Boosting the Performance of Web-based GVE systems: Lessons and a New System Design on Biodiversity Explorations

Jianting Zhang

Abstract—We consider Query Driven Visual Exploration (QDVE) techniques as performance amplifiers of Advanced Geospatial Visualization (AGV) and Geospatial Visual Analytics (GVA). By examining the three Web-based Geospatial Visual Exploration (GVE) systems that we have designed and implemented for different application domains and geospatial data types, we recommend performance boosting techniques for Web-based GVE systems. We further present a new system design to visually explore the complex relationships among geographical, environmental and taxonomic components of global biodiversity data with higher level of performance. In addition to leveraging existing performance boosting techniques, the new deign utilizes data parallel computing power of GPUs that are already available on commodity personal computers for indexing and query processing.

Index Terms—High-Peroramnce, Visual Exploration, indexing, query processing, parallel computing, GPGPU, biodiversity

1. Introduction

Advanced Geospatial Visualization (AGV), Query Driven Visual Exploration (QDVE) of large-scale geospatial data and Geospatial Visual Analytics (GVA) have attracted significant research and application interests over the past few years. We consider QDVE a performance amplifier of AGV and GVA where the focus is on enabling efficient queries on large-scale geospatial data to help users quickly identify and locate interesting geospatial phenomena and their distributions in a visual manner. QDVE is strongly related to high-performance GIS and Spatial Databases (SDB) as response times are crucial to the success of a QDVE system when applied to large-scale geospatial data. Despite the different focuses among AGV, QDVE and GVA techniques, there are strong relationships among them when handling large-scale geospatial data (QDVE may be irrelevant when handling small-scale data). Systems targeted at practical applications may involve all the three categories of techniques. As such, we use the term Geospatial Visual Explorations (GVE) to refer techniques that require efficient query processing (possibly through indexing) on large-scale geospatial data to speed up both AGV and GVA applications.

Similar to evolving desktop GIS to Web-based GIS, there are practical needs to migrate desktop based GVE systems to Web-based ones so that users can access GVE systems anytime and anywhere using mainstream Web browsers. However, due to the limited Internet bandwidths and limited computing resources allocated to Web browsers, developing a high-performance Web-based GVE system is more challenging than desktop-based ones. In this paper, we discuss several techniques that we have explored in building a few Web-based GVE systems for multiple types of geospatial data targeted at different application domains. We then present the design of a new Web-based GVE system to facilitate exploring and analyzing the relationships among the taxonomic-geographical-environmental components of global biodiversity data both individually and collaboratively in distributed computing environments. Due to the large data volumes, versatile data types and complex relationships, the system performance is critical to achieve the design goals. The GVE system is under active developments but we would like to share our designs and implementation progresses with the research community during the workshop.

2. High Performance GVE Systems: From Desktop to Web-based

Over the past few years, we have built a few systems to facilitate visualizing, visual exploring and visual analytics of geospatial data, ranging from remotely sensing data [1,2] to biodiversity data [3, 4]. These systems are desktop-based and some of them integrate open source GIS (such as JUMP) and visualization (such as Prefuse) software packages for fast developments[i]. Since the data involved in the four systems are relatively small and they are run in a desktop computing environment, the designs and developments mostly focused on functionality without considering performance. Our recent research has been focusing more on Web-based GVE where performance becomes a significant issue.

The NYC CrashMap project[ii] jointly developed with the New York City Department of Transportation (NYCDOT) used commercial software (including ESRI ArcGIS and Microsoft SQL Server) and adopted the ArcGIS API for Flex for client side visualization. There are more than 1.7 million traffic crashes in NYC over a 20 years period. Interactively querying and visualizing to explore the distributions of large amounts of crashes and their relationships with geographical locations, time, weather conditions and other auxiliary geospatial and non-geospatial data have imposed significant performance challenges. Following the classic Information Seeking Mantra – "Overview First, filter and zoom and details on demand" [5], we have applied a technique to boost the overall system performance. The idea is to pre-aggregate crash records at the coarse spatial scales so that only aggregated records need to be retrieved from the server side and visualized at the client side at the coarser levels. The idea is similar to the image tiling technique that has been widely used in Internet Map services but applied to vector geospatial data. Due to the limited numbers of records at the coarse spatial scales, the tradeoffs between storage requirements of pre-aggregations and system response times are well balanced. Another effective technique is materializing spatial relationships between crash locations and various administrative zones to avoid runtime spatial joins which are fairly expensive.

The second Web-based GVE system[iii] we have developed is motivated by lacking query capability in Google Map/Earth and similar Internet mapping products. Although Internet mapping technologies and products such as Google Map/Earth have been increasingly used to visualize research data products by scientists, they do not provide functionality to allow querying the data products and identify Region of Interests (ROIs), which is available in many desktop based GIS systems. Our solution is to build a main-memory index called Binned Min-Max Quadtree (or BMMQ-Tree for short) at the server side to speed up query processing. Given a value range, such as precipitation between [p1,p2), the BMMQ-Tree is queried and quadrants that satisfy the query criteria are sent back to Web clients for highlighting purposes [6]. While the server-side indexing technique did work for efficient querying purposes, the data volumes representing the quadrants in a query result set can still be signficant and drawing the quadrants can also impose a signficant overhead on a Web browser. As such, a dynamic tiling approach [7] is further developed to convert the resultant quadrants into tiled binary images on-the-fly. The tiled binary images are organized as Open Geospatial Consortium (OGC) compliant Web map services so that they can be overlaid with original image/raster data. The lessons we have learnt in the development of the raster GVE system are the following. First, efficient indexing is important in speeding up query processing. Many tree-based indexing approaches can reduce linear scan to logarithmic searching. Second, it is important to reduce data volumes in query result sets as they need to be sent over the Internet to Web clients which typically is about 2-3 orders slower than local disk accesses (100K versus 100M bytes per second). As such, proper encoding/compression is necessary to boost the overall performance. Third, as most Web browsers are designed to handle images more efficiently than vectors, it would be better to encode the query results as images so that they can be easily visualized in Web browsers for various purposes.

Our third Web-based GVE system[iv] is targeted at the biodiversity research using the 4000+ bird distribution maps in the whole West hemisphere. While it is straightforward to employ the mainstream Web-GIS technologies by putting all the distribution maps in a spatial database and connect an Internet mapping software with the spatial database to query and visualize the species distributed in a query window, the performance can be very poor as intersecting a query window with millions of complex polygons is very computationally intensive. The polygons are also highly overlapped due to the existences of many birds that have wide geographical distributions. This makes classical spatial indexing approaches highly inefficient. We have adopted two approaches to tackle the problems. The First approach rasterizes polygons into quadrants at the different levels according to a hierarchical raster tessellation (or quadtree tessellation) and associate species identifiers with the highest levels of quadrants that fully cover the distributions at the quadrants. As detailed in [8], the number of quadrants is significantly reduced by the technique. The hierarchical quadrants expressed as tree paths and the associated species identifier lists represented as arrays which can be efficiently indexed by PostgreSQL. This approach transforms spatial queries into tree matching queries to avoid inefficient indexing on overlapping polygons which has been proven to be effective. As the open source PostgreSQL does not support parallel query evaluations yet, we have also developed a technique that can effectively utilize the multiple CPU cores available on the server for experiments. Basically a query window is decomposed into multiple ones according to the quadtree tessellation and thus multiple query evaluations can be performed in parallel as different database processes with each CPU core will handles one database process. The parallel query results are then combined to formulate the final query results. The second approach takes advantage of the increasingly large and inexpensive main memory capacities on commodity computers and loads the whole quadtree in main memory [9]. A query processing module is then developed to recursively traverse the quadtree to collect species identifiers that are associated with different levels of quadrants. Although the first approach is able to speed up query processing significantly when compared with the naïve approach, the main-memory based approach reduces the query responses times to an average of 10 milliseconds for 400 randomly generated query windows which is sufficiently fast for visual explorations even in a Web environment with a reasonable Internet data communication speed. The lessons that we have learnt are two-folds. First, although disk-resident databases technologies are currently used to provide query processing backend for many Web based applications, main-memory databases that are optimized for domain-specific applications can be orders faster. Given that large capacity CPU memories are increasingly becoming affordable, main-memory database technologies should be exploited in GVE applications on large scale datasets. This is very similar to the lesson we have learnt in the second GVE system although different types of main-memory based indexing and query processing techniques have been utilized. Second, parallel processing techniques can have a great potential to speed up query processing using both existing software stacks and new parallel-aware software stacks. A practically useful approach, as the first approach we have used in developing the GVA system, is to decompose large query tasks into smaller ones and make multiple CPU cores process the smaller tasks in parallel. While the approach is inferior to parallel databases that can make use of multiple CPU cores natively, such a practical approach often can be beneficial in boosting end-to-end performance in GVE applications.

In summary, based on the lessons we have learnt from developing the three GVE systems for exploring large-scale geospatial data, including traffic crash data (points), global precipitation data (rasters) and birds distributional data (polygons), we recommend the following techniques to boost the performance of Web-based GVE applications. These techniques are prioritized based on the cost-effectiveness according to our experiences: (1) making full use of main-memory based indexing structures and query processing algorithms. As many GIS and visualization packages, such as JUMP and Prefuse, are mostly main-memory based, this recommendation does not seem to be new. However, we argue that cache misses can significantly affect the performances of main-memory data structures and algorithms especially when handling large-scale datasets. Cache conscious designs are needed for GVE systems on large-scale data that require higher performance. (2) performing application-specific materialization and aggregations and balancing between storage overheads and performance gains. The effectiveness can be further enhanced by client side caching and/or storing the "hot" data in faster media (e.g., flash memory or CPU memory). (3) encoding query results properly to reduce data transfer time over the Internet and client side rendering time. Using compressed images that are organized according to Web standards (e.g., OGC WMS) is a good solution based on existing Web technologies and Web-GIS practices. (4) using parallel computing whereas possible. This includes both tightly coupled parallel systems on multicore CPUs and manycore GPUs as well as distributed shared-nothing cluster computers that are made of identical commodity computing nodes. Quite a few parallel programming schemes, such as OpenMP on multicore CPUs, CUDA on GPUs and MapReduce on cluster computers, are already widely available. The key to the success of adopting this technique is to identify the inherent parallelisms in query processing and design viable schemas for problem decomposition and result combination.

3 HIGH-PERFORMANCE GVE OF BIODIVERSITY DATA: A NEW SYSTEM DESIGN

3.1 Motivations and Backgrounds

We are motivated to design a better high-performance GVE system for more comprehensive and effective explorations of global biodiversity data with the following driving forces. First, the available biodiversity data is growing fast. Several enabling technologies have made biodiversity data available at much finer scales in the past decade, including DNA barcoding for species identification, geo-referring for converting descriptive museum records to geographical coordinates, database technologies for managing species presence locations and related taxonomic and environmental data, and, GIS for species distribution data modeling and analysis. The newly emerging cyberinfrastructure technologies have made exchanging and sharing species distribution data over the Web much easier. On the environmental data side, advancements in remote sensing technology and instrumentation have generated huge amounts of remotely sensed imagery. Still yet, numerous environmental models have generated even larger volumes of geo-referenced raster model output data. The increasing resolutions and data volumes of both species distribution data and environmental data have made it possible to discover new patterns and foster new theories in exploring species distributions and analyzing biodiversity patterns..

Second, parallel hardware architectures become the state-of-the-art. Virtually all computing devices, ranging from desktop CPUs, GPUs to handheld mobile devices, are parallel hardware and it is even difficult to find uniprocessors in mainstream commodity computers now days. A reasonably current workstation can have more than a dozen CPU cores and more than a thousand GPU cores. The combined computing power may easily achieve a few terra flops in a single computing node. In addition, cluster computing resources, including both institutional clusters for grid computing and commercially available cloud computing resources are readily available to researchers at affordable costs. As such, it makes sense to evolve GVE systems from desktop based computing that mostly can only utilize a single CPU core to the broad spectrum of parallel computing at the server side and boos the performance of Web-based GVE systems. We refer to [10] for a review and comparison of different categories of parallel computing in the context of geospatial applications. We note that although there are a few existing systems (e.g., VisIt) typically running in high-performance computer centers, that allow offline graphics rendering for visualization purposes, to the best of our knowledge, research on utilizing parallel computing power on commodity hardware that are widely available to the majority of researchers for GVE applications is pretty much still an uncharted area which requires multidisciplinary efforts. Towards this end, by leveraging our previous works, we have designed a new GVE system to visually explore the relationships among the taxonomic-geographical-environmental components of global biodiversity data on commodity workstations equipped with multicore CPUs and high-end GPUs. By designing new parallel GVE techniques, including some new GPU based indexing and query processing techniques, we expect the new GVE system is capable of answering a variety types of queries in real time to facilitate the visual exploration process. The GVE system can be easily duplicated and federated to support a large-number of users and the visual exploration logs can be used to discover the patterns of visual exploration processes. The findings and conclusions of the visual exploration processes by domain scientists as well as the general public can be subsequently used to seek community consensuses and further test unusual (potentially novel) discoveries.

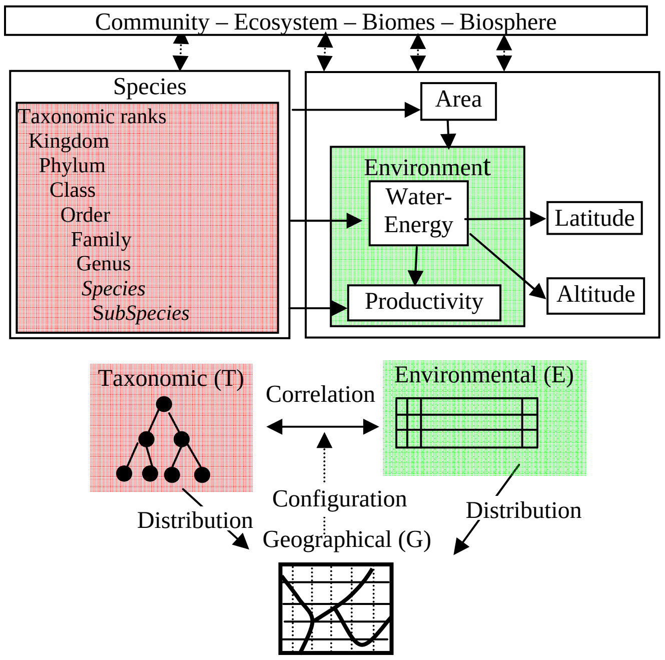

Fig. 1 Conceptual Framework of Large-Scale Biodiversity Data

3.2 High-Level Designs

Fig. 1 illustrates the conceptual framework of large-scale biodiversity data which has been discussed previously from different aspects [3, 4, 8, 10]. We categorize the relevant data into three categories: taxonomic, geographical and environmental. Taxonomic data are the classifications of organisms (e.g., Family, Genus and Species) and the environmental data are the measurements of environmental variables (e.g., precipitation and temperature) on the Earth. The geographical data defines the spatial tessellation of how the taxonomic data and the environmental data are observed/measured, which can be based on either the vector polygonal or the raster grid tessellation. The potential research in exploring species distributions and their relationships with the environment is virtually countless given the possible combinations of geographic/ecological regions, species groups and environmental variables. Fig. 2 illustrates the four identified research tasks that are relevant to GVE in this system design. The four tasks are selected in such a way that they can be used to exemplify how GVE techniques can be used to help realizing both the classic Information Seeking Mantra – "Overview First, filter and zoom and details on demand" [5] and more recent Visual Analytics extension - "Analyze First-Show the important-Zoom Filter and Analyze Further-Details on Demand" [11]. Due to space limit, we will skip the details of the designs on T2 T3 and focus on T1 (efficient GPU-based data structures and algorithms for indexing and query processing) and T4 (case studies using the integrated GVE environment). We note that the massively data parallel architecture of GPUs [12] are more naturally suitable for computation-intensive tasks such as clustering and permutation tests.

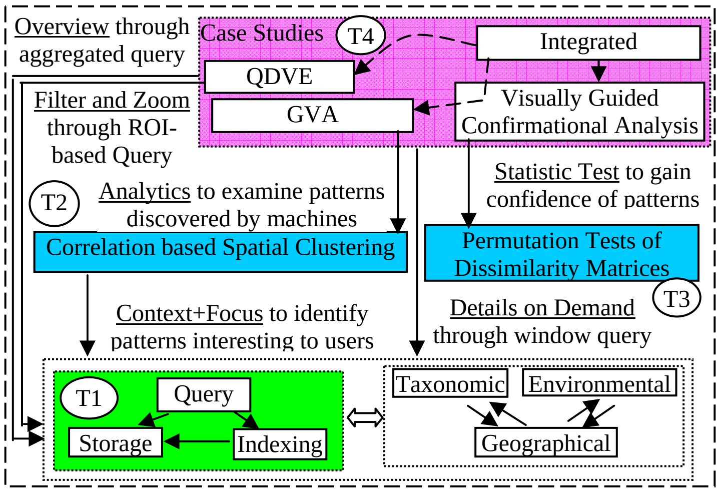
Overview through aggregated query
Case Studies
T4
Integrated
QDVE
Filter and Zoom through ROI-based Query
GVA
Visually Guided Confirmational Analysis
Statistic Test to gain confidence of patterns
T2
Analytics to examine patterns discovered by machines
Correlation based Spatial Clustering
Permutation Tests of Dissimilarity Matrices
T3
Context+Focus to identify patterns interesting to users
Details on Demand through window query
T1
Query
Storage
Indexing
Taxonomic
Environmental
Geographical

Fig. 2 System Architecture

In parallel with developing indexing and query processing techniques on CPUs using both open source PostgreSQL database [8] and in-house developed main-memory databases [10], we have also developed GPU-based indexing techniques for large-scale raster data which has achieved dozens of speedups when compared with serial CPU implementation [13]. Indexing the biodiversity data is much more challenging than indexing raster data alone. However, we observe that the main-memory quadtree data structure that we have developed for CPU-based indexing of species distribution data can be ported to GPUs by adding a data array to store species identifiers separately and adding a position field in the respective quadtree node (as shown in the right part of Fig. 3) so that these species identifies can be processed in parallel after calculating the proper positions (also in parallel). The construction of the species identifier enhanced quadtree (termed as SpIDQ-Tree) can re-use the algorithm that we have designed for constructing BMMQ-Trees on GPUs [13]. We further note that the BMMQ-Tree can be used to index the environmental data directly. By coordinating the raster tessellation of both SpIDQ-Trees and BMMQ-Trees, many queries can be performed by synchronized traversal of such trees in parallel. We note that while classical quadtrees, such as the ones we have used in our previous work [9], suffer from the signficant overheads when loading the quadtrees from disks (represented as linear quadtree paths) to CPU memory. An additional drawback of classic main-memory based quadtrees is significant cache missies because quadtree nodes are dynamically constructed and may point to disparate memory locations. In contrast, both SpIDQ-Trees and BMMQ-Trees are stored as linear arrays and are much faster to serialize and deserialize among disks, CPU main memories and GPU device/global memories as shown in Fig. 3.

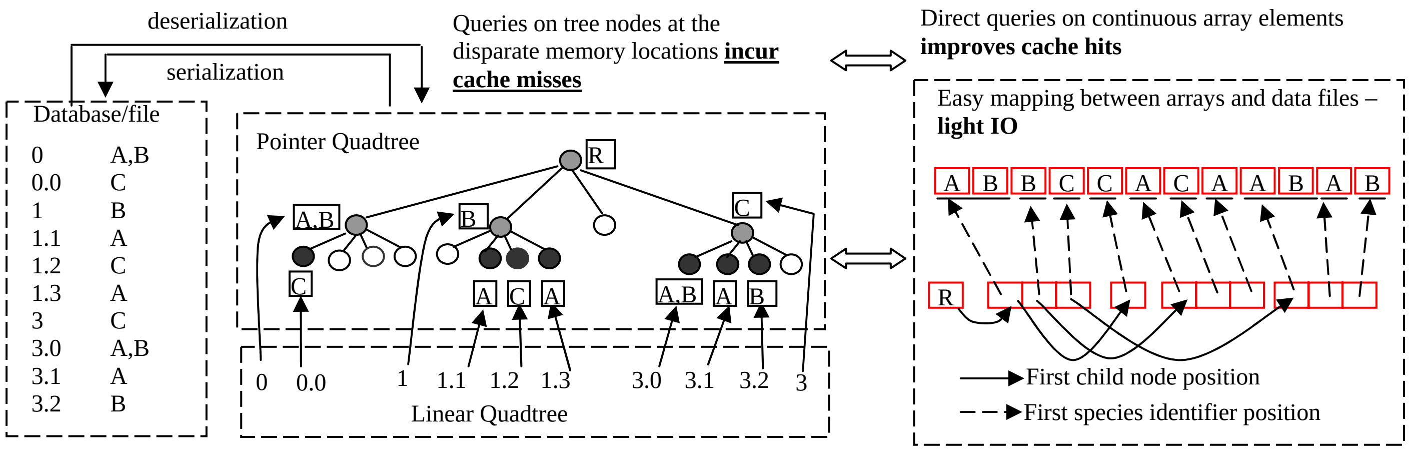
deserialization
serialization
Queries on tree nodes at the disparate memory locations **incur cache misses**
Direct queries on continuous array elements **improves cache hits**
Database/file
0 A,B
0.0 C
1 B
1.1 A
1.2 C
1.3 A
3 C
3.0 A,B
3.1 A
3.2 B
Pointer Quadtree
R
A,B
B
C
C
A
C
A
A,B
A
B
0 0.0 1 1.1 1.2 1.3 3.0 3.1 3.2 3
Linear Quadtree
Easy mapping between arrays and data files – **light IO**
A B B C C A C A A B A B
R
First child node position
First species identifier position

Fig. 3 SpIDQ-Tree: Adapting BMMQ-Trees for Indexing Species Identifiers using a Quadtree Framework

For system realizations, the popular N-tier architecture will be used to achieve high interoperability by adopting open standards such as OGC WMS and WFS. The visual exploration client connects with the server backend dynamically to retrieve relevant information. The servers work with the client to decide whether client side vector graphics rendering or server side image rending is more suitable. We plan to adopt a case study approach to demonstrate the efficiency of the GPU based indexing and query processing approaches and the effectiveness of the visual exploration system. The WorldClim global 30 arc-seconds bioclimate datasets[v] will be used as environmental data. The West hemisphere bird distribution data as well as the global mammal and reptile range maps[vi] will be used to study the characteristics of species distribution data in order to simulate the range maps of species at the millions order before the efficiency of the query processing backend and the visual exploration client can be tested. We also plan to obtain large-scale point species occurrences data by pulling GBIF data portal[vii]. The GVE system supports identifying arbitrary regions of interests for subsequent explorations. We expect the response time of any instances of the G->T, G->E, T->G+E, E->G+T and T+E->G operations to be at the sub-second level due to the performance boosting techniques.

4 Summary and Future works

In this paper, we have reported our experiences and lessons on developing high-performance Web-based GVE systems which can improve the performance of both advanced geospatial visualization and geospatial visual analytics. By examining three individual systems that we have developed over the past few years, we have recommended four techniques to boost the performance of Web based GVE systems. We further provide the high-level designs of a new GVE system to explore the complex relationships among the three components, namely geographical, environmental and taxonomic, of global biodiversity data with higher level of performance by utilizing massively data parallel computing power of GPUs that are already available on commodity personal computers.

For future work, the primary task is to realize the design of the SpIDQ-Tree and the synchronized tree traversal algorithms for various types of queries that are needed by visual explorations. Second, while the current designs on system realizations focus on using GPUs, it is more interesting (and practically valuable) to incorporate the computing power of multicore CPUs to handle control-flow intensive queries and further boost the performance of visual exploration tasks. Finally, the success of such a GVE system can only be measured by the effectiveness of facilitating scientific discovery. As such, the high performance GVE system should be user friendly and highly available to domain scientists and support collective and collaborative explorations. There are considerable system engineering issues need to solve before the system can be developed and deployed. These are left for future works.

References

[1] J. Zhang and L. Gruenwald: Opening the Black Box of Feature Extraction: Incorporating Visualization into High-Dimensional Data Mining Processes. ICDM 2006: 1188-1192

[2] J. Zhang, L. Gruenwald and M. Gertz: VDM-RS: A visual data mining system for exploring and classifying remotely sensed images. Computers & Geosciences 35(9): 1827-1836 (2009)

[3] J. Zhang, D. Pennington and X. Liu: GBD-Explorer: Extending open source java GIS for exploring ecoregion-based biodiversity data. Ecological Informatics 2(2): 94-102 (2007)

[4] J. Zhang, L. Gruenwald: Embedding and extending GIS for exploratory analysis of large-scale species distribution data. GIS 2008: 28

[5] B. Shneiderman: The eyes have it: A task by data type taxonomy for information visualizations. In: Proceedings of the 1996 IEEE Symposium on Visual Languages, 336-343

[6] J. Zhang and S. You: Supporting Web-Based Visual Exploration of Large-Scale Raster Geospatial Data Using Binned Min-Max Quadtree. SSDBM 2010: 379-396

[7] J. Zhang and S. You: Dynamic tiled map services: supporting query-based visualization of large-scale raster geospatial data. COM.Geo 2010

[8] J. Zhang, M. Gertz and L. Gruenwald: Efficiently managing large-scale raster species distribution data in PostgreSQL. GIS 2009: 316-325

[9] J. Zhang: A high-performance web-based information system for publishing large-scale species range maps in support of biodiversity studies. Ecological Informatics 8: 68-77 (2012)

[10] J. Zhang. Towards Personal High-Performance Geospatial Computing (HPC-G): Perspectives and a Case Study. Proceedings of the ACM SIGSPATIAL Workshop HPDGIS 2010.

[11] D.A. Keim, F. Mansmann,. et al: Challenges in visual data analysis. In: Proceedings of 2006 the IEEE conference on Information Visualization, 9-16

[12] D. B. Kirk and W. Hwu. Programming Massively Parallel Processors: A Hands-on Approach. Morgan Kaufmann, 2010.

[13] J. Zhang, S. You and L. Gruenwald: Indexing large-scale raster geospatial data using massively parallel GPGPU computing. GIS 2010: 450-453

[i] Source code releases available online at http://www-cs.ccny.cuny.edu/~jzhang/LabSoftDemos.htm
[ii] http://www-cs.ccny.cuny.edu/~jzhang/NYCCrashMap.htm
[iii] http://geoteci.engr.ccny.cuny.edu/rasterexplorer/comgeotiling/TestOverlay.html
[iv] http://geoteci.engr.ccny.cuny.edu/geoteci/SPTestMap.html
[v] WorldClim Climate data. http://www.worldclim.org/
[vi] NatureServe Animal Data. http://www.natureserve.org/getData/animalData.jsp
[vii] Global Biodiversity Information Facility (GBIF). http://data.gbif.org