HERCULES: High Performance Computing Primitives for Visual Explorations of Large-Scale Biodiversity Data

By Jianting Zhang @CCNY

Project Summary

Identifying solutions and responses to global environmental change is one of the key challenges of the 21st century. Research on ecosystem and species responses to climate changes and human impacts is vital to identify science-based solutions to ecological problems. Large-scale biodiversity data plays a key role in understanding the relationships among species distributions and the environment. The last decade has seen massive developments of biodiversity related information systems on the Internet and the available species distribution data has been increased dramatically with respect to spatial resolutions and taxonomic groups. The computing power offered by modern multi-core processors, accelerators and grid/cloud computing facilities also increases dramatically. While their combinations can potentially leverage our understanding of global and regional biodiversity patterns to a new level, existing software tools are mostly based on serial computing and suffer from scalability problems. Lacking support for visual explorations seriously affect their effectiveness in understanding and analyzing complex biodiversity patterns. The proposed HERCULES project (High Performance Computing Primitives for Visual Explorations of Large-Scale Biodiversity Data) aims at developing high-performance computing primitives for large-scale biodiversity data that scale up along geographical dimension (spatial resolutions), taxonomic dimension (species and species groups) and ecosystem dimension and support interactive visual explorations of global and regional biodiversity patterns.

Intellectual Merits: This project develops novel data structures, algorithms and information systems to address the computing challenges of managing and analyzing large-scale biodiversity data by exploiting massively parallel General Purpose Graphics Processing Unit (GPGPU) computing technologies. The project will (1) develop a suit of memory efficient data structures and novel GPU-based parallel algorithms for storing, indexing and querying species distribution data and the associated environmental data that scale up to millions of species at the global sub-kilometer resolution. (2) transform traditional offline model-driven statistical analysis into online query-driven exploratory analysis to facilitate interactive visual explorations by fully utilizing GPU computing power. (3) integrate a visual exploration system with the high performance computing primitives and test the effectiveness of the framework through three types of case studies: visual data explorations to identify arbitrary regions of interests through efficient query processing, visual analytics to identify and explore spatially constrained clusters, and, visually guided confirmational analysis based on non-parametric permutation tests. The project bridges computer science (databases, data mining, information visualization and parallel computing) and environmental sciences (ecology, biogeography, conservation biology) that are traditionally disparate. While this project focuses on biodiversity data, the proposed computing primitives and the visual exploration framework are general enough to be applied to many other domains (especially environmental sciences) that involve large-scale spatial data.

Broader Impacts: The proposed visual exploration system built on top of a set of high-performance computing primitives that is capable of handling biodiversity data of millions of species at the global sub-kilometer resolution will transform the traditional ways of biodiversity pattern analysis. An integrated and high-performance system will make it much easier for researchers to expand their taxonomic, geographical and ecosystem scopes of interests and encourage synthesis. The computing primitives and the visual exploration system will be disseminated as open source software and will be freely available to the research community. A cyberinfrastructure will be developed to deliver data and analytical functions as Web services so that a variety types of clients can utilize the high performance computing primitives effortlessly, in addition to using the proposed visual exploration system directly.

1 Introduction

Quantifying species-environment relationships, i.e., analyzing how species are distributed on the Earth, has been one of the fundamental questions studied by biogeographers and ecologists for a long time [1]. Species distribution modeling and biodiversity pattern analysis have attracted considerable research interests in the context of global climate change research and biological conservation practices due to their significant societal impacts. While biodiversity data are increasingly available over the Web, most of the existing modeling and analytical approaches are still based on serial computing and are incapable of effectively handling the increasing mismatches between processors, memory and disk I/O speeds. Despite the available computing capabilities are increasing fast, without fundamental changes of the underlying data structures and algorithms of software tools, the gaps between the grand scientific investigation challenges and the utilizable computing capabilities are fast growing rather than shrinking. To effectively support visual explorations of large-scale high-resolution biodiversity data, we propose to develop and integrate high performance computing primitives for indexing, querying, analyzing and visualizing the relevant data which typically includes geo-referenced species distribution data (point occurrences or range maps) and the associated environmental data. These generic primitives can be used as the functional blocks for specific applications. We are particular interested in solutions based on a General Purpose Graphics Processing Unit (GPGPU) [2] accelerated computing architecture. GPGPU allows using graphics processing units for general purpose computing. GPU devices are becoming more powerful yet affordable due to massive demands from game and entertainment industries which have shown great potential for scientific computing [3-4]. As an example, an Nvidia Fermi-based GeForce GTX480 GPU device has 480 cores with a peak performance of 1.35T flops [5] and is now available from market for $500. It has been projected that, while CPU speed improvement will only be around 20% per year [6], GPU speed improvement will be more than 50% per year, over the next decade[7].

High-performance computing tools utilizing massively parallel GPU computing power have the potential to reach every scientist's desk and transform their research. However, parallelizing serial algorithms on GPUs to fully utilize GPU's computing capabilities is technically challenging due to the simplified feature sets on GPU devices. Significant research efforts on developing new data structures and algorithms as well as reevaluating and adapting existing CPU parallel algorithms are required. The proposed multidisciplinary research is based on quite a few research disciplines in computer science and environmental sciences, such as Databases, Data Mining, Information Visualization, Geographical Information System (GIS), Remote Sensing, Ecology, Biogeography and Conservation Biology. GPU-based high-performance computing has the great potential not only to boost the performance of existing computing tools but also will foster new applications enabled by the unprecedented computing power in a personal computing environment. For example, many computationally intensive approaches that usually run in an offline mode can now run online and support interactive explorations. The uninterrupted exploration processes are likely to facilitate novel scientific discoveries effectively.

PI Zhang was trained as both a physical geographer and a computer scientist specializing in spatial databases and GIS. Since he joined the NSF Large Information Technology Research (ITR) project Science Environment for Ecological Knowledge (SEEK) project [8-9] as a post-doc research associate in 2004, he has been continuously working on a variety aspects of managing large-scale biodiversity data. In addition to earlier work on exploring cyberinfrastructure technologies for species distribution data modeling [10-15], his recent work along the direction includes integrated modeling of taxonomic-geographical-environmental components of biodiversity data [16-17], visual exploration [18], correlating biodiversity with environmental data [19-21] and indexing, querying and visualizing large-scale raster environmental data [22-23]. Being a computer scientist working in a biology department while with the SEEK project, he was fortunate to be in a unique position to learn science drives and theories of species distribution modeling and biodiversity pattern analysis. As a new starting point of the PI's long term research goal to develop high performance computing tools and facilitate fundamental biodiversity research, this proposal introduces massively parallel GPU computing and develops novel data structures, parallel algorithms and visual exploration case studies to enable new scientific discoveries. In a way similar to the transformative role of bioinformatics computing tools to molecular biology and biomedical research at the micro-scale, the PI firmly believes that **high-performance computing tools that can efficiently process and visually explore large-scale high-resolution biodiversity data at the macro-**

scale will be transformative to facilitate answering fundamental ecological and environmental questions.

2 Background and Overview of the Proposed Research

The proposed research is directly related to GPU computing and biodiversity data. We will provide a general introduction to the background of the two areas in this section before presenting an overview of the proposed research. A Graphics Processing Unit (GPU) is a hardware device that is originally designed to work with CPU to accelerate rendering of 3D or 2D graphics. The concept of General Purpose GPU (GPGPU) turns the massive floating-point computing power of a modern graphics accelerator's graphics-specific pipeline into general-purpose computing power [2]. GPGPU computing technologies provide a cost effective alternative to cluster computing and have gained considerable interests in many scientific research areas in the past few years [3-4]. According to the Nvidia website, when compared to the latest quad-core CPU, Tesla 20-series GPU computing processors deliver equivalent performance at 1/20th of power consumption and 1/10th of cost [24]. A GPGPU device can be viewed as a parallel Single Instruction Multiple Data (SIMD) machine [25]. Although we will be mainly focusing on Nvidia's Compute Unified Device Architecture (CUDA) [26] enabled GPUs in this project due to its popularity in scientific computing, we argue that the proposed data structures and algorithms can be adapted to other types of GPUs, such as AMD/ATI Stream programming enabled ones [27]. A CUDA-enabled GPU device is organized into a set of Stream Multiprocessors (SMs). Each SM has a certain number of computing cores and is able to launch a large number of threads simultaneously. All the cores in a SM share a certain amount of fast memory called shared memory and all the SMs have accesses to a large pool of global memory on the device. Since each SM has limited hardware resources, such as registers, threads, shared memories and scheduling slots, a SM can accommodate only a certain number of computing blocks subjected to the combination of the constraints.

Several enabling technologies have made biodiversity data available at much finer scales in the past decade, including DNA barcoding for species identification [28-32], geo-referring for converting descriptive museum records to geographical coordinates [33-35], database technologies for managing species presence locations and related taxonomic and environmental data [36-44] and GIS for species distribution data modeling and analysis [45-92]. The newly emerging cyberinfrastructure technologies (e.g., metadata, ontology, Web services and scientific workflow) have made exchanging and sharing species distribution data over the Web much easier [8, 9, 93-139]. On the environmental data side, advancements in remote sensing technology and instrumentation have generated huge amounts of remotely sensed imagery [140]. In addition to raw imagery data, derived data products targeting at domain-specific applications are fast growing as well [141]. Still yet, numerous environmental models, such as Weather Research and Forecast (WRF[142]), 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 modeling species distributions and analyzing biodiversity patterns.

Fig.1. depicts our understandings of the relevant components in exploring species-environment relationships and biodiversity patterns. 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 and we refer to [143-145] for reviews. While research on specific combinations is certainly valuable to enrich our understanding of biodiversity patterns, a scalable system that allows research on arbitrary combinations easily without going through complex and tedious data preprocessing is desirable. Such a system also reduces biases on underrepresented species groups and ecosystems and encourages synthesis.

The complexity of the problem domain and the rich and diversity of the research publications make it very difficult to precisely categorize exiting research, identify their computing needs and address high-performance computing challenges. However, we consider three categories among many are most prominent. The first category is predictive modeling of single species distribution by associating the presence and absence of a spices at a collection of point locations with the values of multiple

environmental variables [143,146-158]. Many models are based on statistical regression or machine learning algorithms [159-167] and some of them have been implemented in open source software packages, e.g., OpenModeller [168], Desktop GARP [169], Diva GIS [170], ModEco [171], Maxent [172] and a few R packages e.g., adehabitat [173] and GRASP [174]. The models are valuable to global climate change research by predicting possible species distributions for changed environmental variable values under different global change scenarios. The second category is related to modeling species distributions of multiple species in communities and ecosystems. In addition to running a model in the first category multiple times for each of the species independently, alternative approaches have been proposed that can incorporate biotic interactions and other factors and produce better results [175-189]. The third category is related to research on observed species richness and their diverse relationships (such as species-area, species-water/energy, species-latitude, species-altitude and species-productivity) with the environment at the global and regional scales [190-263]. Research in the third category serves as the scientific foundations for the predictive modeling research in the first two categories. In this project we will limit our targeted application domain to correlating biodiversities and the environment (or biodiversity pattern analysis) for three reasons. First, biodiversity pattern analysis is more suitable for visual explorations to discover large-scale (global and regional) patterns which are more complex and less predictable. Second, compared with single species distribution modeling and prediction that usually has a limited number of samples, biodiversity pattern analysis can be more computationally intensive which often involves pair-wise calculations among a large number of geographical units and permutation based statistical tests. Third, while quite a few sophisticated packages are available for modeling single species distributions, computing support for biodiversity pattern analysis research is generally limited.

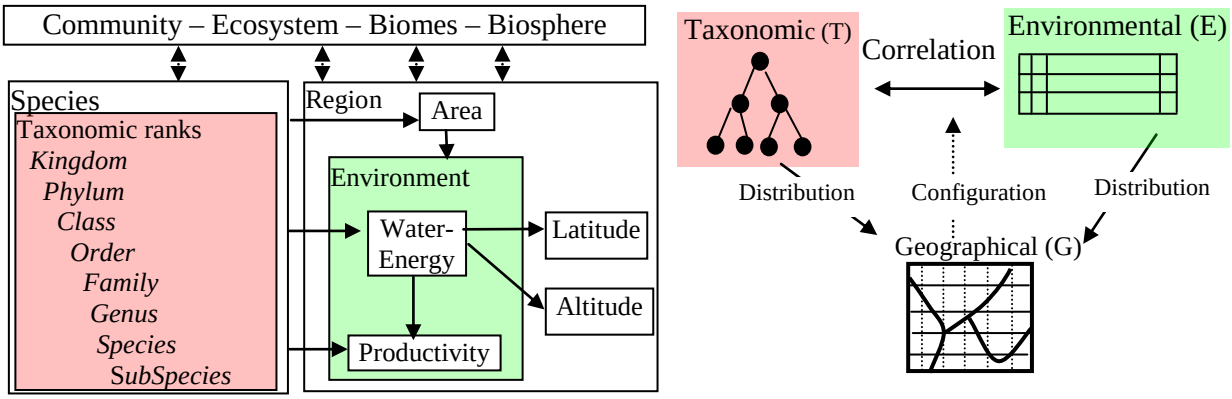

Fig. 1 Conceptual Framework of Large-Scale Biodiversity Data and Pattern Analysis

Despite the increasing availability of biodiversity data, published research works seldom utilize a spatial tessellation that is more than a few hundreds by a few hundreds raster cells. We believe that, like many other disciplines such as biomedical imaging and astronomy, high-resolution data is critical in identifying new patterns. Although coarse resolutions may be suitable for certain types of research and decision making, being able to effectively use large-scale data and perform research at the finer resolutions will also justify the suitability of using coarse resolution data. Unfortunately, existing software [264-266] may not scale-up to large-scale biodiversity pattern analysis research, especially in a visual exploration context due to lacking support for scalable storage, efficient data structures, high-performance parallel algorithms, advanced functionality for visual explorations or their combinations. A highly usable visual exploration system built on top of a set of high performance computing primitives may potentially transform the way of computing in exploring biodiversity patterns from large-scale high-resolution data. Towards this end, the following three research aims have been identified:

(1) Research Aim 1 (R1): Develop a suit of novel **memory efficient data structures and parallel algorithms** for storing, indexing and querying biodiversity data that scale up to millions of species at the global sub-kilometer resolution. We aim at 2-5X speedup for query processing and 5-20X speedup for indexing biodiversity data.

(2) Research Aim 2 (R2): Transform traditional offline model-driven statistical analysis into **online query-driven exploratory data analysis** to facilitate interactive visual explorations by exploiting **massively parallel GPU computing**. We aim at 20-100X speedup for correlation based spatial clustering and Mantel permutation tests.

(3) Research Aim 3 (R3): Design **problem-oriented visual explorations techniques** on top of the backend high-performance computing primitives through **case studies**.

To realize the three research aims, we have selected four research tasks relevant to visual explorations of large-scale biodiversity data (1) indexing and querying large-scale species distribution data and environmental data (2) generating local correlation indices between species occurrence data and environmental data and performing spatial clustering based on the correlations (3) performing permutation tests on dissimilarity matrices based on beta diversity measurements of species occurrence vectors and environmental feature vectors (4) developing an integrated visual exploration environment and performing cases studies that include visual data exploration, visual analytics and visually guided confirmational analysis.

While the details of the individual tasks (including background, related works and proposed research activities) will be presented next in the corresponding sections, we note that the four research tasks are closely related to databases, data mining, statistical analysis and information visualization, respectively. The four tasks are selected in such a way that they can be used to exemplify how major visual exploration schemes can be successfully applied to process large-scale biodiversity data using the proposed high-performance computing primitives. From Fig. 2 we can see that both the classic **Information Seeking** Mantra – "Overview First, filter and zoom and details on demand" [267] and its **Visual Analytics** extension - "Analyze First-Show the important-Zoom Filter and Analyze Further-Details on Demand" [268] can be realized. We next present our proposed research for each of the four tasks, each with one or more research activities. The details on case studies and performance evaluation plan are presented in Section 6.2 after introducing the research on the computing primitives in respective sections.

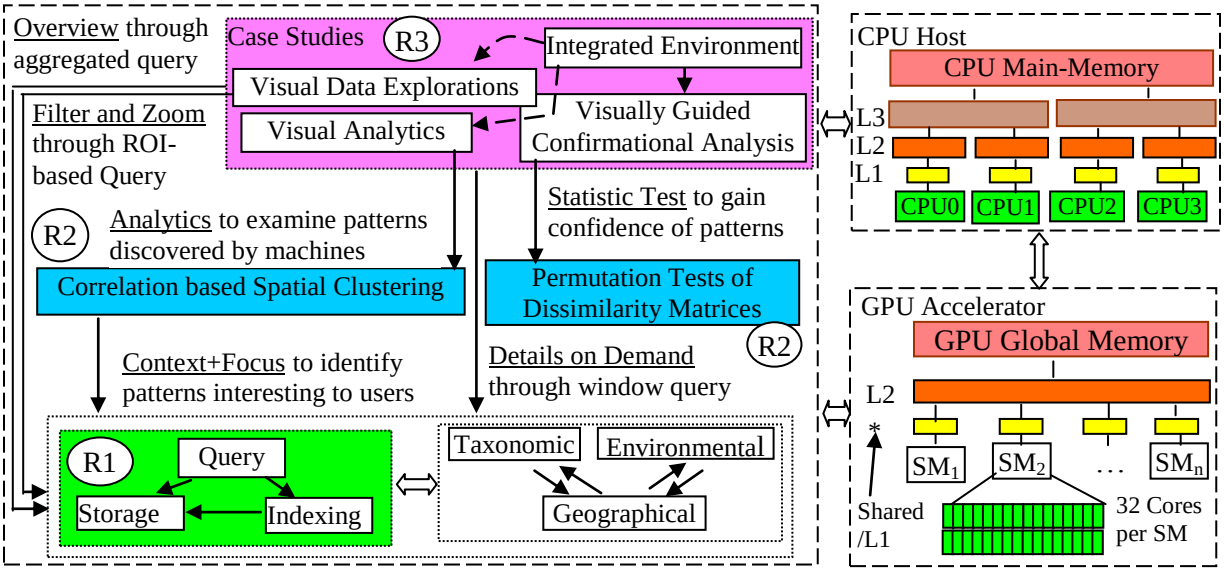

Fig. 2 Overview of Research Tasks on Computing Primitives and Visual Explorations

3 Task 1: Efficient Indexing and Query Processing

3.1 Background and related work

Biodiversity data can be classified into three categories: taxonomic, geographical and environmental. For taxonomic data, the most popular species classification system currently used by taxonomists is called the Linnaean system [269] that breaks down organisms into seven major divisions (or taxa). There are a few repositories [270-272] that provide services to find the taxonomic hierarchy based on the common name or scientific name of a species. It is generally believed that the number of species is at the level of millions. Although current generation desktop computers can easily handle millions of data items, the problem space gets much larger when taxonomic data are coupled with geographical and environmental data which makes high-performance indexing and query processing essential. Taxonomic data alone does not tell the geographical distributions of species. More and more geographical data of the species occurrences are now becoming available [273-277]. The huge data

volumes have imposed significant challenges on querying and analyzing the species distribution data. Researchers at the GBIF have performed experiments on using cloud computing technologies to parallelize their algorithms which showed great potentials of high performance computing for processing species distribution data [278]. Mapping between taxa (species and species groups) and their geographical distributions is a basic function for exploring biodiversity patterns. As species range maps are often represented as individual layers, it is very difficult to efficiently query all the species whose distributions intersect with an arbitrary spatial query window. This is largely due to the fundamental limitations of layer based vector data model in supporting co-location type query as discussed by Samet in [279]. A few Web portals [280-283] have been developed to allow visualizing and querying individual species and their point occurrence locations. However, many of these systems suffer from long response times when the numbers of occurrence records and query window sizes become large due to lack of advanced indexing and query processing techniques. The GBIF data portal [277] has utilized a pre-computing approach to improve query performance by allowing query window sizes of multiplications of 1 by 1 degree only. While the technique may work well for Web-based overview purposes, it is not sufficient for precise queries that involve arbitrary sizes of query windows. Furthermore, it is desirable to develop data structures and algorithms that can scale up to all known species in the order of millions. Numerous environmental data are now available through in-situ observations, remote sensing, sensor networks and model simulations. We refer to the National Ecological Observation Network (NEON) design documents [284] for categories of environmental data that are relevant to ecological studies. While traditionally most environmental data come from ground observations, satellite derived products are increasingly used in the analysis due to their broad spatial coverage and continuous temporal coverage [285]. Indexing and query processing of point and polygonal geographical data are well established research topics in spatial databases and GIS (for surveys, see [286-287]). Although point data can be efficiently indexed and queried in many spatial databases, unfortunately, research in indexing and querying large-scale raster environmental data remains limited other than image pyramid and tiling for simple display purposes. As discussed in [288], research on parallel processing of geospatial data prior to 2003 has very little impact on mainstream geospatial data processing applications, possibly due to the accessibility of hardware and infrastructures in the past. A few recent works on parallel processing of geospatial data on cluster computers have been reported [289-291]. GPGPU technologies have been applied to relational databases. A set of GPGPU primitives to support relational operators have been developed on top of classic parallel algorithms including sorting and scan [292]. Bakkum and Skadron [293] have implemented a subset of the SQLite command processor directly on GPU. More recently, GPU has been used to batch processing a large number of simultaneous queries on tree structures [294]. GPU has also been used to speed up similarity joins on point data [295].

In our ACM-GIS'08 paper [17], we have discussed the disadvantages of using classic layer-based GIS data model to manage large-scale species distribution data. Identifying species distributed in a query window requires cross-layer queries that cannot be efficiently supported by the current data models and systems. A simple integrated geographical-taxonomic-environmental data model was proposed by extending our previous work on integrating geographical and taxonomic data for exploratory analysis [16] and extending the classic GIS object-relational data model as discussed in [17]. The integrated data model was realized in a main-memory based visual exploratory analysis system. Due to the scalability problem of the main-memory based system, we have developed an approach to building a specialized quadtree by rasterizing all species distribution polygons and associating species identifies with different levels of the quadtree nodes [296]. Using the classic linear quadtree techniques, the quadtree nodes can be stored as tree paths and the associated species identifiers can be stored as arrays in the PostgreSQL database. The tree paths can then be indexed by PostgreSQL to speed up query processing using a query transformation technique as reported in [18]. Both the average and maximum query response times are in the order of a few seconds for query window up to 10 by 10 degrees for the NatureServe West Hemisphere bird data [273]. It is desirable to further improve query performance and reduce response times to sub-second level for lager numbers of species at the finer spatial scale. On the indexing and query processing of raster environmental data side, in our SSDBM'10 paper [22], we have proposed a Binned Min-Max Quadtree (BMMQ-Tree) data structure to speed up Region-of-Interests (ROI) type of queries, e.g., find all the quadrants whose precipitation is between [P1,P2). Using the global 30-arc seconds (approximately 1km) January precipitation data from the WorldClim dataset [297], experiments have shown that the maximum query response time for two value ranges and eight selected query windows is about a quarter of a second based on a 32-bin quantization. As the experiments in [22]

showed that rendering of the resulting quadrants was a bottleneck for Web-based visual explorations, we have adopted a server-side rendering approach to efficiently generate image tiles of query results by utilizing the tree indices to speed up client side rendering. As reported in our Com.Geo'10 paper [23], experiments results have shown that the prototype system achieves an end-to-end performance in the order of sub-second for 1024*1024 pixels display area using 16 tiles, a dramatic improvement over the results reported in [22]. To speed up BMMQ-Tree construction, we have designed a parallel tree construction algorithm that can run on Nvidia CUDA-enabled GPU devices. Our preliminary experiments show that an Nvidia Quadro FX3700 GPU device with 112 cores can index a 4096*4096 single-variable raster in less than half a second [298]. The results are quite encouraging in the sense that real-time indexing of global 1km raster datasets is becoming a reality on GPUs.

3.2 Proposed research activities

Our goal is to develop a suit of novel memory efficient data structures and parallel algorithms for storing, indexing and querying biodiversity data that scale up to millions of species at the global sub-kilometer resolution which translate to a raster tessellation of 43200*21600 cells. Due to the pruning power of tree-based indices, it is likely that tree indexing will continuously be utilized for querying large-scale datasets. While parallel computing is a natural choice to process large-scale data, constructing tree indices requires global knowledge of all the data items and its parallelization is more difficult than applications that process data items locally and independently. In addition, visual explorations have a much stricter requirement on query responses (usually below a few seconds). These technical challenges necessitate re-examining the suitability of existing data structures and algorithms, identifying suitable decomposition schemes for effective parallelization and assembling available high-performance computing techniques to achieve the desired goal.

Activity 1: Cache-conscious and memory efficient data structures. We propose to use the Cache Conscious Quadtree (CCQ-Tree) that we have developed for raster geospatial data [298] and extend CCQ-Tree for species distribution data. CCQ-Tree uses an array representation to store quadtree nodes in a breadth-first traversal order. A CCQ-Tree node has a data field and a position field. The position field is used to store the position of a node's first child node on the array. We propose to reuse the rasterization codebase that we have developed [296] to build a CCQ-Tree for species distribution data that links a quadtree node with a list of species identifies [18] and we term the data structure as CCQ-Tree-SP. As the number of species identifiers may vary significantly among the quadtree nodes and it is inefficient to allocate a fixed space for the data field to store species identifiers, we propose to use a separate array to store all the species identifies (SPID_Array) based on the breadth-first traversal order while use the data field of the CCQ-Tree node to store the position of the first species identifier on SPID_Array (as shown in the right part of Fig. 3). Compared with the pointer quadtree and linear quadtree techniques (left side of Fig. 3), in a way similar to the discussions of the CCQ-Tree in [298], the benefits of CCQ-Tree-SP include (1) Memory footprint reduction by storing only the position of the first child on the node array instead of four pointers. (2) Cache conscious due to the fact that sibling nodes are often retrieved together. (3) Easy mapping between main memory arrays (quadtree node array and SPID_Array) and data files without going through an expensive pointer quadtree reconstruction process.

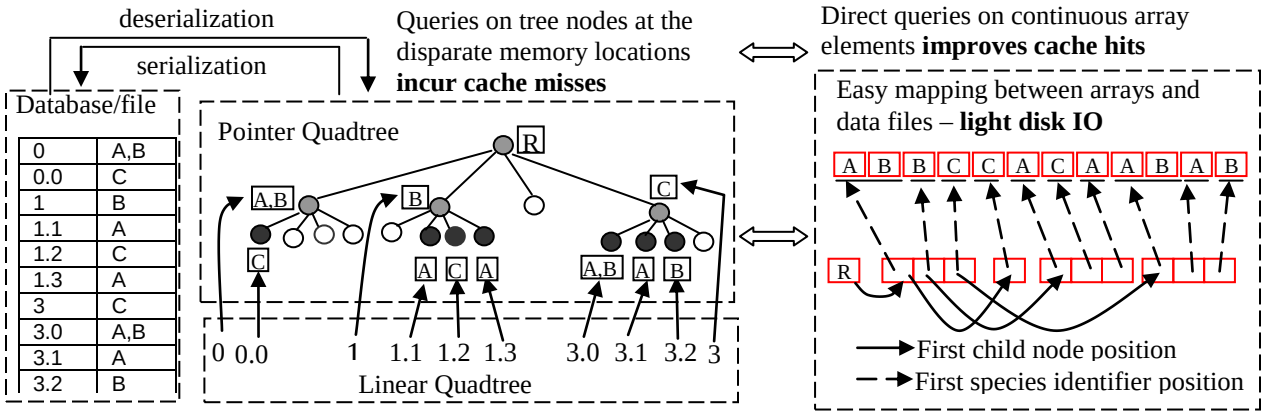

Fig. 3 CCQ-Tree-SP: Extending CCQ-Tree for Species Distribution Data

Activity 2: GPU based real time indexing of raster geospatial data. Traditionally database indexing is considered computationally expensive and requires offline processing for large datasets. We believe that the computation power provided by GPU devices with hundreds of cores may make real time indexing on certain types of data a reality based on our previous experiments [298]. The Nvidia Fermi GPU architecture has quite a few new features that are appealing to data intensive computing, such as larger shared memory, configurable (16k/48k) L1 cache per SM and unified L2 cache [5]. We propose to use GPU devices as accelerators for indexing large-scale raster environmental data and exploit the new features of the Fermi GPU architecture to further improve indexing speed so that query processing engines can utilize the constructed indices in a real time manner. We propose the following strategies to achieve the goal: (1) Minimizing unnecessary work load by utilizing data semantics. For example, many global datasets do not have valid data in oceans which cover 70% of the Earth surface and can be excluded from processing. The research challenge is to develop a light-weighted module within a data management system to manage the data semantics. (2) Utilizing the new L1/L2 caches fully to hide memory access latency as global memory access can be two orders more expensive than simple computation in tree-based indexing on GPUs. This is a largely uncharted research area and significant research efforts are needed to understand how GPU caches affect overall throughputs. (3) Pipelining CPU and GPU processing units through task and data decomposition. While space-based data decomposition for dense raster data is a plausible heuristic, we plan to investigate how spatial granularities in the decomposition affect the efficiencies of different pipelining schemes. We also plan to extend the previous work on deriving cost-models and CPU-GPU co-processing on relational data [292] to species distribution data and environmental data and investigate suitable pipelining schemes.

In addition to the two research activities, a system approach will be adopted to realize Research Aim 1, i.e., 2-5X speedup for query processing and 5-20X speedup for indexing biodiversity data, by integrating the proposed research with existing techniques. We expect the query processing speedup will be realized by the cache conscious data structures and CPU-GPU co-processing. The indexing speedup is expected to be realized by massively parallel GPU-based indexing.

4 Task 2: Correlation based Spatial Clustering

4.1 Background and related work

Clustering is one of the major data mining techniques. Spatial clustering imposes spatial constraints (e.g., adjacency) when clustering spatial data. Quite a few algorithms have been proposed [299-313], but only a few variations based on the DBSCAN algorithm are known parallelizable [314-315]. Both general clustering and spatial clustering algorithms have been applied to biodiversity pattern analysis [316-331]. In particular, an important concept called Ecoregion has been widely used in many branches of environmental science including ecology, geography and biogeography literature [275,332-344]. Ecoregions can be delineated by environmental scientists but more and more ecoregion systems are being generated by clustering algorithms. One problem with many existing approaches is that clustering is based on the distances among individual data items and they are incapable of handling clustering based on correlation that requires a set of individual data items to compute a correlation coefficient reliably. Delineating regions that biodiversity measurements are highly positively or negatively correlated with certain environmental variables is very useful for domain scientists to understand spatial distributions of the correlations in a large spatial extent (e.g. global), identify regions of interests for further investigations and subsequently seek causal relationships. We propose to integrate Geographically Weighted Regression (GWR) [345-347] with a contour tree [348-349] based regionalization approach to develop a novel correlation based spatial clustering algorithm. The GPU implementation of the algorithm will be used as a computing primitive for visual explorations as shown in Fig. 2.

GWR extends the traditional regression framework by allowing local parameters to be estimated [345-347]. Given a neighborhood (or region) definition of a data item, a traditional regression can be applied to data items that fall into the neighborhood or region. In the simplest form, when only one independent variable is involved (which is assumed in the task), each data item will obtain a correlation coefficient between the dependent and independent variables. The correlation coefficients for all the geo-referenced data items (raster cells or points) form a scalar field which has motivated us to use a terrain metaphor [350] to explore the topologies of the distributions of the correlation coefficients. GWR has been widely used in geospatial analysis to understand the local variations inherent in many geospatial and environmental phenomena [351-371], including a few studies on biodiversity pattern analysis [379-389].

GWR analysis is available in open source R system [383] and recently become available in ESRI ArcGIS 9.3.1 [384]. While the potential of using GWR for visualization has been recognized in [385-388], GWR is usually very computationally intensive and is not suitable for interactive visual explorations of large datasets. Furthermore, none of the previous works have suggested clustering the derived GWR correlation coefficients as a means of identifying overall correlation patterns in the context of visual analytics [389-390]. The work reported in [391] provides a parallel implementation in a shared-nothing grid/cluster computing environment by splitting the computing task to multiple processors with each processor handling a single data item at a time. We consider GWR is suitable for massively parallel GPU computing due to GPU floating point computing power. The high memory bandwidth on GPU devices also makes it more efficient to exchange data between processors and memory when compared with network bandwidths among cluster computing nodes. Contour tree (CT) is a tree-structured graph representing the transitions of iso-lines and iso-surfaces (appear, disappear, joint and split) in continuous scalar fields with increase or decrease of the threshold of field value (iso-value) [392]. An example of contour tree is shown in the right part of Fig 4. Contour tree has been extensively studied in computer graphics for generating and rendering iso-surfaces [349,393] and describing terrains [348, 394-396]. More recently, contour trees have been used to build user interfaces reporting the complete topological characterization of a scalar field [392, 397-398]. While a previous work has combined contour tree and level set segmentation to better segmenting images [399], contour tree techniques are largely unconnected with spatial clustering except the seminal work on linking spatially constrained clustering with upper level set scan for hotspot detection [400]. Quite a few efficient algorithms have been proposed to derive contour trees from 2D or 3D images [349,395,401-404].

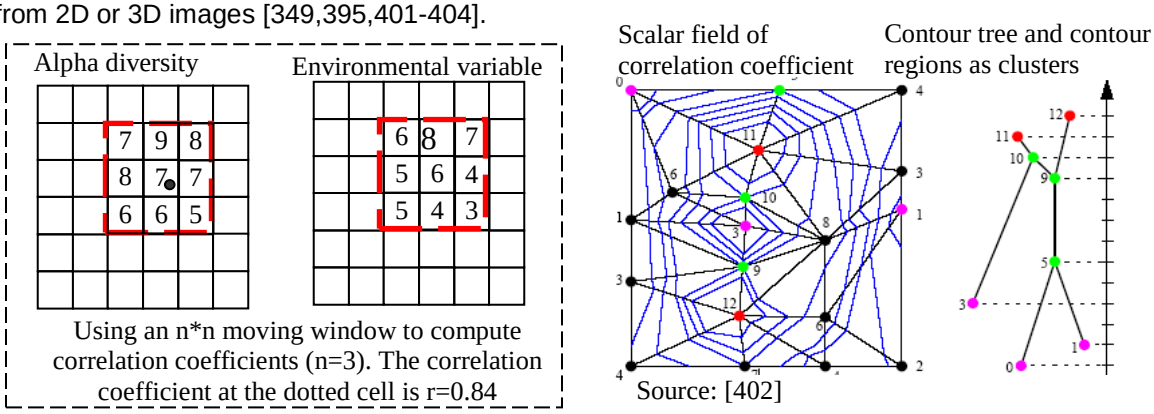

Fig. 4 Correlation based Spatial Clustering by Integrating GWR and Contour Tree Techniques

It is well known that the slowest unit determines the overall performance in parallel computing and it is best to evenly distribute computing loads to symmetric processors as much as possible. Unfortunately, real world data very often are skewed. As example, 70% of the Earth is covered by oceans and they are often excluded from terrestrial biodiversity pattern analysis. In many geospatial computing, e.g., window query and spatial interpolation, the number of data items within the neighborhood of a focal data item grows with the degree of data skewness. As such, the computing intensity can grow quadratically with the skewness in the densest area when an even space decomposition approach is adopted. Subsequently the overall parallel performance degrades quadratically with the skewness. While research on load balancing for parallel computing is an established area, very few are specifically designed for geospatial data (but see [289] for an example).

4.2 Proposed research activities

We aim at providing a new spatial clustering approach for exploring biodiversity patterns by integrating GWR and Contour Tree techniques for both raster and point data in the geographical domain. The key idea is illustrated in Fig. 4. The approach has the following steps: (1) Indexing geographical data using a proper quadtree data structure. This step may be skipped if the quadtree already exists. (2) For each data items, compute the data items that fall within a search window definition (or bandwidth according to GWR). Data skewness needs to be carefully handled for parallel computing in this step. (3) For a selected independent and dependent variable pairs, compute the desired statistic indicator (z) for each data item at (x, y). (4) Compute a contour tree by treating the set of the (x, y, z) pairs as a scalar

field and use the derived contour regions as clusters. We propose to develop a novel data decomposition approach and an efficient parallel algorithm for computing the GWR statistics.

Activity 1: Handling data skewness and data decomposition. We propose a new approach to handling the data skewness by utilizing the quadtrees that have been built for query processing. As shown in Fig. 5 (top), the number of descendents of all the quadtree nodes can be computed by post-order traversal of the tree. Assuming that each processing unit can accommodate K data items, by recursively pre-order traversal of the tree, the quadtree nodes that satisfy the following two conditions can be identified: (1) the number of descendents of its parent node is greater than K (2) the number of descendents of at least one of its child nodes is less than K. For each of the identified node during the traversal, determine an assignment scheme to divide its child nodes whose numbers of descendents are less than K into groups where each group has less than K descendents. The proposed tree partition heuristic has a low computing overhead: only one full traversal to compute number of decedents and a partial traversal to locate suitable nodes for decomposition are required. For each located node, there are only 15 possible combinations for four quadrants. Some may not be desirable and can be discarded. The low computing overhead makes it suitable for online data decomposition; however, the tradeoffs between efficiency and qualify in data decomposition need to be carefully modeled and experimented.

Activity 2: Parallel computation of GWR statistics. Assuming the search window in GWR has a size of (w,h), then each data item at location (x,y) needs to find other data items that fall within a window of (x-w,y-h,x+w,x+h) to compute a GWR coefficient. An extended tile-based approach using the data decomposition scheme discussed above is proposed to fully utilize GPU computing power. The solution is to use the data item groups identified above as the following (lower-right part of Fig. 5 – best viewed in color). For each group, its bounding box (x1, y1, x2, y2) can be easily computed. The groups whose bounding boxes intersect with rectangle (x1-w, y1-h, x2+w, y2+h) can then be retrieved by querying the quadtree. A pair list to record the intersection spatial relationship among the groups will be subsequently built. During the parallel execution to compute correlation coefficients, each group pair will be assigned to a computing block to compute partials of the statistics of all the data items in the two blocks, i.e., Σx_i, Σy_i, $\Sigma x_i y_i$, Σx_i^2 and Σy_i^2. These partial statistics can be easily combined to compute the corresponding total statistics for the relevant data items before their correlation coefficients are computed. A more detailed explanation is provided in [487].

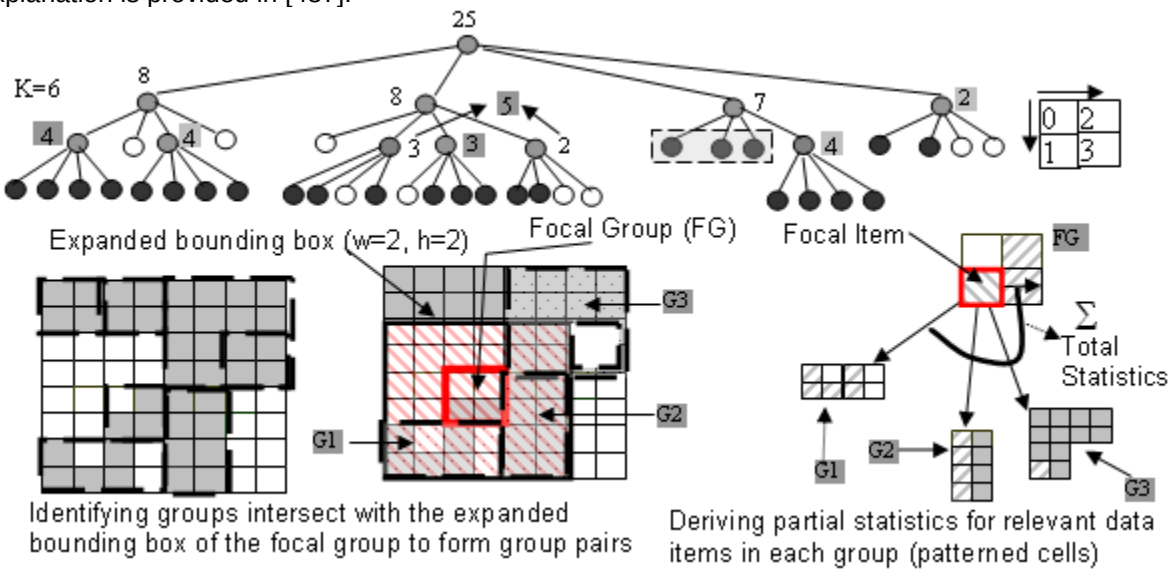

Fig. 5 Proposed Approach on Skewed Data Decomposition and Task Assignment on GPU

We consider our approach on group-based data decomposition and computation extends classic tile-based approaches that have been frequently used in many computational physics based applications as follows. Tiling is an important technique in shared memory parallel computing architecture. Tiling in CUDA-based applications is effective due to the fact that access to the per-SM shared memory is about two orders faster than accessing GPU global memory. Caching data items on the per-SM shared memory significantly reduces memory access times. Different from the classic tiling techniques that require rectangular sub-image areas, our approach can be applied to both skewed point data and raster data with

large portions of no-data which is much more flexible. The computation overhead of our approach is identifying groups and group pairs using tree indices. The storage overhead is storing group pairs.

5 Task 3: Permutation Tests of Dissimilarity Matrices

5.1 Background and related work

Permutation test is a non-parametric statistical test approach that can be dated back to the first half of the last century in the work of Fisher [405]. Compared with traditional normal-theory tests, permutation test is computationally intensive and only becomes feasible when access to computer power becomes available [406-407]. Permutation tests have been widely used in correlating biodiversity and the environment in the past few decades. Major breakthroughs [408-410] have attracted thousands of citations in a variety of research disciplines. Many of the works on biodiversity pattern analysis discussed in Section 2 rely on permutation tests on dissimilarity matrices derived from species occurrence records and environmental feature vector at a collection of geographical units (point locations, polygon centers or raster cells), respectively. Beta diversity measurements [411-418] have been extensively used to compute the difference of species composition between two geographical units while Euclidian distance is the most common way to compute the difference between two feature vectors with each environmental variable value as an element in the vectors. Several R-packages that implement permutation tests are available [264-266]. However, they are serial implementations and the number of data items that can be tested in a reasonable time frame is severely limited by the computing power of a single CPU core. When the number of data items are in the order of thousands or larger, it becomes unrealistic to use permutation tests in an interactive visual exploration setting. On the other hand, the floating point computing power provided by GPU devices is ideal for permutation tests. This motivates us to develop a GPU based high-performance permutation test approach.

A permutation test includes two steps: the first step is to randomly generate a permutation sequence and the second step is to compute statistics based on the sequence. The most popular algorithm to generate a random permutation sequence might be the Shuffle algorithm [419] among the rest (see [420] for a survey). We note that quite a few permutation tests that have been implemented as R-packages for biodiversity pattern analysis are based on the Shuffle algorithm. Both the Shuffle algorithm in the first step and computing statistics in the second step have a linear complexity with respect to the number of data items in a single permutation test run. However, unlike computing statistics in step 2 that only requires sequential read accesses to data items array, the Shuffle algorithm requires both read and write accesses to a permutation sequence and addresses memory in an unpredictable way. Since most of the commodity CPUs do not allow explicitly managing caches, the Shuffle algorithm incurs significant cache misses. Random access to memory can cost hundreds of CPU cycles and the gap is getting larger on modern CPUs. According to [421], memory access costs may account up to 80% wall clock time on an Intel Pentium processor in generating a permutation sequence. While random access to memory incurs serious performance issues on modern commodity CPUs due to their hardware controlled cache management policies, the Shuffle algorithm can be suitable for GPU implementation due to the availability of the per-SM fast shared memory on CUDA-enabled GPUs. The per-SM fast shared memory is about two orders faster than device global memory. Read and write accesses to shared memory cost only a few machine cycles. Users can explicitly program the fast shared memory based on application semantics. Data in the fast memory can be copied from and to device global memory in a coalesced way by synchronizing a large number of threads. Our proposed research on high performance permutation tests takes advantage of the commodity GPU hardware and is detailed in the next subsection. We also note that after the Parallel Random Access Machine (PRAM) abstraction for parallel computing came into being in 1970s, quite a few parallel algorithms on random permutations (e.g. [422]) have been reported based on different types of PRAM models. While these algorithms are valuable in understanding the theoretical aspects of the random permutation problems, their practical performance is largely unknown. Cong and Bader [423] compared a few parallel algorithms for random permutation generation on symmetric multiprocessors (SMPs). The recent works reported by Gustedt [421,424] tackled the problem of parallel sampling random permutations of very large numbers of data items on cluster computers. However, we are not aware of pervious work on permutation tests on GPUs, especially in the context of visual explorations of biodiversity data.

5.2 Proposed research activity

We propose to develop a high performance parallel algorithm for Mantel permutation tests [408] on CUDA-enabled GPUs. Since a significance level of 0.05 to 0.001 is often desired in practice, which translates to 20-1000 runs of permutations, we adopt a coarse-grained parallelization strategy by assigning each permutation test run to a GPU computing block. The grid of the computing blocks representing all permutation runs can be automatically scheduled by GPU hardware. While the task decomposition is straightforward, our technical contributions lie in efficiently implementing the Shuffle algorithm on a single computing block on GPU. The key idea is to utilize fast read/write accesses to shared memory and generate local permutations for chunks of data items before the local permutations are combined to generate a full permutation sequence. As shown in Fig. 6, the proposed algorithm has the following steps. First of all, an array storing the permutation sequences for all the runs (across computing blocks) is allocated on the global device memory. An array element (permutation sequence) can be uniquely addressed by the computing block identifier. Second, for a permutation sequence corresponding to a computing block, it is segmented into a number of chunks (assuming b) based on the available shared memory. Each chunk can be locally permuted in parallel by launching a number of threads proportional to the number of data items in the chuck. The chunk is then copied back to the corresponding place in the device global memory by synchronizing threads available to the computing block. The third and the final step is to generate a b*b communication matrix and exchanges data item identifies between the relevant chunks based on the method proposed in [421]. The relevant chunks will be copied back and forth between the device global memory and per-SM shared memory allocated to the computing block to modify their elements based on the communication matrix. As discussed previously, read from and write to shared memory require only a few cycles (two orders improvement) and thus memory access latency is no long a problem. We thus expect significant speedup in generating random permutation sequences.

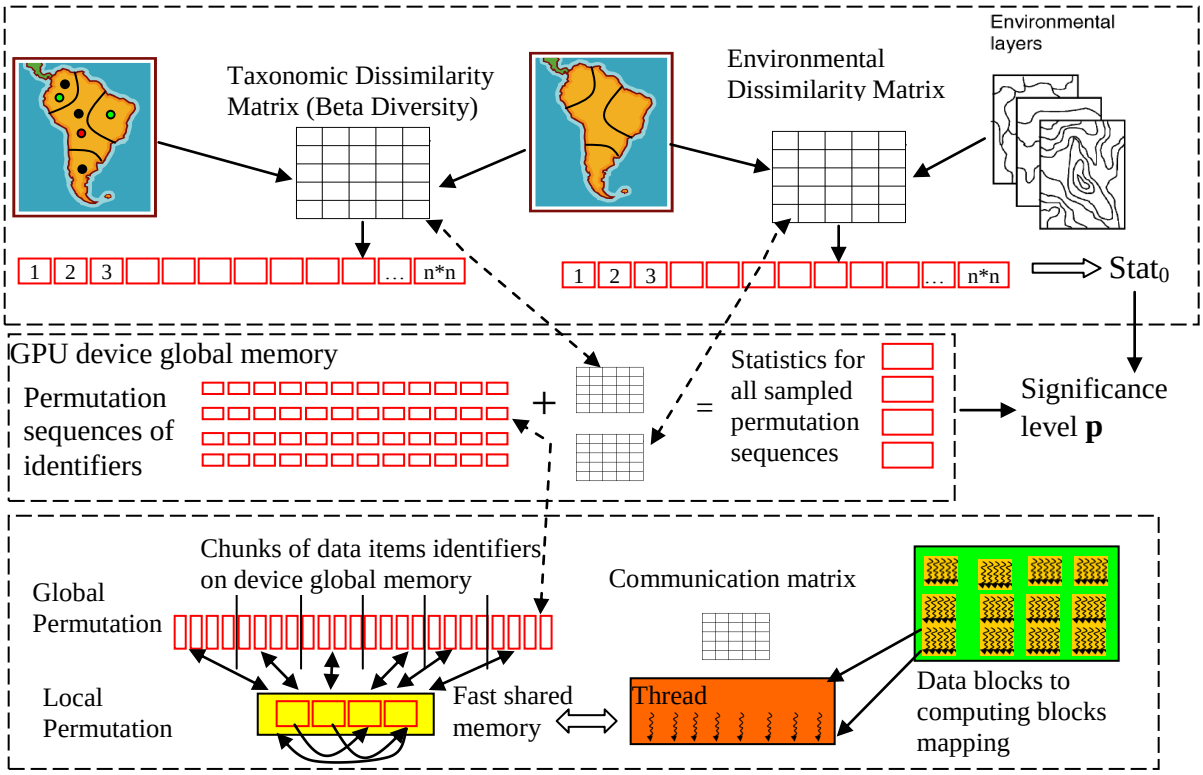

Fig. 6 Illustration of the GPU Parallel Algorithm for Permutation Test

Compared to the implementation using a cluster computer architecture that requires exchanging data item identifies among processors across networks, data exchange between GPU device global memory and shared memory can be 2-3 orders faster. Once a permutation sequence is generated, thousands of threads can be launched with each computing partial statistics of the sequence in parallel. A

scan (e.g., prefix-sum) primitive [425-426] can be applied to compute a statistic value for the sequence, in a way similar to computing statistics from groups discussed in Section 4.2 (A more detailed explanation is provided in [487]). The statistics of all permutation sequences can then be ranked on either GPU or CPU to derive a p-value to indicate statistical significance for the correlation being tested. Launching a large number of threads not only fully utilizes computing power of multiple GPU cores assigned to a computing block but also hides access latencies to GPU device global memory. Overall, we expect GPU based permutation tests to be 1-2 orders faster than CPU based ones due to the combined efficiency improvements in both steps. However, the real effectiveness can not be measured until the GPU permutation test algorithm is implemented and fully tested. If our goal of realizing permutation tests for reasonable data item numbers (e.g., in the order of a few hundreds to a few hundreds of thousands) in a few seconds can be achieved, there are significant implications which may transform research on biodiversity pattern analysis due to the advances of computing. Using the interactively selected or query derived regions of interests, users can obtain statistical significances of various correlations among the data items in the regions on the fly instead of going through complex, tedious and long-waiting processes of traditional permutation tests. The uninterrupted exploration processes are likely to facilitate novel scientific discoveries effectively.

6 Task 4: Case Studies of Visual Explorations

6.1 Background and related work

A large number of information visualization techniques have been developed and we refer to the introductory article [427] and the review papers [428-440] for further information. Visualization and visual explorations of taxonomic, geographic and environmental data individually has been extensively studied over the past few decades and significant progresses have been made [441-447]. However, to the best of our knowledge, very few of previous works integrate these three types of data seamlessly to allow visual explorations of complex biodiversity patterns. Most of the existing information visualization systems, such as DEVise [448], GeoVista[449], Polaris[450], Improvise [451-452], InfoVis Toolkit [453], Prefuse [454], GeoDa [455] and Protovis [456], assume that all data can be held in main memory so that various visualization techniques can be applied on the fly and support user interactions. While these systems provide excellent visual exploration functionality, they may suffer from various scalability problems as identified in [438]. VisIt [457] is one of the few visualization systems that supports distributed and parallel computing to speed up rendering processes. A few research on multi-level cache based [458] and GPU accelerated [459] remote rendering have been reported. However, both VisIt and the remote rendering research are mostly designed for 3D applications with spatial resolutions comparable to screen sizes, e.g., 1024*1024, while our research focuses on 2D applications that have much higher spatial resolutions, e.g., 43200*21600 for global 1km resolution data. We argue that simply tiling multiple screens to use larger display areas (e.g., [460]) may not be the best for visual exploration purposes as the vast amount of visual information may be beyond the ability of human being to perceive at the same time [438]. In contrast, our approach allows users to formulate complex queries and highlight the regions that satisfy the criteria through the high-performance query processing primitives. The context+focus and filter-zoom design (Fig. 2) matches the information visualization practices better than simply displaying large-scale data. While correlation based spatial clustering and permutation tests primitives are specific to certain types of applications in the context of visual explorations of large-scale biodiversity patterns, they can serve as the running examples to demonstrate how high-performance data mining and statistic computing primitives can be integrated into the visual exploration processes as another level of functionality on top of simple display and query-driven visual explorations [461-466]

Compared with a large body of existing research on exploratory analysis and visual analytics of geo-referenced environmental data in various domains, such as geography, air quality, medical, economics, social sciences, transportation and urban planning, very little research has applied visual explorations and visual analytics paradigms for exploratory analysis of species distributions and biodiversity patterns. We argue that an integrated visual exploration system built on top of high-performance computing primitives may potentially transform the research area due to the following reasons. First, the complexities of the Earth's biosphere and ecosystems are well recognized. Existing bioclimate envelope based species distribution prediction models have been criticized for being overly simplified [467]. Adopting a data-driven exploratory framework may provide an attractive alternative to model-driven approach in many cases which have been proven effective in quite a few other areas

[361,468]. Second, an integrated and high-performance system will make it much easier for researchers to expand their taxonomic, geographical and ecosystem scopes of interests and encourage synthesis. For example, scientists may be interested in finding the geographical and environmental correlations among multiple taxonomic groups or in a different geographical area/ecosystem that they would have not planned if such a system were not exist. Finally, the visual exploration system may potentially transform traditional meta-analysis [144-145,223,248] into on-demand re-evaluations that can provide richer information with fewer inconsistency problems.

Our GBD-Explorer prototype system [16] allows users to link species taxonomy with their geographical distributions by using an open source GIS. A few typical scenarios on exploring biodiversity patterns using WWF WildFinder [274] and Ecoregion [275] datasets were discussed. A prototype called LEEASP (Linked Environment for Exploratory Analysis of Large-Scale Species Distribution Data) [17] was subsequently developed to support environment data. In addition to visualizing individual dataset in geographical, taxonomic, ecoregion and environmental data views, LEEASP provides a set of coordination operations to facilitate exploring the relationships among the data in the four views. LEEASP were tested using 679 tree species in North America whose range maps were provided by USGS Little dataset [469]. Similar to GBD-Explorer, LEEASP is also main-memory based which limits both the geographical data and environmental data to 0.5 by 0.5 degree in order to achieve reasonable performance on a typical desktop computer. While both GBD-Explorer and LEEASP have received positive feedbacks from ecologists and conservation biologists, neither the design nor the implementation can be scaled up to millions of species at the global sub-kilometer resolution that we are aiming at. We have also developed a visual data mining system for remote sensing data [470] following a similar Coordinated Multiple View (CMV) scheme where different data views are linked with a decision tree classifier view to help users better understand both data and the data mining algorithm. The technique can be extended to link correlation based spatial clustering results with different data views in a way similar to [471].

6.2 Proposed research activities

A scalable visual exploration system is proposed by integrating our previous works on exploratory species distribution data analysis [16-17] and the high-performance computing primitives proposed in the previous sections based on the framework presented in Fig. 2. The utility of the visual exploration system is demonstrated through case studies.

Activity 1: Scalable system development. The system has an N-tier architecture to achieve high interoperability by adopting open standards, e.g., ITIS [271] and OGC WMS [472] and WFS [473] standards. The visual exploration client connects with the server backend dynamically to retrieve relevant information by invoking proper computing primitives. The servers work with the client to decide whether client side vector graphics rendering or server side image rending is more suitable. For large species distribution data and raster environmental data, we consider the dynamic tiling technique that we have developed previously [23] a strong candidate. We propose to re-evaluate the five typical operations among taxonomic (T), geographical (G) and environmental (E) data that have been identified in our previous work [17] (i.e., G->T, G->E, T->G+E, E->G+T and T+E->G) and investigate how hierarchical regionalization (based on either expert knowledge or clustering algorithms) will affect the categorization of operations among the different data types. The findings will be used to guide the decisions on online/offline indexing, with/without materialization and client/server side rendering, in addition to optimizing query execution plans and develop view coordination techniques.

Activity 2: Case studies. We propose to adopt a case study framework to demonstrate the efficiency of the high-performance computing primitives and the effectiveness of the visual exploration system. The 54 WorldClim global 30 arc-seconds bioclimate datasets [295] will be used as environmental data which have been used in our previous studies [17, 22-23]. For species distribution data, we propose to first test on 4000+ birds range maps in the West Hemisphere [273] that have been extensively explored in our previous studies [18,301]. The bird data as well as the global mammal and reptile range maps [273] 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 species distribution data by pulling data from GBIF data portal [277].

The following three categories of use cases are considered: (1) Visual data explorations to identify 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 within a few seconds. (2) Visual analytics to identify and explore clusters. By specifying a set of species, a set of environmental variables, a moving window size and an optional region of interests, the scalar field of the correlation coefficients across the study area can be derived and visualized. The correlation based spatial clustering algorithm is expected to return a contour tree and the contours associated with the saddle points in the contour tree. The properties of regions corresponding to the contours can then be derived, evaluated and sorted. Besides visualizing the topography of the scalar field directly, the clustering analytical process helps users to spatially delineate ecoregions more accurately and more systematically. (3) Visually guided confirmational analysis. While scientist may visually identify interesting biodiversity patterns based on their domain knowledge and experiences as for the use cases in the first two categories, very often they may want to know whether the identified patterns are significant based on some statistic tests. The use cases in this category integrate the visual exploration system and the permutation test primitive to achieve the goal. Scientists can modify the regions or cluster boundaries and/or taxonomic groups and environmental variables that have been selected for exploration interactively and repeat the permutation test until a conclusion can be made. We note that the visual exploration system can "learn" from users by recording the relevant input data (including taxonomic groups, geographic regions, ecosystem types and environmental variables), analytical models (and their parameters) and the discovered biodiversity patterns and applying suitable machine learning algorithms to discover usage patterns.