[DA-046] Computational Geography

Abstract

Computational Geography emerged in the 1980s in response to the reductionist limitations of early GIS software, which inhibited deep analyses of rich geographic data. Today, Computational Geography continues to integrate a wide range of domains to facilitate spatial analyses that require computational resources or ontological paradigms beyond that made available in traditional GIS software packages. These include novel approaches for the mass creation of geospatial data, large-scale database design for the effective storage and querying of spatial identifiers (i.e., distributed spatial databases), and methodologies which enable simulations and/or analysis in the context of large-scale, frequently near-real-time, spatially-explicit sources of information. The topics studied within Computational Geography directly enable many of the world's largest public databases, including Google Maps and Open Street Map (OSM), as well as many modern analytic pipelines designed to study human behavior with the integration of large volumes of location information (e.g., mobile phone data) with other geospatial sources (e.g., satellite imagery).

Keywords: data capture, data science, geocomputation, geospatial data science, modeling, spatial analytics, spatial data management, spatial data model, spatial data science

Author & citation

Runfola, D. (2022). Computational Geography. The Geographic Information Science & Technology Body of Knowledge (1st Quarter 2022 Edition), John P. Wilson (Ed.). DOI: 10.22224/gistbok/2022.1.7.

This entry was first published on March 7, 2022. No earlier editions exist.

Explanation

1. Overview
2. Optimizing Spatial Data Indexing and Retrieval in a Computational Geography Framework
3. Primitive Geospatial Operators in Partitioned and Parallelized Environments
4. Spatially Distributed Analyses
5. Tools & Techniques Common to Computational Geography

1. Overview

The term "Computational Geography" was coined by David Mark and colleagues at the National center for Geographic Information and Analysis in the 1980s, and formalized by Stan Openshaw with the founding of The Centre for Computational Geography at University of Leeds (Batty, 2020; for a broader review on the history of quantitative analysis in geography, see Mark, 2003). Used interchangeably with "GeoComputation", the computational geography movement emerged in response to reductionist decisions made to simplify spatial data creation, retention and analysis during the emergence of Geographic Information Systems (GIS) through the 1980s (Gahegan & GeoComputation

International Steering Group, 2001). While many of these early limitations in GIS capabilities have been improved over the last four decades, Computational Geographers continue to engage with challenges that are technically or ontologically infeasible to implement in contemporary GIS software platforms. Today, the overlap between computational geography and GeoComputation remains substantial, with the core distinction between the groups being an emphasis on computer engineering and optimization at a database or system-level within the computational geography community (i.e., building general purpose distributed spatial databases (Hughes et al. 2015), primitive operators in parallel environments (Shook et al. 2016), or new strategies for distributed processing (Worboys & Duckham 2006)) as contrasted to a more frequent focus on model-specific optimizations in the GeoComputation community (i.e., deriving novel ways to distribute agent based models).

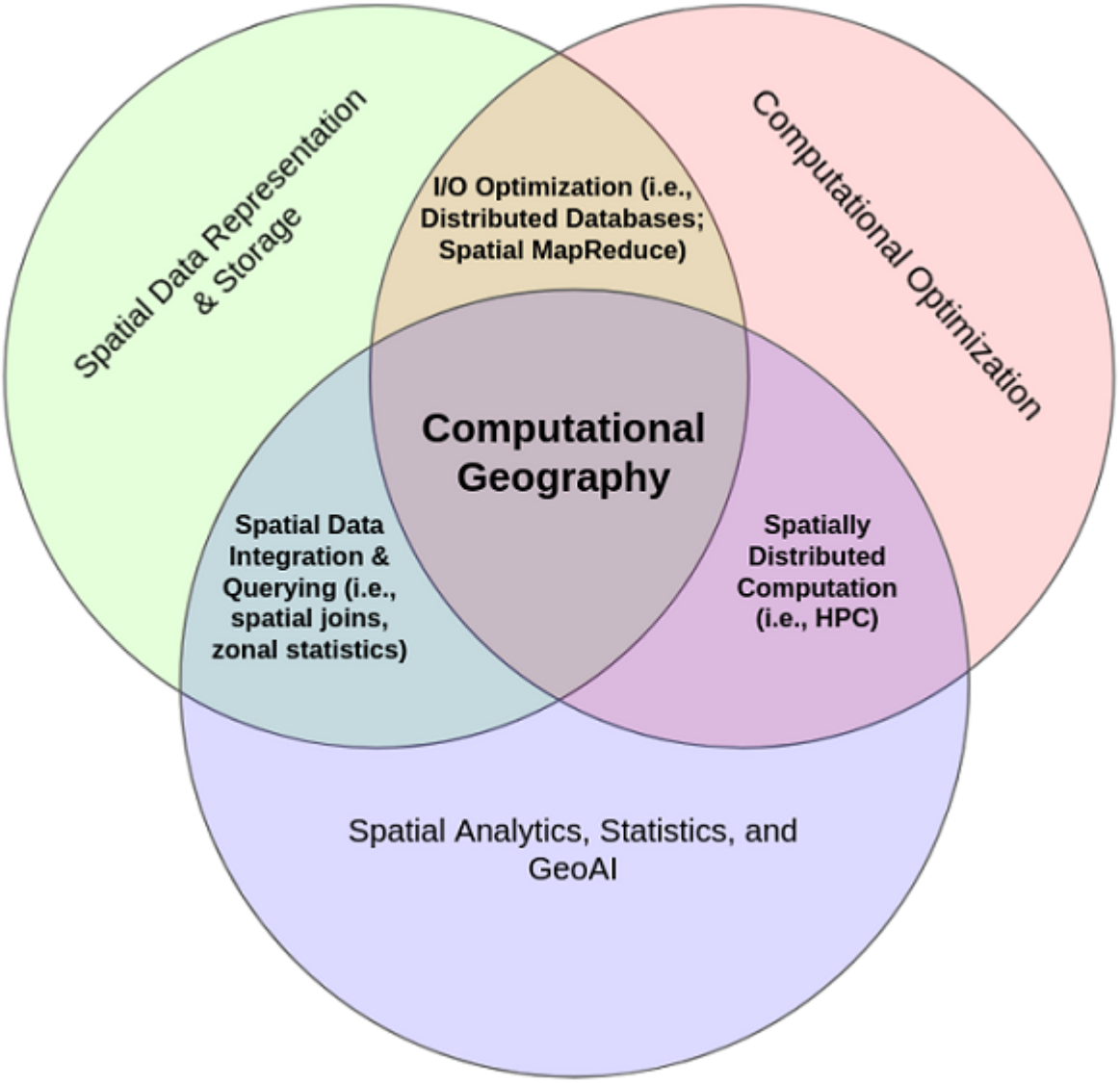


Figure 1. Diagram illustrating topics comprising the field of Computational Geography, circa 2021. Source: author.

The topics that Computational Geography has engaged with have been redefined over

time, focusing on discovery-oriented research that augment or improve contemporary, tradecraft-driven GIS capabilities. Alongside this evolution, the disciplinary contributors to Computational Geography have also shifted – growing to incorporate not only geographers with strong computational skills, but also information and computer scientists that engage with spatial data sources. This interdisciplinary shift is reflected in academia today, with the two programs offering Ph.D. degrees in Computational Geography both sitting outside of Geography departments (Texas A&M, 2021; William & Mary, 2021).

The rapid increase in available information has resulted in both practical and theoretical challenges to traditional GIS-based modalities of inquiry – ranging from the inability of modern desktop GIS software to handle large datasets in non-distributed environments, to ontological and technical discussions around how spatial datasets can be more helpfully represented than allowed for in common GIS frameworks (Bostock et al., 2017). Many tools and techniques derived by computational geographers have, in response, begun to provide solutions to these challenges. These contributions have been at the nexus of three topical areas: (1) spatial data representation and storage, (2) spatial analytics, statistics, and GeoAI, and (3) computational optimization. Recognizing that each of these topical areas warrant individual articles, here we focus specifically on contributions at the intersections of each of these topical areas (see figure 1): spatial data indexing and retrieval, distributed spatial data integration and querying, and spatially distributed analytics.

2. Optimizing Spatial Data Indexing and Retrieval in a Computational Geography Framework

As the volume and nature of spatial information has grown and changed, so too have the technical capabilities required to record and query that data. Within the GIS community, the development of Spatial Database Management Systems (SDBMS) enabled new data models, storage, and querying techniques based on geographic information, providing new capabilities for analysis in both GIS desktop and programmatic environments. In parallel to these efforts, the computer science and computational geography communities sought to implement spatial querying functionality into traditional, a-spatial database environments. These efforts - inclusive of software engineering (i.e., Hughes et al. 2015) and the development of new referencing systems (Tsui 1997) - have today resulted in a wide range of solutions for optimizing spatial data indexing and retrieval, depending on the task the researcher is most interested in.

The addition of spatial query functionality to relational databases was one of the earliest developments, largely due to the prevalence of relational databases in enterprise data storage systems. With each data entity represented as a row, a range of techniques were derived to add a geometry column of information, as well as implement algorithms to select rows based on the values contained within these geometry column. These techniques continue to be advanced to this day, with recent research focused on the parallelization or distribution of spatial queries (Giannousis et al., 2019; Ilba, 2021) and optimal spatial indexing strategies in relational databases (Chaves Carniel et al., 2018; Schön et al., 2013).

Distinct from relational databases, document-oriented approaches sought to enable a more flexible data acquisition strategy, in which an explicit schema does not need to be defined a-priori. This allows for – for example – the addition of new types of data collection for some

units within the database, alongside a range of other benefits. Novel algorithms to enable spatial indexing and searching within document store databases were developed alongside many of the databases themselves (i.e., MongoDB's 2dsphere), to mixed effect dependent on the computational task of interest (Bartoszewski et al., 2019; Makris et al., 2021). Research into improving the efficiency of spatial querying in document-oriented databases continues today (Makris et al. 2021; Xiang et al., 2016; Yan et al., 2016; Zhu & Gong, 2014).

In addition to relational and document-oriented database approaches, key-value stores have recently been a focus of research. The advantage of key-value stores lies in their underlying, column-based structure, which are highly efficient for the processing of sparse datasets. Further, most key-value stores are designed to operate efficiently in highly distributed environments. The majority of research into this topic focuses on an implementation of spatial querying on top of the popular hBase and Accumulo databases, with a range of algorithms supporting spatial querying and indexing in this paradigm being derived. The most popular of these is the GeoMesa project (Hughes et al., 2015).

For researchers interested in non-traditional modalities for spatial analysis, a small number of graph-based databases have been created in which network relationships can be conceptualized as spatial, social, or other factors. Most common of these is Neo4j, which enables unstructured entity characterization, with entities being members of one or a series of inter-related networks with defined nodes and edges (Webber, 2012). Neo4j-Spatial specifically enables querying across a network defined by geography, including common operations such as searching within specified regions or distances from a point (Taverner, 2012). Other graph-based databases with spatial functionality include RedisGraph and AllegroGraph.

A wide range of more nascent frameworks for distributed spatial databases have recently been proposed by the community, built on or combining the technologies noted above. These include SpatialSpark, GeoSpark, Simba, LocationSpark, SparkGIS, TrajSpark, DITA, Gragoon, and likely many more. A full survey of these tools, and their contributions to this growing suite of techniques, can be found in Alam et al., 2021.

3. Primitive Geospatial Operators in Partitioned and Parallelized Environments

The challenge of representing spatial data continues to evolve, with different conceptions promoting markedly different definitions of spatial boundaries (i.e., fuzzy boundaries), geographic relationships (i.e., network representations), and geographic attributes (i.e., fuzzy class membership vs. discrete; see Goodchild et al. 1998). As the volume of geographic data has grown, accessing information that may be collected and stored across a broad range of modalities is a core challenge. The most common solutions today involve the distribution of primitive geospatial operations across multiple computer nodes, aggregating, integrating, or selecting data that may be of use for a particular analysis.

An intuitive example of this challenge can be explored with a traditional "Zonal Statistics" operation, in which a grid of values (i.e., temperature measured every 500 meters over a surface) is aggregated to a coarser geographic region (i.e., a country). This is occasionally a necessary first-step in an analysis which seeks to understand the relationship(s) between one geospatial dataset – i.e., precipitation – and another – i.e., country boundaries. In cases with millions or billions of points, the time costs of averaging individual values across an

entire country on an individual computer can be untenable, stretching into years, decades, or more. As an alternative, multiple computers can load partitions of the data, aggregate the values for a set of points, and then relay their findings to a centralized node to produce the final calculation (Goodman et al., 2019; J. Zhang et al., 2014). To support the wide range of different geospatial operations, implementations of primitive geospatial operations in parallel environments have included point-in-region searches (Kondor et al., 2014; Priya & Kalpana, 2018; Tarmur & Ozturan, 2019), spatial joins (Aghajarian & Prasad, 2017; You et al., 2015), clips (Puri & Prasad, 2015), and a wide range of techniques – frequently based on the R-Tree algorithm - designed to automatically partition spatial data for distribution across nodes for general application (Roumelis et al., 2017, 2019; You et al., 2013; J. Zhang & You, 2013). These techniques provide the “glue” between database architectures and analyses (i.e., machine learning or simulation models) a researcher may ultimately seek to perform.

4. Spatially Distributed Analyses

The demand for spatial data analysis is growing from a range of industries, but the computational costs (i.e., time required to process on a computer core) of geometry-aware algorithms has inhibited this growth. One area of intensive inquiry to overcome this bottleneck has been in improving our capability to distribute spatial analytic models across a range of processors or computational nodes (i.e., spatial parallel or distributed computing). The models being distributed can be highly variable in nature – ranging from agent based simulations to convolutional neural network (i.e., Brewer et al., 2021; Goodman et al., 2020) or other machine learning approaches. Two approaches have been taken: specialized solutions specific to a single typology of analytic model, and generalized approaches that seek to distribute data across any arbitrary target model.

Model-specific distribution approaches have emerged in a number of subfields. One of the most prominent of these have been agent-based simulation and modeling efforts, with packages such as OpenABL (Cosenza et al., 2018) providing easily accessible, distributed approaches for a range of primitive functions required by the agent based modeling community. OpenABL built heavily on a long lineage of ABM-specific distributed systems, including the well-known REPAST, REPAST-HPC, Mason, D-Mason, Flame, and more (see Cosenza et al., 2018, for a full review of the evolution of these and related efforts).

The remote sensing community has been highly active in the development of distributed analytic models, predominantly for large-scale classification of satellite imagery as a part of near-real-time algorithms (Hawick & James, 1997), and on-demand processing of archival imagery (Yang et al., 2005; M. Zhang et al., 2015). Building on these approaches, recent research has begun to focus on the analysis of satellite imagery in distributed environments using deep learning models – i.e., distributing across infrastructure such as GPUs specialized for deep learning algorithms (Li & Choi, 2021; Sedona et al., 2019).

Augmenting these specialized approaches, general frameworks to enable distributed spatial model analyses have included GeoBeam (He et al., 2019), Niharika (Ray et al., 2013), and a number of novel implementations of R-tree indexing to promote concurrent spatial operations (Dai, 2009). More broadly, distributed frameworks such as Dask and Spark have also been used as flexible engines to assist in the distribution of spatial models across

arbitrary numbers of nodes (i.e., Erlacher et al., 2021).

5. Tools & Techniques Common to Computational Geography

Today, study in Computational Geography requires a depth of understanding of topics which include:

- GPU and CPU based parallelization and distribution of spatial algorithms.
- Distributed database and file system architectures, and their pros/cons for use in the context of spatial data (i.e., the implications of partitioning strategies in common architectures such as HDFS, hBase, Accumulo, Sharded Mongo, and Hive).
- Distinctions and use cases for different database technologies in the context of spatial querying.
- Computational theory regarding algorithm optimization and fundamental search and sort strategies.
- Fundamental GIS knowledge (i.e., data representation, projection, topology, data types).
- Fundamental Remote Sensing knowledge (i.e., atmospheric correction, physical reflectance).
- Understanding of different spatial statistical techniques, inclusive of spatial regression models and other autocorrelative techniques (i.e., kriging strategies).
- Understanding of machine learning models such as SVM, KNN, Regression Trees, and ANNs.
- Understanding of computer vision techniques, especially convolutional networks.

Outside of the small selection of computational geography programs today, most scholars in the field acquire this diverse range of skills through interdisciplinary collaboration, or through multiple degrees in interrelated fields (i.e., computer science or data science and geography).

References

Aghajarian, D., & Prasad, S. K. (2017). A Spatial Join Algorithm Based on a Non-uniform Grid Technique over GPGPU. GIS: Proceedings of the ACM International Symposium on Advances in Geographic Information Systems, 2017-November.

Alam, M. M., Torgo, L., & Bifet, A. (2021). A Survey on Spatio-temporal Data Analytics Systems. ACM Comput. Surv, 1, 44.

Bartoszewski, D., Piorkowski, A., & Lupa, M. (2019). The Comparison of Processing Efficiency of Spatial Data for PostGIS and MongoDB Databases. Communications in Computer and Information Science, 1018, 291–302.

Batty, M. (2020). Foreword I: Charting Computational Social Science from a Spatial Perspective. In Spatial Synthesis (pp. 3–5).

Bostock, M., Davies, J., & Stucki, J. (2017). Topojson. NYT Graphics, D3.

Brewer, E., Kemper, P., Lin, J., Hennin, J., & Runfola, D. (2021). Predicting Road Quality using High Resolution Satellite Imagery: A Transfer Learning Approach. PLoS One.

Carniel, A. C., Ciferri, R. R., Vassilakopoulos, M., Roumelis, G., Corral, A., & Dutra De Aguiar Ciferri, C. (2018). An Efficient Flash-aware Spatial Index for Points. Proceedings XIX GEOINFO, December 05-07, 2018, Campina Grande, PB, Brazil.

Cosenza, B., Popov, N., Juurlink, B., Richmond, P., Chimeh, M. K., Spagnuolo, C., Cordasco,G., & Scarano, V. (2018). OpenABL: A Domain-Specific Language for Parallel and Distributed Agent-Based Simulations. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics),11014 LNCS, 505–518.

Dai, J. (2009). Efficient Concurrent Operations in Spatial Databases. Unpublished doctoral dissertation, Virginia Polytechnical Institute and State University [Virginia Tech].

Erlacher, C., Anders, K.-H., Jankowski, P., Paulus, G., & Blaschke, T. (2021). A Framework for Cloud-Based Spatially-Explicit Uncertainty and Sensitivity Analysis in Spatial Multi-Criteria Models. ISPRS International Journal of Geo-Information 10(4), 244.

Gahegan, M., & GeoComputation International Steering Group. (2001). GeoComputation.

Giannousis, K., Bereta, K., Karalis, N., & Koubarakis, M. (2019). Distributed Execution of Spatial SQL Queries. Proceedings - 2018 IEEE International Conference on Big Data, Big Data 2018, 528–533.

Goodchild, M. F., Montello, D. R., Fohl, P. and Gottsegen, J. (1998). Fuzzy spatial queries in digital spatial data libraries. 1998 IEEE International Conference on Fuzzy Systems Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98CH36228), Anchorage, AK, USA, 1998, pp. 205-210 vol.1.

Goodman, S., BenYishay, A., & Runfola, D. (2020). A convolutional neural network approach to predict non-permissive environments from moderate-resolution imagery. Transactions in GIS, 25(2), 674–691.

Goodman, S., BenYishay, A., Lv, Z., & Runfola, D. (2019). GeoQuery: Integrating HPC systems and public web-based geospatial data tools. Computers and Geosciences, 122,103–112.

Hawick, K. A. & James, H. A. (1997). Distributed high-performance computation for remote sensing. SC '97: Proceedings of the 1997 ACM/IEEE conference on Supercomputing. pp 1-13.

He, Z., Liu, G., Ma, X., & Chen, Q. (2019). GeoBeam: A distributed computing framework for spatial data. Computers & Geosciences, 131, 15–22.

Hughes, J. N., Annex, A., Eichelberger, C. N., Fox, A., Hulbert, A., & Ronquest, M. (2015). GeoMesa: a distributed architecture for spatio-temporal fusion. Geospatial

Informatics, Fusion, and Motion Video Analytics V, 9473(21), 94730F.

Ilba, M. (2021). Parallel algorithm for improving the performance of spatial queries in SQL:The use cases of SQLite/SpatiaLite and PostgreSQL/PostGIS databases. Computers & Geosciences, 155, 104840.

Kondor, D., Dobos, L., Csabai, I., Bodor, A., Vattay, G., Budavári, T., & Szalay, A. S. (2014). Efficient classification of billions of points into complex geographic regions using hierarchical triangular mesh. SSDBM '14: Proceedings of the 26th International Conference on Scientific and Statistical Database Management. 4:1-4.

Li, G., & Choi, Y. (2021). HPC cluster-based user-defined data integration platform for deep learning in geoscience applications. Computers & Geosciences. 155: 104868.

Makris, A., Tserpes, K., Spiliopoulos, G., Zissis, D., and Anagnostopoulos, D. (2021). MongoDB vs PostgreSQL: A comparative study on performance aspects. Geoinformatica , 25, 268.

Mark, D. (2003). Geographic Information Science: Defining the Field. In Foundations of Geographic Information Science. Eds: Duckham, M., Goodchild, M.F., and Worboys, M.F. Taylor and Francis, New York, NY, USA. pp. 1-17.

Priya, M., & Kalpana, R. (2018). Distributed processing of location based spatial query through vantage point transformation. Future Computing and Informatics Journal, 3(2), 296–303.

Puri, S., & Prasad, S. (2015). A Parallel Algorithm for Clipping Polygons with Improved Bounds and a Distributed Overlay Processing System Using MPI, 15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid).

Ray, S., Simion, B., Brown, A. D., & Johnson, R. (2013). A parallel spatial data analysis infrastructure for the cloud. GIS: Proceedings of the ACM International Symposium on Advances in Geographic Information Systems, 274–283.

Roumelis, G., Vassilakopoulos, M., Corral, A., & Manolopoulos, Y. (2017). Efficient query processing on large spatial databases: A performance study. Journal of Systems and Software, 132, 165–185.

Roumelis, G., Velentzas, P., Vassilakopoulos, M., Corral, A., Fevgas, A., & Manolopoulos, Y. (2019). Parallel processing of spatial batch-queries using xBR+ trees in solid-state drives. Cluster Computing 23:3, 23(3), 1555–1575. DOI: 10.1007/S10586-019-03013-0

Schön, B., Mosa, A. S. M., Laefer, D. F., & Bertolotto, M. (2013). Octree-based indexing for 3D pointclouds within an Oracle Spatial DBMS. Computers & Geosciences, 51, 430–438.

Sedona, R., Cavallaro, G., Jitsev, J., & Strube, A. (2019). Remote Sensing Big Data

Classification with High Performance Distributed Deep Learning. Remote Sensing, 11(24).

Shook, E., Hodgson, M. E., Wang, S., Behzad, B., Soltani, K., Hiscox, A. and Ajayakumar, J. (2016). Parallel cartographic modeling: a methodology for parallelizing spatial data processing. International Journal of Geographical Information Science 30 (12):2355–2376.

Tarmur, S., & Ozturan, C. (2019). Parallel classification of spatial points into geographical regions. Proceedings - 2019 18th International Symposium on Parallel and Distributed Computing, ISPDC 2019, 9–15.

Taverner, C. (2012). Neo4j Spatial. Neo Technology.

Texas A&M University. (2021). Geospatial Computer Science, PhD | Texas A&M University-Corpus Christi.

Tsui, P. H. Y. and Brimicombe, A. J. (1997). The hierarchical tessellation model and its use in spatial analysis. Transactions in GIS 2 267-279.

Webber, J. (2012). A programmatic introduction to Neo4J. SPLASH'12 - Proceedings of the 2012 ACM Conference on Systems, Programming, and Applications: Software for Humanity, 217.

William & Mary. (n.d.). Graduate Program in Applied Science. Accessed 2021.

Worboys, M. F., & Duckham, M. (2006). Monitoring qualitative spatiotemporal change for geosensor networks. International Journal of Geographical Information Science, 20(10),1087-1108.

Xiang, L., Huang, J., Shao, X., & Wang, D. (2016). A MongoDB-Based Management of Planar Spatial Data with a Flattened R-Tree. ISPRS International Journal of Geo-Information, Vol. 5, Page 119, 5(7), 119.

Yan, L., Dongho, K., & Byeong-Seok, S. (2016). Geohashed Spatial Index Method for a Location-Aware WBAN Data Monitoring System Based on NoSQL. Journal of Information Processing Systems, 12(2), 263–274.

Yang, Y., Rana, O. F., Walker, D. W., Williams, R., Georgousopoulos, C., Caffaro, M., & Aloisio, G. (2005). An agent infrastructure for on-demand processing of remote-sensing archives. International Journal on Digital Libraries 5:2, 5(2), 120–132.

You, S., Zhang, J., & Gruenwald, L. (2013). Parallel spatial query processing on GPUs using R-trees. Proceedings of the 2nd ACM SIGSPATIAL International Workshop on Analytics for Big Geospatial Data, Big Spatial 2013, 23–31.

You, S., Zhang, J., & Gruenwald, L. (2015). Spatial Join Query Processing in Cloud: Analyzing Design Choices and Performance Comparisons. Proceedings of the International Conference on Parallel Processing Workshops, 2015-January, 90–97.

Zhang, J., & You, S. (2013). High-performance quadtree constructions on large-scale geospatial rasters using GPGPU parallel primitives. 27(11), 2207–2226.

Zhang, J., You, S., & Gruenwald, L. (2014). Parallel online spatial and temporal aggregations on multi-core CPUs and many-core GPUs. Information Systems, 44, 134–154.

Zhang, M., Wang, H., Lu, Y., Li, T., Guang, Y., Liu, C., Edrosa, E., Li, H., & Rishe, N. (2015). TerraFly GeoCloud: An Online Spatial Data Analysis and Visualization System. ACM Transactions on Intelligent Systems and Technology (TIST), 6(3).

Zhu, Y. & Gong, J. (2014). A real-time trajectory indexing method based on MongoDB. 2014 11th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), Xiamen, China, 2014, pp. 548-553