Map-Reduce-based work, where each row represents a system or a body of work related to big spatial data. Generally speaking, these techniques can be categorized into implementation approaches, *on-top*, *from-scratch*, and *built-in*, as follows.

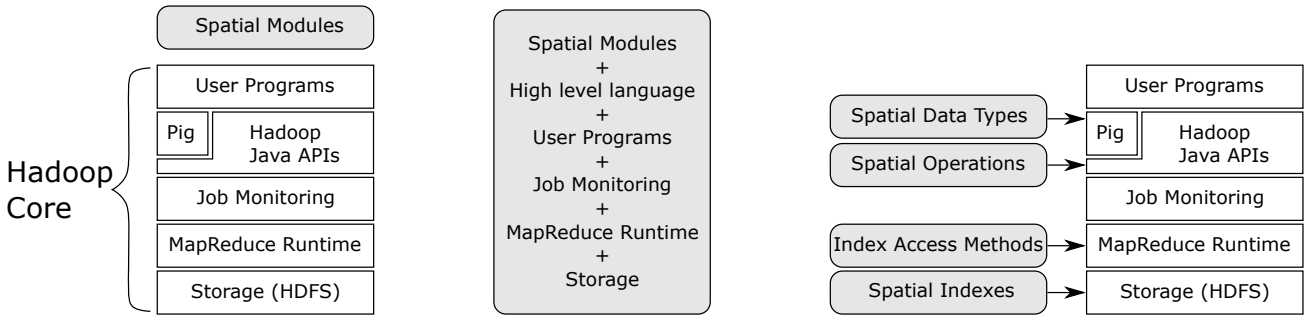
The on-top approach

In this approach, an existing Map-Reduce framework is used as a black box while the spatial logic is added through a standard API (9(a)). For example, in Hadoop, the spatial logic is injected through the standard *map* and *reduce* functions. This approach has two main advantages, *simplicity* and *portability*. First, it is very simple to implement as it uses the existing APIs exposed by the underlying system, e.g., map and reduce functions. Second, the techniques in this approach are highly *portable* to future system releases as long as they support the same interface. On the other hand, a huge disadvantage of the *on-top* approach is the poor performance as the core of the underlying system is still unaware of spatial data properties, hence, it misses many optimization opportunities. The algorithms in this approach can be further classified into *scan-based* and *partition-based* techniques. In *scan-based* techniques, the performance is limited by the speed of scanning the whole file even for the simplest operations. Examples of such techniques were applied to spatial range queries [118], k-nearest-neighbor

	Approach	Language	Indexes	Queries	Visualization
R-tree construction [16]	On-top	-	R-tree	Image quality	-
SJMR [54, 99, 118, 119]	On-top	-	R-tree	RQ, KNN, SJ, ANN	-
K-Means [121]	On-top	-	-	K-means	-
MR-DBSCAN [42]	On-top	-	-	DBSCAN	-
Voronoi Diagram [8]	On-top	-	-	VD, NN Queries	-
3D Visualization [95]	On-top	-	-	-	Single level
KNN Join [57, 116]	On-top	-	-	KNN Join	-
Multiway SJ [38]	On-top	-	-	Multiway SJ	-
BRACE [97]	From-scratch	BRASIL	Grid	SJ	-
PRADASE [59]	Built-in	-	Quad-tree	RQ	
Hadoop GIS [7]	Built-in	QL^{SP}	Grid	RQ, KNN, SJ	-
SpatialHadoop [28, 32, 33]	Built-in	Pigeon*	R tree, Quad tree, others	RQ, KNN, SJ, CG	Single-, multi-level
ScalaGiST [56]	Built-in	-	GiST	RQ, KNN	-
ESRI API for Hadoop [111]	Built-in	HiveQL*	PMR Quad Tree	RQ, KNN	-

$*$ OGC-compliant

Table II
MAP-REDUCE-BASED WORK IN THE AREA OF BIG SPATIAL DATA



(a) On-top (b) From-scratch (c) Built-in

Figure 9. Examples of implementations approaches in Hadoop

queries [7, 118], and K-means clustering [121]. In *partition-based* techniques, the data is first spatially partitioned so that nearby records are assigned to the same partition, and then each partition is processed independently to produce intermediate results which are finally combined to produce the final answer. Examples of such techniques were applied in the SMJR spatial join technique [119], which partitions the space according to a uniform grid and joins the contents of each grid cell. The SJMR algorithm is further expanded to multi-way spatial join [38]. Other examples include measuring satellite image quality [16], R-tree construction [16], spatial clustering [42], and various forms of nearest-neighbor queries [8, 57, 99, 116].

The from-scratch approach

The *from-scratch* approach is the other extreme, where a new system is built from scratch to support big spatial data processing (9(b)). This gives a better opportunity to inject as much spatial logic as possible into the core of the system. Such system would still follow one of the big data system architectures, such as Map-Reduce, but it optimizes the system core for spatial data processing. Although they perform much better than systems with *on-top* approach, *from-scratch* systems are very hard to maintain as they have to repeat all the components of big systems such as load-balancing and fault-tolerance. Furthermore, if users wish to mix spatial and non-spatial data processing, it would be impractical to use these systems as they might perform poorly when compared to other general-purpose systems. The prime example of this approach in the Map-Reduce world is BRACE [97], which performs behavioral simulation using in-memory Map-Reduce-based processing. BRACE rebuilds all the components from scratch where it uses an in-memory grid-based storage to store all the data. It runs a series of spatial join queries in hundreds of iterations to perform the simulation. BRACE allows the execution of multiple Map-Reduce jobs while the data is in memory without writing any intermediate or final data to disk. To ensure fault-tolerance, it occasionally writes a checkpoint to disk so that it can revert back to that point later. This new

	Global*	Local*	# partitions	Clustered$
PRADASE [59]	Q	-	# Blocks	C
Hadoop-GIS [7]	G	-	User-defined	C
S-Hadoop [32]	G, R, Q	R, Q	# Blocks	C
ScalaGiST [56]	K	GiST	User-defined	C&U
ESRI Index [111]	Q	Q	# Machines	C&U

* G: Grid, K: K-d tree, Q: Quad-tree, R: R-tree, Z: Z-curve

$ C: Clustered, U: Unclustered

Table III
SUMMARY OF SPATIAL INDEXES

design makes it more suitable to run behavioral simulation via iterative spatial joins.

The built-in approach

The *built-in* approach balances efficiency with simplicity as it injects spatial data awareness inside an existing general-purpose system (9(c)). This makes it efficient as the internal system becomes aware of spatial data and still it is not as complicated as building an entire system from scratch. Besides, it is more practical for users who wish to mix spatial and non-spatial workloads as it maintains the efficiency of the system with non-spatial data. In this approach, an existing general-purpose system is enhanced by injecting specific spatial components into the core without having to rebuild all components from scratch. The designers of the enhanced system choose which components to inject into the general-purpose system based on the needs of the applications that are ought to be supported. By carefully identifying the components that need to be added, this technique becomes much simpler than the from-scratch approach as there is no need to re-implement the basic features that are already supported by the general-purpose system such as fault-tolerance and load-balancing. In addition, since the general-purpose system functionality is present, the resulting system will then be more appealing to users who need to combine both spatial and non-spatial data processing. Most of the performance gain in the built-in approach is a direct result of using spatial indexes. Unfortunately, there is a major drawback to built-in systems which is their tight coupling with a specific version of the underlying system. Porting the internal modifications to a later release of the same system could be very difficult and time consuming.

Built-in Map-Reduce systems for big spatial data include PRADASE [59], Hadoop-GIS [5-7, 94, 96], SpatialHadoop [27, 29, 31, 32], ScalaGiST [56], and ESRI Tools for Hadoop [111]. In each of these systems, spatial indexes are injected into the storage layer in order to speed up some spatial operations. In general, spatial indexes in these systems follow a two-level layout of one *global index* and multiple *local indexes*. The global index determines how records are partitioned across machines, while local indexes organize records inside each machine. This high-level design is flexible to model a wide range of spatial indexes based on grid, R-tree, Quad-tree, K-d tree, Z-Curve, and others. Table III outlines the indexes deployed in each of these systems in the global and local levels. The table also shows the *number of partitions* used in the global index. This number could be equal to the number of machines to have one partition per machine or it could be adjusted such that each partition has a fixed predefined size. Finally, the table shows whether the index is *clustered* or *unclustered*. A clustered index means that the actual records are reordered to match the order of the index to minimize random access to the data. An unclustered index is built without having to reorder the actual data which could be necessary if multiple indexes are to be built for the same data set.

One of the early attempts *built-in* systems is PRADASE [59], which extends Hadoop with a clustered three-dimensional quad-tree-based index to speedup range queries on trajectory data. Hadoop-GIS [5-7, 94, 96] comes with its own language QL^{SP} geared towards processing medical data, is supported by a grid-based index, and supports range queries, k-nearest neighbor queries, and *self* spatial joins. SpatialHadoop [27, 29, 31, 32] has its own language, termed Pigeon [30], equipped with main spatial operations and functions. It supports grid, quad-tree, and R-tree spatial index structures, which support range queries, k-nearest-neighbor queries, binary spatial joins, and a suite of computational geometry operations [28]. Among all *built-in* systems, SpatialHadoop is the only one that provides built-in visualization functionality for big spatial data [33, 34], where it supports single- and multi-level images. A multi-level image is where a new image will be loaded for each zoom-in and out operation. ScalaGiST [56] uses KD-tree as a global index with a clustered and unclustered local index structures based on Generalized search Index Structure (GiST) [44], geared towards supporting range and k-nearest-neighbor queries. Finally, ESRI Tools for Hadoop [111] employs a HiveQL language with quad-tree-based index structures to support range and k-nearest-neighbor queries. Integration of Map-Reduce-based systems with HPC hardware including graphics accelerators remains largely not addressed.

VII. FRONTIERS IN CYBERGIS AND GEOSPATIAL DATA SCIENCE

CyberGIS

CyberGIS (also known as cyber geographic information science and systems (GIS) based on advanced computing and cyberinfrastructure) has emerged as new-generation GIS, comprising a seamless integration of advanced cyberinfrastructure, GIS, and spatial analysis and modeling capabilities while leading to widespread research advances and broad societal impacts [10, 102]. During the past several years, cyberGIS has grown as a vibrant interdisciplinary field as evidenced through impactful publications and number of hardware and software capabilities, collaborative projects,

meetings, conferences, and workshops. For example, NSF-funded multi-institution project led by Shaowen Wang's group (www.cybergis.org): CyberGIS Software Integration for Sustained Geospatial Innovation involves a number of academic institutions, industrial partners, U.S. government agency partners, U.S. federally funded research and development laboratories, and multiple international partners. With an international scope, the project has established a novel cyberGIS software framework while achieving major scientific and technological advances in tackling challenging environmental and geospatial problems [104]. Cyber-

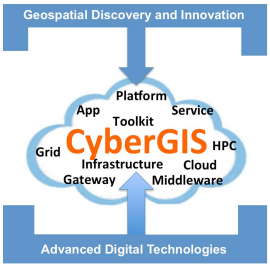


Figure 10. CyberGIS Ecosystem [104]

GIS software established by the NSF CyberGIS Project includes three key interrelated modalities: 1) CyberGIS Gateway (http://gateway.cybergis.org/) for providing easy access to online cyberGIS analytics and services; 2) CyberGIS Toolkit for open source communities to develop and share scalable geospatial software modules, and 3) GISolve middleware for managing the complexity of cyberinfrastructure and cyberGIS capabilities to simplify application development [105, 108, 109]. CyberGIS software and diverse applications have also driven computing system innovation. In particular, ROGER was developed (https://wiki.ncsa.illinois.edu/display/ROGER/) as the first hybrid supercomputer with three computing modalities tightly integrated including high-performance computing with advanced Graphics Processing Units (GPUs), data-intensive computing with Hadoop and Spark, and cloud computing with OpenStack. The system is integrated through fast network and shared management of high-performance data storage while the cyberGIS software is deployed to glue the three computing modalities together for enabling a variety of research advances across bio, geo, engineering, and social sciences [103].

Scientific approaches based on cyberGIS have already had significant impacts in a number of domains (e.g., hydrology and water resources, complex coupled human-natural systems, emergency management, econometrics, geophysical sciences, and public health) through processing of complex and massive amounts of geospatial data and performing associated analysis, simulation, and visualization [10, 55, 107, 110]. In the foreseeable future, cyberGIS is expected to play critical roles in many fields as solving complex scientific problems and improving decision-making practices increasingly depend on cyberGIS capabilities to handle very large and complex spatial and spatiotemporal data, and effectively manage sophisticated geospatial analysis and visualization.

Geospatial Data Science

The complexity, diversity and rapid growth of geospatial data have increased significantly over recent decades and are driving discoveries and innovations in a large number of application and science domains [112]. Access to and interaction with geospatial big data collected from numerous sources are increasingly fundamental to explore natural, human and social systems at unprecedented scales and provide tremendous opportunities to gain dynamic insights into complex phenomena through big compute (e.g., cloud and high-performance computing) and cyberGIS approaches [103]. Though geospatial big data have played important roles in many domains and promise to enable a wide range of decision-making practices with significant societal impacts, geospatial data science remains to be established for advancing leading-edge research and education in the era of big data.

An exciting frontier in geospatial data science is to infuse geospatial domain knowledge into the holistic life-cycle of geospatial big data ranging from acquisition, access, management, and processing to analysis and visualization. Oftentimes, scientists or users do not have a complete picture of how geospatial data are derived from various sources. Therefore, to assure scientific rigor of transforming diverse geospatial data poses significant challenges. In response to such challenges, computational reproducibility is expected to emerge as an important research direction for geospatial data science particularly related to data and model coupling [85]. A related frontier is centered on machine learning and prediction based on integration of systems such as cyberGIS and domain-specific models [103]. Future advances of geospatial data science will depend on desirable GIS innovation, and in turn, enable a broad array of research communities through integration with GIS analytics and workflow for solving significant scientific and decision-making problems.

VIII. Time-Sensitive High Performance Computation over Polygonal Datasets

Time-critical Applications and HPC

This section focuses on polygonal/vector datasets (as opposed to raster) which are widely employed for their accuracy but are hard to process due to their irregular structure. We examine fundamental Geo-spatial algorithms, data structures and system issues and why HPC infrastructures, including harnessing graphics accelerators are needed for time-critical applications. For many applications, these primitive-like operations are also time sensitive, either due

to emergency or real-time decision-making requirements and even Map-Reduce-like parallel systems are inadequate. These operations are often used by city planners, businesses and federal and other government agencies (e.g., FEMA, CDC) to combine and analyze multiple geospatial layers like physical mapping data, street layout, water bodies, population distribution, etc., to get actionable information. For emergency response in a major natural disaster, such as a tornado, hurricane or large earthquake, every second lost in the rescue and recovery effort could mean more property damage, additional injuries or even loss of life. Since these operations require significant processing power, a heterogeneous scientific computing platform consisting of multi-core CPUs and manycore graphics processing units (GPUs) is necessary to accelerate such computations. Shared and distributed memory platforms comprising of GPUs with 1000s of processing cores in a single chip have the potential to speedup the computations by one-to-two orders of magnitude [66]. With the availability of this massively parallel hardware, algorithmic research on efficient parallelization of core data structures such as R-tree and geospatial primitives such as intersection, buffer, difference, merge, union, etc., is required. This section describes data structure, algorithm and systems work and future challenges led by Prasad's group. Table IV summarizes them along with some other systems in the literature.

MPI-GIS System

For many GIS and Spatial Database applications, spatial overlay or join on two or more layers of geospatial data is necessary. However, using sequential paradigm to process them is time-consuming. For instance, it takes roughly 20 hours to compute the spatial join of a polyline table with 73M records representing the contiguous USA with itself on an Amazon EC2 instance [71]. These operations involve irregular I/O and computation due to varying number of vertices in different shapes with no well-defined communication pattern due to irregular spatial and/or temporal task or data distributions. These irregularities make parallelization, partitioning, and load balancing challenging. Prasad's group has undertaken parallelization of R-tree, spatial join and overlay algorithms using GPU and MPI [3, 66, 68]. Their MPI-GIS system, depicted in Figure 12, achieves 44X speedup while processing about 600K polygons (4GB) in two real-world GIS shapefiles (USA Detailed Water Bodies and USA Block Group Boundaries) within 20 seconds on a 32-node cluster with 8 cores each [70]. Execution time plot for MPI-GIS system is shown in Figure 11.

Parallel R-Tree on GPU

R-Tree is an industry standard spatial data structure used in many domains such as GIS, spatial database management systems and VLSI. Implementing R-Tree on the GPU is difficult due to both non-linear tree topology and the stream

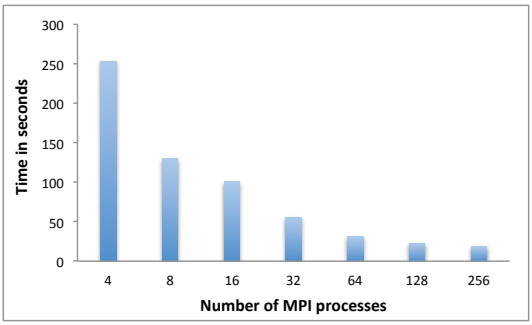


Figure 11. Execution time of *MPI-GIS* with varying number of MPI processes for Polygon Overlay operation on two layers - USA Water Bodies and USA Block Group Boundaries on *NSERC Carver* cluster with 32 compute nodes having 8 cores/node.

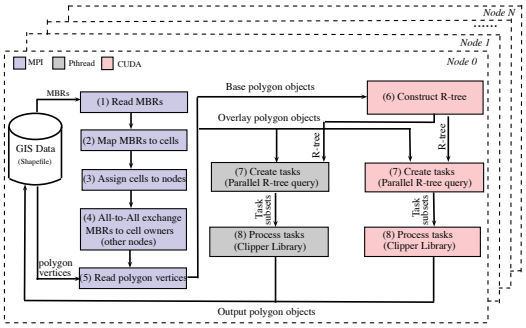


Figure 12. MPI-GIS Architecture

architecture of the GPU itself [58]. Prasad's group has developed a fast GPGPU implementation of an R-Tree [66, 67]. Their GPU implementation yields over 200-fold speedup for R-tree construction and up to 180-fold speedup for query on real world datasets.

Previously, parallel and distributed construction for R-Tree was attempted on multi-core and distributed systems, however, these did not include GPU implementations [19, 61, 78]. [58] reports a 20x increase in construction and a 30x increase in query speed over sequential. A bulk loading construction similar to Prasad's group was outlined by [114] but does not perform "Geo-Packing" and thus does not retain implicit spatial relations between objects nor does it pre-sort the objects to minimize branching. Also outlined in [114] are several algorithms for querying, however, they report only a 10x speedup over then current multicore implementations due to overhead costs associated with their data primitives.

Spatial Join on GPU

Prasad's group has recently developed a GCMF system which can perform spatial join over non-indexed polygonal datasets of more than 3 GB file size comprising over 600,000 polygons on a single GPU within 8 seconds, representing 39-fold end-to-end speedup versus optimized sequential rou-

System	Operations	Speedup	Baseline	Data size(objects)
GCMF [4]	MBR query, ST_intersect	20-39	Optimized GEOS	Up to 0.6M
MPI-GIS [68]	Polygon overlay	44	ArcGIS	Up to 0.6M
CudaGIS [117]	point-in-polygon	19	Sequential CPU	1.7M points, 40K polygons
GPU-Rtree [67]	R-tree construction/querying	91-226	Sequential CPU	Up to 50K
GPU-Rtree [58]	R-tree construction/querying	20-30	Sequential CPU	Up to 2.2M

Table IV
SOME RECENT HPC SYSTEMS FOR SPATIAL PROCESSING SYSTEMS

tines of GEOS C++ library as well as PostgreSQL spatial database with PostGIS [4]. This system performs a two-step filtering phase: 1) A sort-based Minimum Bounding Rectangle (MBR) filtering step (SMF) detects potentially overlapping polygon pairs up to 20 times faster than the optimized GEOS library routine. 2) A linear time Common MBR filtering (CMF) step (based on the overlapping area of two given MBRs) is employed which not only eliminates two-third of the candidate polygon pairs but also reduces the number of edges to be considered in the refinement phase by almost 40-fold on an average.

Algorithms for Polygonal Datasets

In addition to parallel data structures, advances in parallel algorithms are also needed for fundamental operations such as polygonal overlay. For two polygons with n vertices, the number of intersections can be $O(n^2)$. Sequential algorithms for this problem are in abundance in literature but there are very few parallel algorithms solving the polygonal overlay problem in its most general form. Using segment tree data structure, Puri and Prasad have developed the first output-sensitive parallel algorithm, which can perform polygon intersection and union in $O(log\ n)$ time using $O(n + k)$ processors by parallelizing Greiner Hormann polygon clipping algorithm, where n is the number of vertices and k is the number of intersections [69, 70]. This improves upon another $O(\log n)$ time algorithm by Karinthi, Srinivas, and Almasi which unlike Prasad and Puri's algorithm is not output-sensitive, and must employ $\theta(n^2)$ processors to achieve $O(\log n)$ time [47].

Future Roadmap

Going forward, we envision a software stack for an exascale system that is one to two orders faster than the present systems and meets the needs of a broad spectrum of spatial applications and workloads. This is in line with President Obama's National Strategic Computing initiative whose one key goal is to develop software tool sets effectively exploiting the system and programming software stacks of both the traditional HPC (MPI, CUDA) and data analytics (Hadoop/Map-Reduce) environments [1]. The current state-of-art system and software for spatial joins and overlay in the HPC arena can handle data sets of size in GBs on a small to medium size cluster. In order to effectively leverage heterogeneous platforms, further research into new parallel algorithms, data structures, and system scalability is necessary. We envision an HPC software stack comprising of 1) parallel I/O libraries on top of Lustre/GPFS filesystems, 2) suite of concurrent spatial data structures like segment tree, TPR-tree, etc., 3) scalable spatial algorithms on multi-core and manycore architectures, and 4) effective load balancing algorithms to deal with ill-structured polygonal data. The overall goal would be to develop optimized HPC software tools and libraries for spatial geo, bio and other computations akin to Linear Algebra Package (LAPACK) for matrix operations and Nanoscale Molecular Dynamics (NAMD) tools for molecular dynamics simulation, etc., that can leverage supercomputers.

For system scalability, the challenges are CPU-GPU integration, lack of parallel I/O libraries for vector data and optimized spatial primitives for accelerators. For data intensive applications, MPI-IO integration is required in MPI-GIS to handle terabyte-size vector datasets. There are existing PnetCDF and HDF5 libraries for parallel I/O to access files of array-oriented NetCDF and HDF format. These libraries support data with raster format or hierarchical data format. These cannot handle irregular vector data like shapefiles or CSV files currently supported by MPI-GIS. Moreover, for compute-intensive applications like joins and overlays, GPU spatial primitives libraries need to be developed and integrated with MPI-GIS and other systems like SpatialHadoop and HadoopGIS.

ACKNOWLEDGMENT

Prasad's group research is partially supported by NSF grant 1205650. Rey's group research is partially supported by NSF grant 1421935. Fusheng Wang's group research is partially supported by NSF grants ACI 1443054 and IIS 1350885. Shekhar's group research is partially supported by NSF grant IIS-1320580 and USDOD grant HM0210-13-1-0005.

REFERENCES

[1] Executive order, creating a national strategic computing initiative, july 29, 2015. https://obamawhitehouse.archives.gov/the-press-office/2015/07/29/executive-order-creating-national-strategic-computing-initiative. Accessed: 2017-05-03.

[2] SaTScan. https://www.satscan.org/. Accessed: 2017-05-13.

[3] Dinesh Agarwal, Satish Puri, Xi He, and Sushil K Prasad. A system for gis polygonal overlay computation on linux cluster-an experience and performance report. In *Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW), 2012 IEEE 26th International*, pages 1433–1439. IEEE, 2012.

[4] Danial Aghajarian, Satish Puri, and Sushil Prasad. GCMF: an efficient end-to-end spatial join system over large polygonal datasets on GPGPU platform. In *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, page 18. ACM, 2016.

[5] Ablimit Aji, Xiling Sun, Hoang Vo, Qiaoling Liu, Rubao Lee, Xiaodong Zhang, Joel H. Saltz, and Fusheng Wang. Demonstration of Hadoop-GIS: a spatial data warehousing system over MapReduce. In *Proceedings of the ACM Symposium on Advances in Geographic Information Systems, ACM SIGSPATIAL*, pages 518–521, November 2013.

[6] Ablimit Aji, George Teodoro, and Fusheng Wang. Haggis: Turbocharge a MapReduce Based Spatial Data Warehousing System with GPU Engine. In *Proceedings of the ACM SIGSPATIAL International Workshop on Analytics for Big Geospatial Data*, pages 15–20, November 2014.

[7] Ablimit Aji, Fusheng Wang, Hoang Vo, Rubao Lee, Qiaoling Liu, Xiaodong Zhang, and Joel H. Saltz. Hadoop-GIS: A High Performance Spatial Data Warehousing System over MapReduce. *Proceedings of the VLDB Endowment, PVLDB*, 6(11):1009–1020, 2013.

[8] Afsin Akdogan, Ugur Demiryurek, Farmoush Banaei-Kashani, and Cyrus Shahabi. Voronoi-based Geospatial Query Processing with MapReduce. In *International Conference on Cloud Computing Technology and Science*, pages 9–16, 2010.

[9] L Anselin. Spatial econometrics. In A T C Mills and K Patterson, editors, *Palgrave Handbook of Econometrics: Volume 1, Econometric Theory*. Palgrave McMillan, New York, 2006.

[10] Luc Anselin and Sergio J Rey. Spatial econometrics in an age of cybergiscience. *International Journal of Geographical Information Science*, 26(12):2211–2226, 2012.

[11] Marc P Armstrong, Claire E Pavlik, and Richard Marciano. Parallel processing of spatial statistics. *Computers & Geosciences*, 20(2):91–104, 1994.

[12] Furqan Baig, Mudit Mehrotra, Hoang Vo, Fusheng Wang, Joel Saltz, and Tahsin Kurc. Sparkgis: Efficient comparison and evaluation of algorithm results in tissue image analysis studies. In Fusheng Wang, Gang Luo, Chunhua Weng, Arijit Khan, Prasenjit Mitra, and Cong Yu, editors, *VLDB Workshop on Data Management and Analytics for Healthcare and Medicine*, pages 134–146, 2015.

[13] Sajib Barua and Reda Alhajj. Parallel wavelet transform for spatio-temporal outlier detection in large meteorological data. *Intelligent Data Engineering and Automated Learning-IDEAL 2007*, pages 684–694, 2007.

[14] J. Bolton and P. Gader. Application of multiple-instance learning for hyperspectral image analysis. *Geoscience and Remote Sensing Letters, IEEE*, 8(5):889 –893, sept. 2011.

[15] Jasmine Burguet and Philippe Andrey. Statistical Comparison of Spatial Point Patterns in Biological Imaging. *PLOS ONE*, 9(2):1–12, 02 2014.

[16] Ariel Cary, Zhengguo Sun, Vagelis Hristidis, and Naphtali Rishe. Experiences on Processing Spatial Data with MapReduce. In *Proceedings of the International Conference on Scientific and Statistical Database Management, SSDBM*, pages 302–319, 2009.

[17] Varun Chandola and R. Raju Vatsavai. A scalable gaussian process analysis algorithm for biomass monitoring. *Statistical Analysis and Data Mining*, 4(4):430–445, 2011.

[18] Varun Chandola and Ranga Raju Vatsavai. Scalable time series change detection for biomass monitoring using gaussian process. In *CIDU*, pages 69–82, 2010.

[19] J. K. Chen, Yin-Fu Huang, and Yeh-Hao Chin. A Study of Concurrent Operations on R-Trees. *Inf. Sci.*, 98(1-4):263–300, 1997.

[20] Noel Cressie. *Statistics for spatial data*. John Wiley & Sons, 2015.

[21] Noel Cressie and Christopher K Wikle. *Statistics for spatio-temporal data*. John Wiley & Sons, 2011.

[22] Jeffrey Dean and Sanjay Ghemawat. MapReduce: Simplified Data Processing on Large Clusters. *Communications of ACM*, 51:107–113, 2008.

[23] Thomas G. Dietterich, Richard H. Lathrop, Tomas Lozano-Perez, and Arris Pharmaceutical. Solving the multiple-instance problem with axis-parallel rectangles. *Artificial Intelligence*, 89:31–71, 1997.

[24] Juan C Duque, Luc Anselin, and Sergio J Rey. The max-p-regions problem. *Journal of Regional Science*, 52(3):397–419, 2012.

[25] Juan C Duque, Richard L Church, and Richard S Middleton. The p-regions problem. *Geographical Analysis*, 43(1):104–126, 2011.

[26] Juan C Duque, Xinyue Ye, and David C Folch. spmorph: An exploratory space-time analysis tool for describing processes of spatial redistribution. *Papers in Regional Science*, 94(3):629–651, 2015.

[27] Ahmed Eldawy. SpatialHadoop: Towards Flexible and Scalable Spatial Processing using MapReduce. In *The PhD Symposium in the International Conference on Management of Data, SIGMOD*, pages 46–50, June 2014.

[28] Ahmed Eldawy, Yuan Li, Mohamed F. Mokbel, and Ravi Janardan. CG_Hadoop: Computational Geometry in MapReduce. In *SIGSPATIAL*, pages 284–293, 2013.

[29] Ahmed Eldawy and Mohamed F. Mokbel. A demonstration of spatialhadoop: An efficient mapreduce framework for spatial data. *Proceedings of the VLDB Endowment, PVLDB*, 6(12):1230–1233, 2013.

[30] Ahmed Eldawy and Mohamed F. Mokbel. Pigeon: A Spatial MapReduce Language. In *Proceedings of the International Conference on Data Engineering, ICDE*, pages 1242–1245, 2014.

[31] Ahmed Eldawy and Mohamed F. Mokbel. The Ecosystem of SpatialHadoop. *SIGSPATIAL Special*, 6(3):3–10, 2014.

[32] Ahmed Eldawy and Mohamed F. Mokbel. SpatialHadoop: A MapReduce Framework for Spatial Data. In *Proceedings of the International Conference on Data Engineering, ICDE*, pages 1352–1363, 2015.

[33] Ahmed Eldawy, Mohamed F. Mokbel, and Christopher Jonathan. A Demonstration of HadoopViz: An Extensible MapReduce System for Visualizing Big Spatial Data. *Proceedings of the VLDB Endowment, PVLDB*, 8(12):1896–1907, 2015.

[34] Ahmed Eldawy, Mohamed F. Mokbel, and Christopher Jonathan. HadoopViz: A MapReduce Framework for Extensible Visualization of Big Spatial Data. In *Proceedings of the International Conference on Data Engineering, ICDE*, pages 601–612, May 2015.

[35] Vijay Gandhi, Mete Celik, and Shashi Shekhar. Parallelizing multiscale and multigranular spatial data mining algorithms. In *Partitioned Global Address Space Programming Models Conference*, 2006.

[36] David Ginsbourger, Rodolphe Le Riche, and Laurent Carraro. Kriging is well-suited to parallelize optimization. In *Computational Intelligence in Expensive Optimization Problems*, pages 131–162. Springer, 2010.

[37] J. Graesser, A. Cheriyadat, R.R. Vatsavai, V. Chandola, J. Long, and E. Bright. Image based characterization of formal and informal neighborhoods in an urban landscape. *Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal of*, 5(4):1164 –1176, aug. 2012.

[38] Himanshu Gupta, Bhupesh Chawda, Sumit Negi, Tanveer A. Faruquie, L. V. Subramaniam, and Mukesh Mohania. Processing multi-way spatial joins on map-reduce. In *Proceedings of the 16th International Conference on Extending Database Technology*, EDBT, pages 113–124, New York, NY, USA, 2013.

[39] Metin N. Gurcan, Laura E. Boucheron, Anant Madabhushi Ali Can, Nasir M. Rajpoot, and Bulent Yener. Histopathological Image Analysis: A Review. *IEEE Rev Biomed Eng.*, 2:147171, 2009.

[40] Apache Hadoop, 2015. http://hadoop.apache.org/.

[41] Frank Hardisty and Alexander Klippel. Analysing spatio-temporal autocorrelation with lista-viz. *International Journal of Geographical Information Science*, 24(10):1515–1526, 2010.

[42] Yaobin He, Haoyu Tan, Wuman Luo, Shengzhong Feng, and Jianping Fan. MR-DBSCAN: A Scalable MapReduce-based DBSCAN Algorithm for Heavily Skewed Data. *Frontiers of Computer Science*, 8(1):83–99, 2014.

[43] Richard Healey, Steve Dowers, Bruce Gittings, and Mike J Mineter. *Parallel processing algorithms for GIS*. CRC Press, 1997.

[44] Joseph M. Hellerstein, Jeffrey F. Naughton, and Avi Pfeffer. Generalized search trees for database systems. In *VLDB*, pages 562–573, 1995.

[45] Michael Isard, Mihai Budiu, Yuan Yu, Andrew Birrell, and Dennis Fetterly. Dryad: Distributed Data-Parallel Programs from Sequential Building Blocks. In *EuroSys*, pages 59–72, 2007.

[46] Zhe Jiang, Shashi Shekhar, Xun Zhou, Joseph Knight, and Jennifer Corcoran. Focal-test-based spatial decision tree learning. *IEEE Transactions on Knowledge and Data Engineering*, 27(6):1547–1559, 2015.

[47] R. Karinthi, K. Srinivas, and G. Almasi. A parallel algorithm for computing polygon set operations. In *Proceedings of 8th International Parallel Processing Symposium*, pages 115–119, Apr 1994.

[48] Baris M Kazar, Shashi Shekhar, David J Lilja, Daniel Boley, Dale Shires, James Rogers, and Mete Celik. A parallel formulation of the spatial auto-regression model. In *II International Conference and Exhibition on Geographic Information*, 2005.

[49] Jun Kong, Lee AD Cooper, Fusheng Wang, David A Gutman, Jingjing Gao, Candace Chisolm, Ashish Sharma, Tony Pan, Erwin G Van Meir, Tahsin M Kurc, et al. Integrative, multimodal analysis of glioblastoma using tcga molecular data, pathology images, and clinical outcomes. *Biomedical Engineering, IEEE Transactions on*, 58(12):3469–3474, 2011.

[50] Jason Laura, Wenwen Li, Sergio J Rey, and Luc Anselin. Parallelization of a regionalization heuristic in distributed computing platforms–a case study of parallel-p-compact-regions problem. *International Journal of Geographical Information Science*, 29(4):536–555, 2015.

[51] Yanhui Liang, Hoang Vo, Ablimit Aji, Jun Kong, and Fusheng Wang. Efficient In-Memory Based Spatial Queries for Large-Scale 3D Data with Complex Structures. Submitted to SIGSPATIAL 2017.

[52] Yanhui Liang, Hoang Vo, Ablimit Aji, Jun Kong, and Fusheng Wang. Scalable 3D Spatial Queries for Analytical Pathology Imaging with MapReduce. In *SIGSPATIAL*, pages 52:1–52:4, 2016.

[53] Yanhui Liang, Fusheng Wang, Darren Treanor, Derek Magee, George Teodoro, Yangyang Zhu, and Jun Kong. A 3d primary vessel reconstruction framework with serial microscopy images. In *Medical Image Computing and Computer-Assisted Intervention*, 2015.

[54] Haojun Liao, Jizhong Han, and Jinyun Fang. Multi-dimensional Index on Hadoop Distributed File System. *International Conference on Networking, Architecture, and Storage*, 0:240–249, 2010.

[55] Tao Lin, Shaowen Wang, Luis F Rodríguez, Hao Hu, and Yan Liu. CyberGIS-enabled decision support platform for biomass supply chain optimization. *Environmental Modelling & Software*, 70:138–148, 2015.

[56] Peng Lu, Gang Chen, Beng Chin Ooi, Hoang Tam Vo, and Sai Wu. ScalaGiST: Scalable Generalized Search Trees for MapReduce Systems. *PVLDB*, 7(14):1797–1808, 2014.

[57] Wei Lu, Yanyan Shen, Su Chen, and Beng Chin Ooi. Efficient Processing of k Nearest Neighbor Joins using MapReduce. *Proceedings of the VLDB Endowment, PVLDB*, pages 1016–1027, 2012.

[58] Lijuan Luo, Martin DF Wong, and Lance Leong. Parallel implementation of R-trees on the GPU. In *Design Automation Conference (ASP-DAC), 2012 17th Asia and South Pacific*, pages 353–358. IEEE, 2012.

[59] Qiang Ma, Bin Yang, Weining Qian, and Aoying Zhou. Query Processing of Massive Trajectory Data Based on MapReduce. In *International Workshop on Cloud Data Management, CloudDB*, pages 9–16, 2009.

[60] Oded Maron and Toms Lozano-Prez. A framework for multiple-instance learning. In *ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS*, pages 570–576. MIT Press, 1998.

[61] V. Ng and T. Kameda. The r-link tree: A recoverable index structure for spatial data. In *Database and Expert Systems Applications*, pages 163–172. Springer, 1994.

[62] Christopher Olston, Benjamin Reed, Utkarsh Srivastava, Ravi Kumar, and Andrew Tomkins. Pig Latin: A Not-so-foreign Language for Data Processing. In *Proceedings of the ACM International Conference on Management of Data, SIGMOD*, pages 1099–1110, 2008.

[63] Linsey Xiaolin Pang, Sanjay Chawla, Bernhard Scholz, and Georgina Wilcox. A scalable approach for lrt computation in gpgpu environments. In *Asia-Pacific Web Conference*, pages 595–608. Springer, 2013.

[64] Lluís Pesquer, Ana Cortés, and Xavier Pons. Parallel ordinary kriging interpolation incorporating automatic variogram fitting. *Computers & Geosciences*, 37(4):464–473, 2011.

[65] Karthik Ganesan Pillai and Ranga Raju Vatsavai. Multi-sensor remote sensing image change detection: An evaluation of similarity measures. In *13th IEEE International Conference on Data Mining Workshops, ICDM Workshops, TX, USA, December 7-10, 2013*, pages 1053–1060, 2013.

[66] Sushil K Prasad, Michael McDermott, Satish Puri, Dhara Shah, Danial Aghajarian, Shashi Shekhar, and Xun Zhou. A vision for GPU-accelerated parallel computation on geospatial datasets. *SIGSPATIAL Special*, 6(3):19–26, 2015.

[67] Sushil K Prasad, Shashi Shekhar, Michael McDermott, Xun Zhou, Michael Evans, and Satish Puri. GPGPU-accelerated interesting interval discovery and other computations on geospatial datasets: A summary of results. In *Proceedings of the 2nd ACM SIGSPATIAL International Workshop on Analytics for Big Geospatial Data*, pages 65–72. ACM, 2013.

[68] Satish Puri, Dinesh Agarwal, Xi He, and Sushil K Prasad. Mapreduce algorithms for gis polygonal overlay processing. In *Parallel and Distributed Processing Symposium Workshops & PhD Forum (IPDPSW), 2013 IEEE 27th International*, pages 1009–1016. IEEE, 2013.

[69] Satish Puri and Sushil K Prasad. Output-sensitive parallel algorithm for polygon clipping. In *Parallel Processing (ICPP), 2014 43rd International Conference on*, pages 241–250. IEEE, 2014.

[70] Satish Puri and Sushil K Prasad. A parallel algorithm for clipping polygons with improved bounds and a distributed overlay processing system using MPI. In *Cluster, Cloud and Grid Computing (CCGrid), 2015 15th IEEE/ACM International Symposium on*, pages 576–585. IEEE, 2015.

[71] Suprio Ray, Bogdan Simion, Angela Demke Brown, and Ryan Johnson. A parallel spatial data analysis infrastructure for the cloud. In *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 284–293. ACM, 2013.

[72] Sergio J Rey. Space–time patterns of rank concordance: Local indicators of mobility association with application to spatial income inequality dynamics. *Annals of the American Association of Geographers*, 106(4):788–803, 2016.

[73] Sergio J Rey, Luc Anselin, David C Folch, Daniel Arribas-Bel, Myrna L Sastré Gutiérrez, and Lindsey Interlante. Measuring spatial dynamics in metropolitan areas. *Economic Development Quarterly*, 25(1):54–64, 2011.

[74] Sergio J Rey, Luc Anselin, Xun Li, Robert Pahle, Jason Laura, Wenwen Li, and Julia Koschinsky. Open geospatial analytics with pysal. *ISPRS International Journal of Geo-Information*, 4(2):815–836, 2015.

[75] Sergio J Rey, Luc Anselin, Robert Pahle, Xing Kang, and Philip Stephens. Parallel optimal choropleth map classification in pysal. *International Journal of Geographical Information Science*, 27(5):1023–1039, 2013.

[76] Sergio J Rey and Myrna L Sastré-Gutiérrez. Interregional inequality dynamics in mexico. *Spatial Economic Analysis*, 5(3):277–298, 2010.

[77] Sergio J Rey, Philip Stephens, and Jason Laura. An evaluation of sampling and full enumeration strategies for fisher jenks classification in big data settings. *Transactions in GIS*, 2016.

[78] Bernd Schnitzer and Scott T. Leutenegger. Master-Client R-Trees: A New Parallel R-Tree Architecture. In *SSDBM*, pages 68–77, 1999.

[79] Shashi Shekhar, Michael R Evans, James M Kang, and Pradeep Mohan. Identifying patterns in spatial information: A survey of methods. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(3):193–214, 2011.

[80] Shashi Shekhar, Zhe Jiang, Reem Y Ali, Emre Eftelioglu, Xun Tang, Venkata Gunturi, and Xun Zhou. Spatiotemporal data mining: A computational perspective. *ISPRS International Journal of Geo-Information*, 4(4):2306–2338, 2015.

[81] Shashi Shekhar, Chang-Tien Lu, and Pusheng Zhang. Detecting graph-based spatial outliers: algorithms and applications (a summary of results). In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 371–376. ACM, 2001.

[82] Shashi Shekhar, Sivakumar Ravada, D Chubb, and G Turner. Declustering and load-balancing methods for parallelizing geographic information systems. *IEEE Transactions on Knowledge and Data Engineering*, 10(4):632–655, 1998.

[83] Shashi Shekhar, Sivakumar Ravada, Vipin Kumar, Douglas Chubb, and Greg Turner. Parallelizing a GIS on a shared address space architecture. *Computer*, 29(12):42–48, 1996.

[84] Oleg A Smirnov and Luc E Anselin. An o (n) parallel method of computing the log-jacobian of the variable transformation for models with spatial interaction on a lattice. *Computational Statistics & Data Analysis*, 53(8):2980–2988, 2009.

[85] Victoria Stodden, Marcia McNutt, David H Bailey, Ewa Deelman, Yolanda Gil, Brooks Hanson, Michael A Heroux, John PA Ioannidis, and Michela Taufer. Enhancing reproducibility for computational methods. *Science*, 354(6317):1240–1241, 2016.

[86] Ashish Thusoo, Joydeep Sarma Sen, Namit Jain, Zheng Shao, Prasad Chakka, Suresh Anthony, Hao Liu, Pete Wyckoff, and Raghotham Murthy. Hive: A Warehousing Solution over a Map-Reduce Framework. *Proceedings of the VLDB Endowment, PVLDB*, pages 1626–1629, 2009.

[87] P. Torrione, C. Ratto, and L.M. Collins. Multiple instance and context dependent learning in hyperspectral data. In *Hyperspectral Image and Signal Processing: Evolution in Remote Sensing, 2009. WHISPERS '09. First Workshop on*, pages 1 –4, aug. 2009.

[88] Ranga Raju Vatsavai. A data mining framework for monitoring nuclear facilities. In *ICDM Workshops (Industry/Government Track)*, page 917, 2012.

[89] Ranga Raju Vatsavai. Rapid damage explorer (RDX): A probabilistic framework for learning changes from bitemporal images. In *12th IEEE International Conference on Data Mining (Demo Paper), Brussels, Belgium, December 10, 2012*, pages 906–909, 2012.

[90] Ranga Raju Vatsavai. Gaussian multiple instance learning approach for mapping the slums of the world using very high resolution imagery. In *The 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2013, Chicago, IL, USA, August 11-14, 2013*, pages 1419–1426, 2013.

[91] Ranga Raju Vatsavai. Object based image classification: state of the art and computational challenges. In *Proceedings of the 2nd ACM SIGSPATIAL International Workshop on Analytics for Big Geospatial Data, BigSpatial@SIGSPATIAL 2013, Nov 4th, 2013, Orlando, FL, USA*, pages 73–80, 2013.

[92] Ranga Raju Vatsavai, Budhendra L. Bhaduri, Anil Cheriyadat, Lloyd F. Arrowood, Eddie A. Bright, Shaun S. Gleason, Carl Diegert, Aggelos K. Katsaggelos, Thrasos Pappas, Reid Porter, Jim Bollinger, Barry Chen, and Ryan Hohimer. Geospatial image mining for nuclear proliferation detection: Challenges and new opportunities. In *IGARSS*, pages 48–51, 2010.

[93] Mitko Veta, Josien P.W. Pluim, Paul J. van Diest, and Max A. Viergever. Breast Cancer Histopathology Image Analysis: A Review. *IEEE Trans Biomed Eng.*, 61(5):1400–1411, 2014.

[94] Hoang Vo, Ablimit Aji, and Fusheng Wang. SATO: A Spatial Data Partitioning Framework for Scalable Query Processing. In *Proceedings of the ACM Symposium on Advances in Geographic Information Systems, ACM SIGSPATIAL*, pages 545–548, November 2014.

[95] Huy T. Vo, Jonathan Bronson, Brian Summa, João Luiz Dihl Comba, Juliana Freire, Bill Howe, Valerio Pascucci, and Cláudio T. Silva. Parallel Visualization on Large Clusters using MapReduce. In *IEEE Symposium on Large Data Analysis and Visualization, LDAV*, pages 81–88, 2011.

[96] Fusheng Wang, Ablimit Aji, and Hoang Vo. High Performance Spatial Queries for Spatial Big Data: From Medical Imaging to GIS. *SIGSPATIAL Special*, 6(3):11–18, 2014.

[97] Guozhang Wang, Marcos Antonio Vaz Salles, Benjamin Sowell, Xun Wang, Tuan Cao, Alan J. Demers, Johannes Gehrke, and Walker M. White. Behavioral Simulations in MapReduce. *Proceedings of the VLDB Endowment, PVLDB*, 3(1):952–963, 2010.

[98] Jun Wang. Solving the multiple-instance problem: A lazy learning approach. In *In Proc. 17th International Conf. on Machine Learning*, pages 1119–1125. Morgan Kaufmann, 2000.

[99] Kai Wang, Jizhong Han, Bibo Tu, Jiao Dai amd Wei Zhou, and Xuan Song. Accelerating Spatial Data Processing with MapReduce. In *International Conference on Parallel and Distributed Systems*, pages 229–236, 2010.

[100] Kaibo Wang, Yin Huai, Rubao Lee, Fusheng Wang, Xiaodong Zhang, and Joel H Saltz. Accelerating pathology image data cross-comparison on CPU-GPU hybrid systems. *Proceedings of the VLDB Endowment*, 5(11):1543–1554, 2012.

[101] Kaibo Wang, Yin Huai, Rubao Lee, Fusheng Wang, Xiaodong Zhang, and Joel H Saltz. Accelerating pathology image data cross-comparison on CPU-GPU hybrid systems. *Proceedings of the VLDB Endowment*, 5(11):1543–1554, 2012.

[102] Shaowen Wang. A CyberGIS framework for the synthesis of cyberinfrastructure, GIS, and spatial analysis. *Annals of the Association of American Geographers*, 100(3):535–557, 2010.

[103] Shaowen Wang. CyberGIS and spatial data science. *GeoJournal*, 81(6):965–968, 2016.

[104] Shaowen Wang, Luc Anselin, Budhendra Bhaduri, Christopher Crosby, Michael F Goodchild, Yan Liu, and Timothy L Nyerges. CyberGIS software: a synthetic review and integration roadmap. *International Journal of Geographical Information Science*, 27(11):2122–2145, 2013.

[105] Shaowen Wang, Marc P Armstrong, Jun Ni, and Yan Liu. Gisolve: A grid-based problem solving environment for computationally intensive geographic information analysis. In *challenges of large applications in distributed environments, 2005. CLADE 2005. proceedings*, pages 3–12. IEEE, 2005.

[106] Shaowen Wang, Mary Kathryn Cowles, and Marc P Armstrong. Grid computing of spatial statistics: using the teragrid for g i*(d) analysis. *Concurrency and Computation: Practice and Experience*, 20(14):1697–1720, 2008.

[107] Shaowen Wang, Hao Hu, Tao Lin, Yan Liu, Anand Padmanabhan, and Kiumars Soltani. CyberGIS for data-intensive knowledge discovery. *SIGSPATIAL Special*, 6(2):26–33, 2015.

[108] Shaowen Wang and Yan Liu. Teragrid giscience gateway: bridging cyberinfrastructure and giscience. *International Journal of Geographical Information Science*, 23(5):631–656, 2009.

[109] Shaowen Wang, Yan Liu, and Anand Padmanabhan. Open cybergis software for geospatial research and education in the big data era. *SoftwareX*, 2015.

[110] Shaowen Wang and Xin-Guang Zhu. Coupling cyberinfrastructure and geographic information systems to empower ecological and environmental research. *BioScience*, 58(2):94–95, 2008.

[111] Randall T. Whitman, Michael B. Park, Sarah A. Ambrose, and Erik G. Hoel. Spatial Indexing and Analytics on Hadoop. In *SIGSPATIAL*, 2014.

[112] Dawn J Wright and Shaowen Wang. The emergence of spatial cyberinfrastructure, 2011.

[113] Jin Soung Yoo, Douglas Boulware, and David Kimmey. A parallel spatial co-location mining algorithm based on mapreduce. In *Big Data (BigData Congress), 2014 IEEE International Congress on*, pages 25–31. IEEE, 2014.

[114] Simin You, Jianting Zhang, and Le Gruenwald. Parallel Spatial Query Processing on GPUs Using R-trees. In *Proceedings of the 2Nd ACM SIGSPATIAL International Workshop on Analytics for Big Geospatial Data*, BigSpatial '13, pages 23–31, New York, NY, USA, 2013. ACM.

[115] Jia Yu, Mohamed Sarwat, and Jinxuan Wu. GeoSpark: A Cluster Computing Framework for Processing Large-Scale Spatial Data. In *Proceedings of the ACM Symposium on Advances in Geographic Information Systems, ACM SIGSPATIAL*, Seattle, WA, nov 2015.

[116] Chi Zhang, Feifi Li, and Jeffrey Jestes. Efficient Parallel kNN Joins for Large Data in MapReduce. In *Proceedings of the International Conference on Extending Database Technology, EDBT*, pages 38–49, 2012.

[117] Jianting Zhang and Simin You. Speeding up large-scale point-in-polygon test based spatial join on GPUs. In *Proceedings of the 1st ACM SIGSPATIAL International Workshop on Analytics for Big Geospatial Data*, pages 23–32. ACM, 2012.

[118] Shubin Zhang, Jizhong Han, Zhiyong Liu, Kai Wang, and Shengzhong Feng. Spatial Queries Evaluation with MapReduce. In *Proceedings of the International Conference on Grid and Cooperative Computing*, pages 287–292, 2009.

[119] Shubin Zhang, Jizhong Han, Zhiyong Liu, Kai Wang, and Zhiyong Xu. SJMR: Parallelizing spatial join with MapReduce on clusters. In *Proceedings of the IEEE International Conference on Cluster Computing and Workshops*, pages 1–8, 2009.

[120] Lingjun Zhao, Lajiao Chen, Rajiv Ranjan, Kim-Kwang Raymond Choo, and Jijun He. Geographical information system parallelization for spatial big data processing: a review. *Cluster Computing*, 19(1):139–152, 2016.

[121] Weizhong Zhao, Huifang Ma, and Qing He. Parallel *K*-Means Clustering Based on MapReduce. In *CloudCom 2009*, pages 674–679, 2009.