SIZEUP: A TOOL FOR INTERACTIVE COMPARATIVE COLLECTION ANALYSIS FOR VERY LARGE SPECIES COLLECTIONS

BY

Andrew Ozor

Submitted to the graduate degree program in Computer Science
and the Graduate Faculty of the University of Kansas School of Engineering in
partial fulfillment of the requirements for the degree of Master of Science.

_____
Chairperson

Committee members * _____*

_____*

Date defended: _____

The Thesis Committee for Andrew Ozor certifies
that this is the approved Version of the following thesis:

by

Andrew Ozor

SIZEUP: A TOOL FOR INTERACTIVE COMPARATIVE COLLECTION ANALYSIS FOR VERY LARGE SPECIES COLLECTIONS

Committee:

_____
Chairperson*

_____

_____

Date Approved:_____

Chapter 1: Introduction

Biodiversity researchers who investigate the ecological, evolutionary and conservation biology surrounding plant and animal species rely on observational and specimen collection data to document the occurrence of a species at a particular place and time. For over 250 years, natural history museums and herbaria have amassed collections of specimens from biological surveys and inventories of life on earth. An estimated three billion specimens are housed and curated in museums around the world. It is from these collections that species distributions, descriptions and identifications are known. Specimen "vouchers" (also called "occurrence points") represent single or multiple collections of specimens and are the physical basis for the discovery and documentation of new species. Within the last decade, online databases consisting of these specimen occurrence points have begun to gain in popularity. The increasing use of collaborative online databases by institutions represents a potential for comparison among specimen holdings of multiple museums collections. However, despite the online availability of data from specimen vouchers of species from around the world, individual researchers currently have no easy method to aggregate massive amounts of specimen holding data in order to analyze the relevance and value of specimen collections in different museums for their research. In this study we seek to create an interactive environment, suitably intuitive, easy to use, and fast, that will enable large data sets from multiple collections to be efficiently rated and ranked according to standard criteria of relevance to a scientist and her particular research requirements.

Before the advent of distributed query technologies on the Internet in the early 1990s, paper mail, word-of-mouth, reputation of museums, or actually visiting collections to see what they had in their cabinets were the only effective ways to discover specimen research resources for the initiation of a research, thesis or dissertation project in systematics or species biogeography. With the invention of the Gopher and WAIS protocols, and soon thereafter HTTP, biological museums began to take advantage of remote access to distributed specimen database catalogs, thus transforming the specimen discovery process for biodiversity scientists looking for documented and vouchered species records. As more museums computerized their holdings and made them available on the internet via new query-response protocols (DiGIR and TAPIR), the opportunity became available to cache data from museum databases located around the world.

The Global Biodiversity Information Facility (GBIF) was initiated in 2000. GBIF is a service that provides a global distributed network of databases which contain primary biodiversity data made available by participating museums from around the world; museums voluntarily provide as much or as little primary biological data as they see fit. The 'primary biodiversity data' is data associated with specimens in biological collections, as well as documented observations of plants and animals in nature [1]. Scientists, researchers, and students use the service by querying the GBIF databases to get species occurrence data sets. GBIF was created in large part to take advantage of this burgeoning biodiversity informatics infrastructure, by creating such a global cache of museum data, thus in turn creating new

opportunities for data mining and cross-collection comparative analysis of specimen holdings.

Being able to simultaneously query multiple individual museum data providers or the GBIF cache for individual collection records was a great advance over manual methods of species voucher discovery, but long lists of specimen data are an inefficient way to combine, compare, and analyze these individual data points for research project planning and prioritization. This makes seeking out collaboration between museums a hit or miss affair because collection managers do not have readily available resources describing the strengths of another museum's species collection, hence they have no reliable way of determining whether or how a potential collaboration would benefit them. Moreover, these collection managers do not even have readily available data about the weaknesses of their own species collections; hence they do not know the areas in which they should seek collaborations in order to improve the quality of their collection. Such information would allow a collection manager to focus on important areas for future growth.

Collaboration is just one of the many applications of comparative collection analysis. The technique can also provide a way to discover specimen research resources for the initiation of a research, thesis or dissertation project. Furthermore, for museum administrators and funders, knowing the uniqueness and research strength of a collection's specimen holdings is a key factor in evaluating curatorial, staff, and building resources for biological collection repositories. Therefore a

measure of value for their respective collection data would not only help users assess the quality of their data, but also to seek out higher quality data if necessary.

Comparative collection analysis must overcome multiple problems. There is currently no formal definition for 'quality' of a museum's species collection. The quality of a museum's species collection is inherently subjective, and the traits of a valuable collection change based on the biological domain of each user. Therefore, a useful value measure must dynamically tailor how 'quality' is defined. However, there is currently no method to dynamically structure primary biological data based on common attributes. Several technical reasons can be cited. For starters, the computation required to dynamically identify and analyze common attributes among large sets of primary biological data from multiple museums is significant. Furthermore, any useful value measure must take into account the geospatial location of each specimen in a data set. Unfortunately, distance calculation among larges sets of geospatial data (sometimes involving tens of thousands of points) has traditionally been a very time consuming process because one must compare every geospatial point with all others in the data set. Finally, a useful application of comparative collection analysis must provide some sort of user interface to display the spatial distribution of specimen data and provide tools to select relevant comparison criteria. Currently there is no straightforward way to visualize and compare museum species collections with one another.

In this paper we use a quadtree in conjunction with two novel methods developed for this research project: (i) 'branch bypassing', a fast method of speeding

up quadtree traversal time over large sets of clustered spatial data; and (ii) a unique geospatial spread calculation which quickly approximates the relative spread of large sets of geospatial data. Taken together, these methods allow us to create a value measure to determine the strengths and weaknesses of a biological collection based on any number of criteria. For the purpose of museum ranking, we created a standard that consists of three key factors: geospatial spread, environmental attributes, and each collection's 'unique contribution'. The value is then visualized in a real-time interactive environment to display the rank of each collection's specimen data compared to others. Clearly representing the value of a species collection in a visual and interactive manner allows scientists to make more informed decisions, recognize the need for and seek out collaboration when needed, and assess the strengths and weakness of their own collections.

In summary the major contributions are:

- The first-of-its-kind framework to quickly analyze large amounts of geospatial data in order to rank biological collections.
- The 'branch bypassing' operation which improves traversal speed, preserves quadtree structure, and is extensible to any region based quadtree.
- A fast method of assessing geospatial spread amongst large amounts of input data, by exploiting the spatial hierarchy of a quadtree.
- An operational real-time interactive application to visualize our comparative collection analysis.

The rest of the thesis is organized as follows. Chapter 2 gives the problem formation and establishes the three value measure criteria. In Chapters 3, 4, and 5 we present a framework for handling large collections of occurrence point data. We examine the benefits of our data structure when dealing with geospatial data, discuss branch bypassing, and propose a value measure algorithm. Chapter 6 discusses the importance of user interactivity in the modeling and visualization of the rankings and briefly presents the user interface developed to address these needs. In Chapters 7 and 8 we discuss our results and state our conclusions, respectively.

1.1: Why specify a Value?

Individual museums have traditionally been stocked with specimens reflecting their local and historical research interests. With the biological surveys of the western hemisphere by European scientists, collections in Europe and the U.K. became the repositories for new world species. Similarly, extraordinarily diverse tropical habitats later became the focus of research and research collections of scientists from the U.S. and from European museums. In the 20th century, most

countries had established national biodiversity research programs to document the flora and fauna of their own lands. This historical precedent and patchwork of specimen collecting programs and diverse research interests, has led to a highly distributed global repository of species vouchers and related museum preparations. Museums throughout the world now catalog collections of biological samples, which reflect their historical collecting programs and also new collaborations with biodiverse countries.

As a consequence, the research breadth and depth of collections varies widely among museums and herbaria. Museum collection managers and researchers would benefit in many ways from knowing which collection is the strongest or most unique with respect to a certain species. (As suggested above, this is much more complex than just how many specimens they have.) With such information, they could asses the strengths and weakness of their own collection. This would aid in determining road maps for their collections goals in the future. It would also allow them to prominently display their strengths when explaining the importance of their respective collection in proposals, etc. Comparative collection analysis would also greatly aid in museum collaboration. Currently, when a researcher seeks out an institution to collaborate with, they contact various institutions in hopes of finding one that meets their criteria. However, if the researcher had easily accessible concrete evidence, they could more easily target certain institutions. Furthermore, a spatial visualization of species occurrences would not only allow researchers to see in seconds which museums have specimens in a geographical area of their interest, but

would also show the actual specimen distribution localities, densities, spread. Finally, a ranking system would also encourage more museums to make their specimen holdings data available in online databases.

Chapter 2: Previous Work

2.1: Previous efforts to rank museum species collections

To the best of our knowledge, there have been no significant efforts to rank museum biological collections. It is not for a lack of interest, as shown from user testing of our comparative collection analysis (Chapter 7). Museum collection mangers and curators highly value the ability to easily visualize specimen occurrence data and identify strong or weak species collections in their area of interest. There are however many possible reasons why no such ranking has been done. The main reason is the fact that comparisons involving collections around the world have simply not been feasible in the past because museum specimen records were stored on paper and sorted into file cabinets. There was no method to perform a large scale comparison among many species from a museum because each record had to be physically accessed. Museums around the world have only recently begun to digitize their data and make it available online. The GBIF online repository of biological collections started in 2000 and has only 233 different providers [1]. There are still many more museums that could put forward their collections and make their species collection data available online, although the process of converting hundreds of thousands of physical specimen occurrence records to a digital format is a daunting and time consuming task. If museums had more of a reason to make their data available, such as having the ability to easily assess their own collections in comparison to others,

then more museums around the world might contribute to projects like GBIF. Another reason for the lack of previous work is that aggregating a massive amount of distributed data, analyzing it quickly, and then providing the results in a user friendly manner is an imposing task. For example, analyzing large amounts of occurrence point data for such aspects as geospatial spread is traditionally a very time consuming task.

2.2: Previous CS-related work

Biological collections data are essentially registered using geospatial attributes. Given the large numbers of collections and the fact that they tend to be very large, we needed a mechanism to partition the multidimensional data space quickly and efficiently. Data Structures for multidimensional point data are of great interest in the scientific data visualizations literature. These multidimensional data structures are most often defined by hierarchical subdivisions of space. One such multidimensional data structure is the region-quadtree. A quadtree is a hierarchical data structure based on a recursive subdivision of space [15]. More specifically, a region-quadtree subdivides a data space into four equal quadrants until each quadrant contains no more points than specified by some ‘maximum capacity’. If that capacity is exceeded, the quadrant must be split into four equal sub-quadrants. This process of sub-dividing will continue until all the elements of the input data are inserted into the

tree. As will be detailed in chapter 4, we used a region-based quadtree data structure for the purpose of quickly querying and analyzing occurrence points.

There is a history of geospatial applications that make use of quadtree data structures. Rosenfeld, et al. [17] used region-quadtrees to handle cartographic data. They digitize maps and use region-quadtrees to store the images. They found that quadtrees were useful for performing the intersection and union of images and that they greatly reduced the memory required to store the images. Quadtrees have also been used in conjunction with geographic point data as described by Samet, et al. [14]. They concluded that quadtrees were a good fit for geographic point data because of the efficiency with which many types of queries could be preformed over the data.

To the best of our knowledge, quadtrees have not previously been used in conjunction with a value measure to rank point data.

Quadtrees have been proven to work well with geographic data, but a problem arises when using region quadtrees with specimen occurrence data. Specimen occurrence data is gathered from researchers who physically collect and document each specimen sighting. As a result of the way in which specimen occurrence data are collected, the distribution of data is generally clustered. Clustered data sets are a problem for quadtrees, as the worst case performance arises from very small clusters. There has been research into handling clustered data by compressing the quadtree, as shown by Samet [9]. They seek to compress images, and in turn use a method to reduce the size of a resulting quadtree. However, this method and others primarily work with image and video data and compress a quadtree by reducing the amount of

data needed to encode the image or video, resulting in approximations; a trade off normally seen when dealing with compression. While we need compression, we cannot afford to lose information about the input data or structure of the quadtree.

2.3: Summary

In summary, the research we present here has overcome the problem of slow comparison amongst large amounts of geospatial data. We also present the concept of 'branch bypassing', which is extensible to any region-based quadtree and alleviates the problems that arise in quadtrees from a clustered distribution of input data, a common characteristic of occurrence point data. Furthermore, we provide a first-of-its-kind framework to rank museums according to specific criteria in a real-time interactive environment. Details of our work are presented in subsequent chapters.

Chapter 3: The Value Measure

3.1: How to Specify a Value Measure

The idea of using a single value to determine the quality of a species collection is difficult to imagine, let alone justify. From the perspective of a particular species, a strong or valuable collection would contain a large representative sample of the species. However, a small collection might also be valuable if it contains species references in a unique area that no other collection has. As a consequence, the perceived research breadth and depth of collections varies widely among museums and herbaria. Large national museums often have the most important historical collections, whereas regional or university-associated collections frequently have high quality, specialized subsets of species vouchers. For example, university museums often have high quality, in depth specimen representations of narrow taxonomic groups which reflect the personal research interests of their own scientists. Alternatively, species collections may be highly concentrated on a particular geographic region for various historical reasons. For example, a museum in Michigan may have the best collection of Mexican dry forest trees in the world, whereas a collection in Costa Rica may have the best insect representation for Central America, or perhaps the best collection of Central American water beetles, or plants in the coffee family.

There are also many variables in species occurrence data that affect the relative value of that specimen. For instance, the geospatial location, occurrence density, and diversity of environmental variables are just some of the factors that might influence the value of a particular specimen. There are many other potentially valuable attributes, a few of which are the date the specimen was collected, the type of preservation method used, and the amount of specimens in each collection. Therefore, a meaningful value measure might incorporate a combination of all the above traits.

The goal was to find general criteria that could be applied to any species collection. That is, we hoped to establish some sort of a standard baseline. However, we must first look at the data format common to all biological collection data. Occurrence point data is stored in a common data exchange format such as Darwin Core or MaNIS [1]. As a result, each biological domain has different attributes associated with different species. Thus, the only guaranteed common attributes among all occurrence point data sets are the following: scientific name and geospatial location. Our selected criteria are a result of the limitation of these few common attributes, so that a measure of comparison can be reliably performed upon any biological collection. Fortunately, our framework for comparative collection analysis is extensible to easily handle additional attributes specific to a biological domain, as there are no restrictions on the number of attributes that can be added.

The goal was to create a standard by which any museum's biological collections could be compared against that of any other. Therefore, any standard

criteria must consist of a select few common attributes guaranteed to exist in every species occurrence data set. After consulting multiple curatorial and collection managers, at the University of Kansas Biodiversity Institute, with expertise in various biological domains, and considering the previously explained limitations of common attributes, we decided to focus our initial efforts on three common attributes: location, environment, and 'unique contribution', with the ability to include more attributes for specific domains. The location criterion refers to the geospatial spread of a collections specimen occurrence data. Environment determines the environmental diversity of each specimen occurrence. Finally, the unique contribution refers to the amount of non duplicate specimens a collection owns. These three variables are used to create an overall 'value score' for a species collection.

3.2: Three Value Criteria

In the work described here, we take into account three large categories of information about a species sighting: geospatial location, environmental diversity, and the overall contribution of unique occurrence points. Comparing biological collections based on these criteria provides a useful research planning tool for finding specimens of interest to a biodiversity researcher, and to document quantitatively the actual strength of a collection (depth, breadth, currency, etc.).

The first value measure analyzes geospatial spread of the specimen localities (i.e. species occurrences); a stronger collection will hold species in a wide variety of

locations. The geospatial spread value measure will account for the geospatial location of each occurrence point in relation to others. That is, having specimens from diverse geographical regions significantly increases the value of the collection. The geospatial location value also accounts for unique specimen occurrences. An occurrence that lies far removed from others would indicate a special sighting, assuming it is not an error. The special sighting, or outlier, would increase the total value for its associated species collection. By visualizing this value measure in conjunction with geographical representations of species occurrence points, researchers will not only be able see in seconds which museums have specimens in a geographical area of interest, but also show the actual specimen distribution localities, densities, and spread.

The second value measure analyzes the environmental diversity in which the occurrence points were found. It is well known that species vary in many ways throughout their native ranges, and adaptations to local environmental parameters in one part of a species' range will be different from adaptations (morphological, physiological, life history, genetic, etc.) in another part of the species' range. It is important for museums to know if the artifacts they hold represent unique adaptations or some sort of extreme environmental range of the species' distribution. Environmental variables include data such as precipitation, temperature, pressure, etc. which describe the area in which a species was found. If all the species share similar environmental characteristics, this confirms a truth about the species habitat. However, if there exists one or more occurrence points with significantly different

environmental variables, more information can be gained about this species' ability to survive in other habitats. Visually comparing species collections holdings on these additional environmental variables allows quantitative comparisons to be made of the ecological uniqueness of collections based on the environmental variable values associated with species collection points.

Finally, the last criterion in the value measure is the ‘contribution’. This value measure accounts for the contribution of unique occurrence points provided from each species collection. A strong collection will contribute the most unique occurrence points to the data set. Museums which have unique or rare species collections, unduplicated by collections elsewhere, have that much more ammunition when seeking continued financial support and investment on the basis of their scientific uniqueness. “Unique contribution” can be measured by the number of occurrence points not shared by other institutions. Contribution is a legitimate value measure because the goal is to find the data that maximize the information gained about a species. Duplicate occurrence values with the same variables add little new information to the existing data set. However, a species collection that provides a large number of non-duplicate occurrence points contributes a significant amount of new information to the data set, and as such should have a higher value score. The contribution is a very important dimension for collections when they seek funding from external sources, such as public or private foundations. The uniqueness (both taxonomically and spatially) is a prime consideration for long-term financial support and investment.

Taken together, the location, environmental and contribution measures provide meaningful standard criteria to rank value for biological collections. In another sense, relative value – like beauty – lies in the eye of the beholder: in this case, in the eye of a researcher who is interested in research involving a particular set of species and a particular collection of related attribute data. The three value measures, previously defined, can be applied to all biological collections; but each value measure will have a different level of importance to each researcher. Our approach can be tailored to the needs of any such researcher by allowing them to define the value measures of most importance to their current study, and having collections ranked accordingly.

Chapter 4: Reducing the Data Space

4.1: Reducing the Data Space

The size of species occurrence data sets can be very large. In order to rank the species collections, we had to create a way to analyze and compare scores of these geospatial points in a fast and efficient manner. The analysis of occurrence points involves assessing the geospatial spread by comparing the latitude and longitude of each occurrence point with others in the collection. Furthermore, information about the local environment of each occurrence point in relation to others is also assessed. By analyzing the variety of latitude, longitude, and environmental variables across specimen occurrences, we can assess the strength of each species collection based on the three merits discussed in an earlier section.

Additionally, the comparative analysis must be performed quickly. This application will be used as a starting point for reference or future collaboration. A typical use case involves a user opening the application, loading a species occurrence data set, then performing some analysis or reference, such as noting collections that cover an area of interest or identifying a collection with largest holding of a particular species. The user might then load another data set, perform further analysis, and finally shut the program down. Therefore, it is not desirable for the application to consume a large amount of time performing calculations. Typically though, the analysis of accurate latitude and longitude distance calculations between two points

involves math operations dealing with spherical geometry and trigonometric math functions [6]. This can be a time consuming process when distance is determined amongst a large number of geospatial points. Many geospatial applications do not compute latitude longitude geospatial distance between many points. For example, ArcGIS, a geospatial statistical tool, converts points to a Cartesian coordinate system and then uses the Euclidian distance in an X and Y space to compute an exact distance between points [3]. This method provides an exact solution, and proves to be a fast pairwise calculation between two points. However, because pairwise distance calculation is $O(n^2)$, large data sets (>100,000) require a considerable amount of time to compute.

Fortunately, because we seek to rank museums specimen collections against one another, the exact distance between points is not required. For instance, the location value ranks museums according to the geographical spread of their specimen holdings.

A researcher using the application does not need to know the exact distance between occurrence points for each museum, rather they are interested in the 'largest spread', 'smallest spread' etc. Therefore, instead of calculating the distance between a single occurrence point and every other in the data set, we create a spatial hierarchy based on the latitude and longitude of each occurrence point. Thus, the location measure can exploit a hierarchical geospatial representation of occurrence points to quickly calculate relative geospatial spread. Similarly, the environmental analysis is

computed using an ‘environmental space’ to calculate spread, rather than query each point and compare its environment to that of every other.

4.2: A Quadtree

Figure 1a.

0	0	0	0
0	1	1	0
1	0	0	0
0	1	0	1

Figure 1b.

0	0	0	0
0	1	1	0
1	0	0	0
0	1	0	1

Figure 1c.

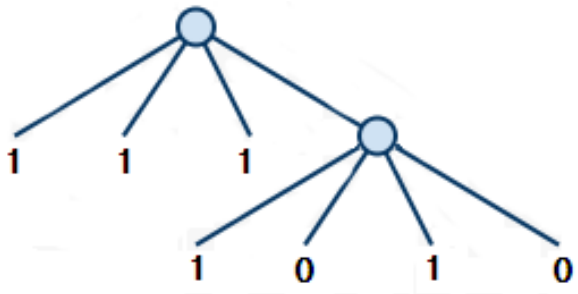


Figure 1a: shows the 2^2x2^2 binary array. 1’s are elements of interest.

Figure 1b: shows a region-based decomposition of the array, with a maximum capacity of 1. At most each quadrant has one element of interest.

Figure 1c: The quadtree representation of the region-based decomposition

In order to create a spatial hierarchy of occurrence points, we use a quadtree. A quadtree is a hierarchical data structure based on a recursive subdivision of space [15]. The type of quadtree used for any given application varies based upon: the input data, the decomposition process, and the resolution of decomposition (variable or not). Quadtrees are most commonly used with point, rectangle, and line data. The process of decomposition is done in either equal parts or based upon the input; wherein the data space is divided into equally sized sub quadrants or the data space is divided at the spatial location of input points respectively. The resolution of a quadtree can be determined a priori, or controlled by properties of the input data.

Two very common types are the point-based and region-based quadtree. As the name implies, the point-based quadtree partitions space based on the location of the input points. A point quadtree is multidimensional generalization of a binary search tree (BST). [11, 4 16] Each node in the point quadtree has four children representing four quadrants of decomposition, and each data point is a node. Thus, the center of each subdivision lies on a point. For instance, the first point inserted serves as the root, while the next point is inserted into an appropriate quadrant of the tree rooted at the first point [11]. This makes the shape of the point quadtree highly dependent on the order of the input data. In contrast, the region-based quadtree is not a generalization of a BST, and instead subdivides the data space into four equal quadrants until each quadrant contains no more points than specified by some ‘maximum capacity’. For instance, when an item is inserted into the tree, it is inserted into a quadrant that encompasses the item's position (or spatial index). Each quad has

a maximum capacity. If that capacity is exceeded, the quadrant then splits into four equal sub-quadrants. This process of sub-dividing will continue until all the elements of the input data are inserted into the tree.

As an example, consider the $2^2 \text{x} 2^2$ binary array shown in Figure 1a. The 1's in the array correspond to elements of interest, whereas the 0's represent blank space. Figure 1c shows a region-based quadtree decomposition of the space, with a maximum capacity of 1. The result is a tree of height 3. The root node of the tree represents the entire array. Each node within the tree corresponds to a subdivision of four equal quadrants: northwest (NW), northeast (NE), southeast (SE), and southwest (SW). Whereas each leaf node represents a region of space containing less-than or equal-to the maximum capacity of elements, in this case 1.

The recursive decomposition of this region-based quadtree enables many benefits. First, by decomposing the $2^2 \text{x} 2^2$ binary space into an equivalent quadtree, the size of the space has been greatly reduced from 16 cells to a much smaller 7 cells of interest. Second, algorithms operating on the elements of interest can now use the hierarchical aggregation to decrease execution time. The speed up is gained from traversal within the tree, usually preorder, which is a linear function of the number of nodes in the quadtree.

4.3: Potential Quadtree Problems

The efficiency and speed of the quadtree operation also largely depends upon the distribution of input data. The use of a quadtree has little benefit if the data is grouped into one small area. “The worst case performance happens when all objects are in one small cluster that is the same size as the smallest Quad; in this case the performance of the quadtree will be slightly worse than just iterating through all objects.” [3]. Fortunately, the nature of specimen occurrence data assures it will rarely occur as a uniform distribution. Unfortunately, there are many cases where specimen occurrences do exist in small areas. One of the significant contributions of the work described here is the development of a novel technique called 'branch bypassing' to address such cases. This technique is described in Section 4.5.

A quadtree data structure can become as good as $O(n)$ complexity but require a large amount of memory as the size of the tree grows. It is possible for a quadtree to increase in size (or total node count) at a rate of $\frac{\left(4^{h+1}\right)-1}{3}$, where h is the height of the quadtree. The rapid increase in size happens because on each successive partition of the data space, four new nodes are created within the tree structure. Each of these nodes takes up a specific amount of machine memory. For example, a node consisting of 5 integer pointers (4 for the children, 1 for the parent) would have a size of 20 bytes. A naïve approach to quadtree construction would result in 4 nodes with 5 pointers each, created on each successive partition of the data space. This growth is not desirable behavior.

As such, there has been extensive research into different approaches to handling the size of quadtrees [3]. As mentioned in [3] there is a method to construct quadtrees without pointers. One such way is through a linear tree. A linear tree stores an image of the quadtree as a preorder traversal of its nodes. The linear tree consists of an array which contains the nodes and leaves of the tree. Using this method, search through the tree can be done in $O(\lg n)$ time. However, the lack of pointers in linear trees makes analysis over the quadtree structure, such as finding neighbors, ancestors, and siblings of nodes, "cumbersome and time consuming"[3]. Alternatively, the number of pointers in the quadtree can be reduced by limiting the amount of nodes created on each partition of the data space. A quadtree can then be constructed where each partition of the data space does not create 4 new nodes, and rather only adds those which are required.

4.4: Our Quadtree

We chose to employ a region based quadtree for occurrence data. The properties of region-based quadtrees are desirable when working with geospatial data [16], more specifically occurrence point data. This is because occurrence point data rarely consists of a uniform distribution. Occurrence data is also bounded by a constant area, the earth, which makes dividing the space into regions very fitting. Furthermore, the benefits of hierarchical aggregation alleviate the need for n by n comparison among all occurrence points. A further benefit of region-based quadtrees

is that their structure allows for sub-division in any data space. Thus we can divide the geographical space based upon latitude and longitude, and also map each occurrence point to its associated ‘environmental space’ simply by changing the latitude and longitude to environmental measures (e.g. average rainfall, temperature, etc) in order to assess the environments of each occurrence point.

Our quadtree implementation goal was to create a quadtree with fast search and node retrieval, while also maintaining memory efficiency. One common implementation method for quadtrees is the pointer based quadtree. A pointer-based quadtree stores parent and child indices at each node. This representation is a very common type of quadtree implementation because it “greatly eases the motion between arbitrary nodes and is exploited by a number of algorithms”[10] such as “search” and “neighbor finding” algorithms. Our pointer based quadtree implementation uses a dynamic array to store the resulting tree. As mentioned in [2], a dynamic array (such as a Vector or ArrayList in Java) minimizes required memory footprint to store a tree.

Our implementation of the region-based quadtree includes a select few enhancements to improve the speed of operations over the quadtree. The optimizations include: storing depth information at each node/leaf, and 'branch bypassing'. These are described in turn below.

Because quadtrees can become very large, we would like to reduce the data space without loosing hierarchical and structural information of the quadtree. In order to achieve this, each node must first store additional information about its location

within the tree. The contents of each node and leaf are described as follows: a node contains its depth within the tree along with indices of its parent and children, if any. Whereas a leaf contains all the variables of a node, including occurrence point data and a value score. Storing depth information eliminates the need to determine the depth of a node when later analyzing and calculating value scores.

4.5: Branch bypassing

Figure 2a

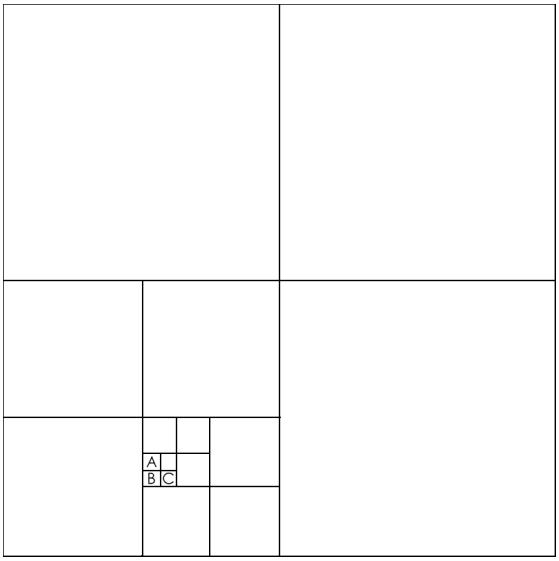


Figure 2b Figure 2c

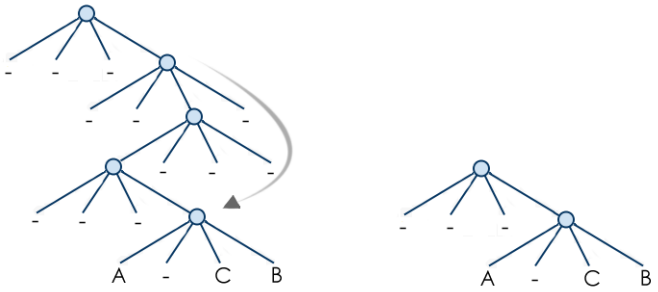


Figure 2a: A region-based decomposition to separate the clustered elements A, B, C.

Figure 2b: The resulting quadtree has a long chain of nodes that contain no information of interest. We can skip those nodes, as long as depth information is stored at every node.

Figure 2c: The resulting smaller quadtree after branch bypassing the long superfluous chain of nodes.

The depth information also provides another benefit, it allows for size reduction optimizations without losing relevant information about the structure of the quadtree. As mentioned in Section 4.3, the inherent nature of occurrence data often has distributions grouped into small areas of the globe resulting in clusters. Small clusters of occurrence points slow quadtree traversal time by creating long branches within the quadtree. Refer to Figure 2b. These long branches convey information about the structure of the quadtree, such as depth. However, these long branches are not absolutely critical because we have previously stored depth and index information in every node. Thus these long branches hold superfluous information already present at each node. We can now eliminate these long branches without loosing information about the tree, through a process called ‘branch bypassing’. Branch bypassing seeks to decrease the traversal time of quadtrees resulting from clustered input data. Branch bypassing the quadtree reassigns parent and children nodes to skip over portions of

the quadtree that are simply long singly-linked lists of nodes. Doing so results in faster tree traversal as documented later.

The branch bypassing algorithm operates by traversing the tree in a pre-order fashion. The algorithm identifies any nodes whose grandchild has only a single node. The grandchild's only child is traversed until a leaf is found or more than one child exists in a node. Finally, the references of child and parent are reassigned between the starting node and the appropriate leaf/node, respectively. It is important to note that no nodes are added or deleted from the internal data structure of the quadtree. As such, the reassignment of node references is a very fast operation.

Before the branch bypassing algorithm can be applied, we assume the quadtree has been fully constructed. The algorithm is run after quadtree construction so it can best detect superfluous branches within the tree and be more easily generalized to any region-based quadtree. Once the branch bypassing algorithm has been executed, no new nodes may be inserted into the tree without first undoing the branch bypassing operation. This restriction is relatively minor, as many reference applications create a quadtree from input files, resulting in a static quadtree that is built once and then later analyzed. For our purpose, we read an input file/s, create a quadtree, and then trim the resulting quadtree; no new nodes are added to the quadtree once it has been created. Although, branch bypassing can be undone just as quickly as it was first performed because the quadtree is stored in a dynamic array as a linear tree. To undo branch bypassing, each parent node that was trimmed is simply flagged during the initial branch bypassing operation. Branch bypassing can be

undone by first traversing to each flagged node. Then the parent pointer of the flagged node's child is reassigned to the node directly to the left of it in the array; and the flagged node's child pointer is reassigned to the node directly to the right of it in the array, restoring the original linear tree.

Branch bypassing can result in a significant reduction of superfluous nodes on highly clustered data sets. For example, consider the real world specimen occurrence data set of *'Dimelaena orenina'* oreina (Ach.) Norman, a lichen species, downloaded from the GBIF online repository. The data set is relatively clustered and contains 203 occurrence points that represent nine species collections in such areas of the earth as Spain, Sweden, North America, and Northern Russia. When we construct our region-based quadtree without the use of branch bypassing, the total number of nodes created is 198, excluding leaf nodes. However, after applying the branch bypassing algorithm, the number of nodes decreases to 135, a 31% reduction in the amount of nodes required for traversal. The reduction of nodes within the tree results in a speed up of traversal time, and consequently decreases the execution time of any algorithm that traverses the tree.

In spite of significant reduction of nodes in clustered data sets, the benefits of branch bypassing decrease according to the distribution of input data. For instance, the occurrence data set for the species *'Argentian anseria'* contains 2,340 occurrence points, but features a less clustered distribution. As such, branch bypassing applied to the subsequent quadtree resulted in only a 7% reduction of superfluous nodes. Refer to Figure 2.1, a chart that shows the effect of branch bypassing on data sets consisting

of different distributions and sizes. On biological collection data, Figure 2.1 confirms that branch bypassing has a greater effect on data sets with a clustered distribution. In contrast, as expected, branch bypassing has little effect on random point data with a random distribution regardless of data set size. Being that biological collection data is most often highly clustered, branch bypassing proves very useful in applications that deal with biological collection data as well as any other point data with a clustered distribution.

DATA SET	DISTRIBUTION	POINTS	TIME TO BYPASS (ms)	NODE REDUCTION
Pisania Ignea	Clustered	75	1.93ms	63%
Dimelaena orenina	Clustered	203	0.72ms	31.8%
Carex scirpoidea	Clustered	611	0.9ms	19.8%
Argentian anseria	Less Clustered	2,340	1.64ms	7.7%
GallinulaChloropus	Clustered	10,000	4.94ms	11.5%
Bombycilla Garrulus	Clustered	30,000	9.30ms	10.45%
Somateria Mollissima	Clustered	50,000	11.62ms	10.75%
Aggregated*	Clustered	100,000	20.99ms	9%
Random Points	Random	1,000	0.92 ms	1.9%
Random Points	Random	50,000	4.56ms	3.3%

Figure 2.1

Figure 2.2

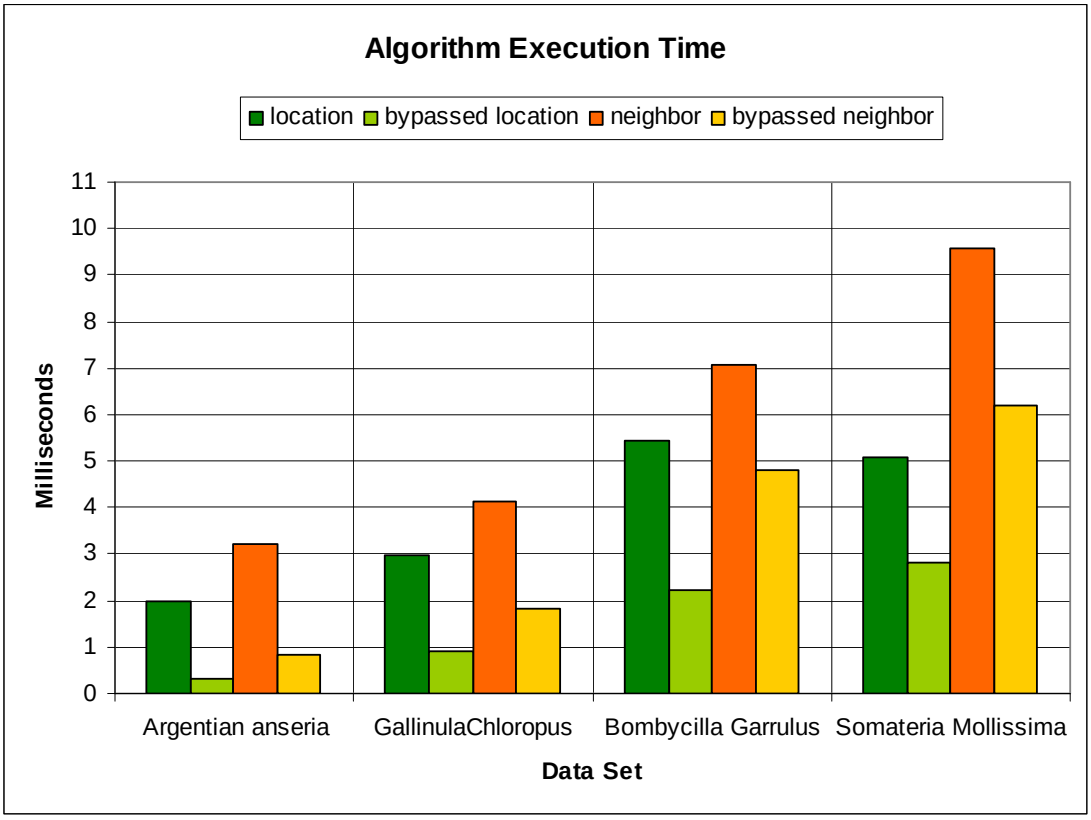
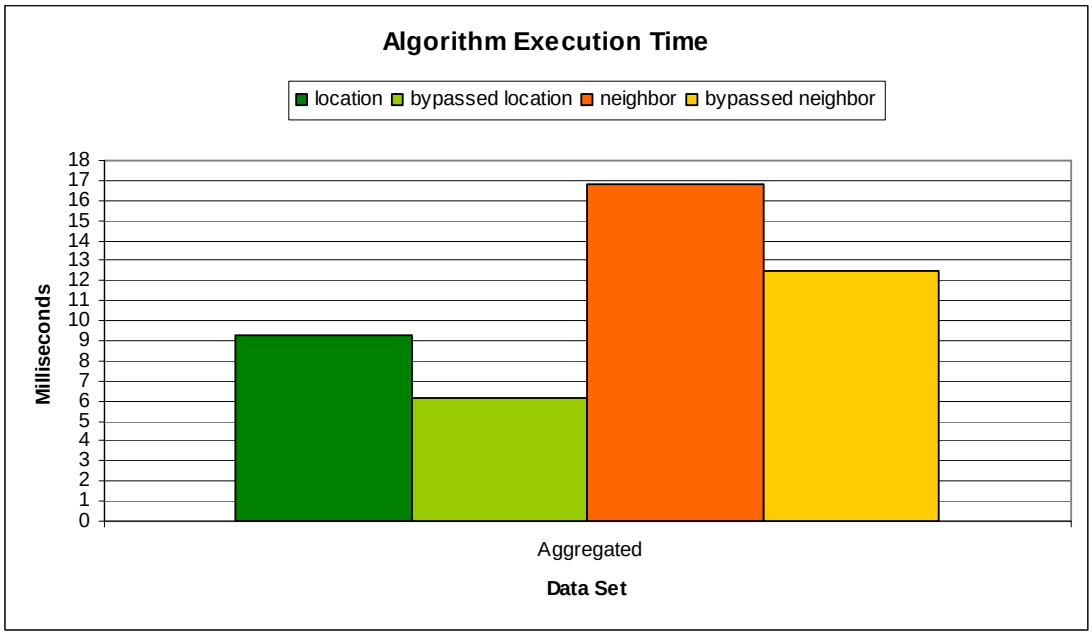


Figure 2.2: A chart that shows four data sets from Figure 2.1. Each data set had two algorithms run (location value measure, and neighbor finding) on untrimmed and trimmed quadtrees.

*There was no single data set with 100,000 occurrence points so we aggregated five different *'Somateria'* data sets to create a file consisting of 100,000 occurrence points.

Figure 2.3



Refer to figure 2.2, which displays the effect of branch bypassing in regards to the execution time of two different algorithms: location value, and neighbor finding algorithms. The location value algorithm assigns a value to each leaf based upon the geospatial spread of its respective occurrence points; the algorithm is described in detail in Chapter 5. We have also used a simple neighbor finding algorithm to test the effect of branch bypassing on quadtrees. The neighbor finding algorithm simply finds an adjacent quadrant, if one exists, directly to the left or right of a given leaf node. For testing purposes our neighbor algorithm attempts to find a neighboring quadrant for each leaf node in the quad tree.

Figure 2.2 reveals that execution time on trimmed quadtrees is reduced for both algorithms. Although, there is no significant measurable reduction in the data set *'Argentian anseria'* because it is less clustered and has a much smaller amount of data. It should also be noted that the amount of speed gained will vary from algorithm to algorithm. Because branch bypassing reduces the total amount of traversable nodes, algorithms that spend time traversing and analyzing the structure of the quadtree will see a greater speed up.

In summary, branch bypassing is a fast operation because only parent and child indices are reassigned and no elements are added to or removed from the dynamic array tree structure. Additionally, branch bypassing is extensible to any region-based quadtree, as the quadtree structure is preserved because depth information is stored at every node and leaf. The branch bypassing algorithm effectively skips over superfluous paths significantly speeding up traversal over a quadtree when the distribution input data is clustered.

Chapter 5: Value Measure Equation

As explained in Chapter 4, a quadtree partitions a data space, allowing us to optimize performance of analysis operations involving the occurrence points. The analysis is based upon the number of implemented value criterion. Our application may use an arbitrary number of value criterion, therefore the number of criterion will be defined as n. These n criteria are combined into an overall score that we use as a value measure. By utilizing the hierarchy of the quadtree, each of the n value criteria equations assigns a score to every occurrence point in the data set. Thus, the total value measure of a provider is the sum of the values over the provider's respective points. The value for a provider, p, is as follows:

$$w_1 * valueCriterion_1(p) + ... + w_{n-1} * valueCriterion_{n-1}(p) + w_n * valueCriterion_n(p)$$

Each 'valueCriterion$_i$' is a value measure ranging from 0 to 1. In addition, the quantities w_i are user assignable weights that also range from 0-1. The purpose of the weights is to provide an interactive way for the user to explore the strengths and weakness of a species collection. Interactivity with the algorithm is explained later in Chapter 6. Therefore, the resulting combined value score for a provider will range from 0 to n. For testing purposes, we have used n=3:

$$w_1 * location(p) + w_2 * environ(p) + w_3 * contribution(p)$$

The first measure determined by the algorithm is the location, more specifically the geospatial spread of occurrence points. The location measure seeks to rank each provider based on the geospatial spread of their respective occurrence

points. The goal is to assign a higher score to providers which possess a large geographic region of coverage, or to providers which have occurrence points in an area that no other provider has. It must be possible to compute the location measure very quickly, without resorting to the costly and time consuming $O(n^2)$ comparison of all occurrence points. In turn, we have developed a fast calculation method that takes advantage of the quadtree, and alleviates the need for n by n comparison. Thus, the location measure algorithm can exploit the hierarchical spatial representation of the quadtree to quickly calculate relative geospatial spread, rather than compare the latitude and longitude of every occurrence point with that of every other point.

In order to do this, the location measure performs a pre-order traversal of the quadtree and assigns a value to each leaf node based upon two criteria. The first criterion is computed by determining the relative depth of the leaf node within the tree. A leaf node deeper within the tree implies a close geospatial proximity to other occurrence points, and as such, the leaf node would be given a lower score. However, a node's depth in the quadtree alone does not reliably determine geospatial spread [10]. The second criterion addresses this problem by adjusting the maximum capacity. The second criterion addresses this problem by adjusting the maximum capacity of each quadrant, in other words adjusting the resolution. The quadtree has an initial maximum capacity that defines the maximum number of data points that reside in each quadrant. When the maximum capacity in increased, a quadrant will encompass more data points. Therefore, in regards to the location measure, by

increasing the quadrant size we encapsulate those occurrences which lie in close geospatial proximity together. The method of interactively increasing the maximum capacity proves useful to normalize the results when one provider has a significantly large number of occurrence points in a small geographical location.

While the location measure does use the spatial hierarchy of the quadtree, it also is fortunately not tightly coupled to the structure of the quadtree so that it can consistently produce valid results. We would expect that the same data sets would have similar results regardless of where in the quadtree they are rooted. For example, a data set of '*GallinulaChloropus'* with 10,000 occurrence points has three providers. On an initial run of the application we recorded the first, second, third and last place rankings. Next, for the sake of testing, we moved the geospatial locations of each provider's occurrence points and ran the application again. The location measure provided a different numerical score, but the application continued to rank the providers in the same order: the previous first place was still in first, the previous second stayed in second, and so on. Further tests moving the providers occurrence data yielded the same results, fluctuations in numerical score while keeping the same ranking. The location measure provides a fast approximation of geospatial spread, and does so regardless of the where in the tree a collection of points of rooted.

Diversity of environmental variables also contributes to the overall value measure. Ideally a user would like to know if a provider has occurrence points in an environment that no other provider has. The second part of the algorithm, environmental measure, provides a measure of diversity. Our application has twenty

environmental layers that species may be compared with. Environmental layers are raster data files representing some region of the world. Each cell in the layer is a measurement of the parameter associated with that layer [7]. The twenty layers in our application each represent the entire earth for some parameter; for example, average precipitation, or mean temperature. The environmental measure then creates a quadtree by mapping each occurrence point to its respective point in a new environmental space.

The environmental space is determined by two user chosen environmental layers that represent the x and y axes. Users may choose any combination of environmental layers. By using a quadtree that consists of two differing environmental variables as the x and y axes, we can create an environmental space in order to identify niches; a niche is the relational position of a species in its ecosystem compared to others of the same species. Then we can traverse the tree and assign values in the same manner as the location value measure. It is important to note that these values are not based on a geospatial distance. The reason is because the environmental space can represent any combination of environmental layers, thus a distance measure does not have biological meaning. Instead, the environmental measure can aid in the identification of niches and correlations within the chosen environmental space. Calculating the environmental diversity in this manner provides the benefit of fast determination of spatial spread and the ability to combine any two environmental variables. The environmental measure provides valuable insight into the relation between environmental attributes of the occurrence points.

The final part of the value measure algorithm computes a contribution amount for each provider. The goal for the contribution measure is to determine a ratio of how much new information a provider adds to the existing set of information. Each provider contributes a certain number of occurrence points to the total data set; a provider may or may not have duplicate occurrence points which affect the amount of unique information the provider contributes. Duplicate points can either be shared among other providers or represent duplicates from within that provider's own species collection. Therefore, a distinction between duplicates must be made. Duplicates can either be 'internal' or 'shared'. An internal duplicate represents a duplicate occurrence point with only one provider. Internal duplicates arise from human error, when the same occurrence point is entered into a provider's data set more than once. Whereas, a 'shared duplicate' (SD) denotes an occurrence point shared amongst multiple providers. The contribution measure can then be computed for each provider by:

$$\frac{ST_{p(x)} - SD_{p(x)}}{(ST_{p(1)} - SD_{p(1)}) + \mathrm{K} + (ST_{p(n-1)} - SD_{p(n-1)}) + (ST_{p(n)} - SD_{p(n)})}$$

Where p is a provider and x denotes the current provider. ST is the 'specimen total', and SD represents 'shared duplicates'.

The ' specimen total' (ST) accounts for every unique occurrence point the provider owns. More specifically, 'specimen total' is the sum of a provider's non-duplicate and internal duplicate occurrence points. However, an internal duplicate only accounts for one specimen occurrence sighting, no matter how many internal duplicates represent the same point. The reason internal duplicates only account for a single occurrence

point is because they arise from human error at the time the data set was created. One must consider that every occurrence point originally existed on a paper record, and these paper records were entered one by one into a data set by a person; there is bound to be the occasional mistake or duplication of a species occurrence point. For example, if a provider held 3 occurrence points and 2 of them represented the same occurrence, the provider would have one non-duplicate and one internal duplicate occurrence point. Therefore, the ' specimen total' would result in 2 occurrence points.

The denominator of the contribution equation essentially represents every unique specimen occurrence point in the data set by summing the result of every providers ST subtracted from its SD. By calculating the contribution for a provider in this manner, we can subtract a providers shared duplicates from its specimen total, and then divide by all unique specimen occurrence points to get a ratio of the amount of unique information the provider contributes to the data set. For instance, if a provider held the majority of occurrence points, but all of its occurrence points were shared duplicates, then that provider would add no new information to the data set.

Our application may be extended, with minimal programmer effort, to include relevant attributes and equations for a specific domain. Additional attributes require a new graphical user interface element (GUI) and may either require a new quadtree or may perform some analysis over an existing quadtree. For instance, the implementation of environmental variables only required a new quadtree with a different data space, environmental rather than geospatial, and a graphical user interface (GUI) slider bar. Similarly, the contribution attribute was added by

including an additional statistical calculation into the final value score, and inserting a GUI slider bar element.

Chapter 6: Real-time Interactive Framework

6.1: User Interface

The comparative analysis of specimen occurrence points produces a large amount of data that must be easily interpreted by the user. Therefore, a suitably intuitive and real-time interactive visual representation is necessary to provide a method for assessment of results. Users must be able to view their specimen occurrence data along with specimen data from other institutions, in order to quickly see the overall geographical layout of a particular species. By displaying each occurrence point (color coded to match their associated provider) on the globe, users can quickly identify the global coverage of each species collection. The application must also provide a clear visual representation of the value for each collection in question. Value should be displayed in such a manner that allows it to be easily and quickly interpreted. An elegant way to represent rank among many objects is a bar chart. Although for our purpose, the bar chart is dynamic and reacts instantly to user adjustments of the value measure algorithm. For instance, users can adjust the weights of each criterion as they see fit, and likewise the bar chart will reflect those adjustments.

For the purpose of visualization, the application uses NASA's WorldWind. Similar to Google Earth in its visual capabilities, WorldWind is an application API and framework that leverages Landsat satellite imagery, Shuttle Radar Topography

Mission data, and a variety of surface elevation models to view the earth's terrain in 3D [18]. However, WorldWind is a more fitting choice than Google Earth because it is open source, and it is designed to be used as an application framework to facilitate the development of custom applications such as ours. WorldWind also provides a visually rich 3D environment to view the earth along with any other form of geospatial data.

The user interface for our application consists of three main panes. (See Figure 3.) The first and largest pane holds an interactive view of the earth and all of the occurrence data. Users can zoom, spin, and change the projection (2D sinusoidal, 2D lat lon, 3D globe, etc) of the earth as they see fit. The second pane resides on the bottom of the screen and displays a dynamic bar chart. The bar chart displays the current value of each species collection. Additionally, the bar chart changes in real time as the user adjusts the weights of the three value measures. Finally the third pane, which sits on the right side of the application, displays statistics about the currently selected provider. Such statistics include the score for location, environment, and contribution. Most importantly, this pane holds the sliders to adjust the weights of each value measure criterion.

Figure 3

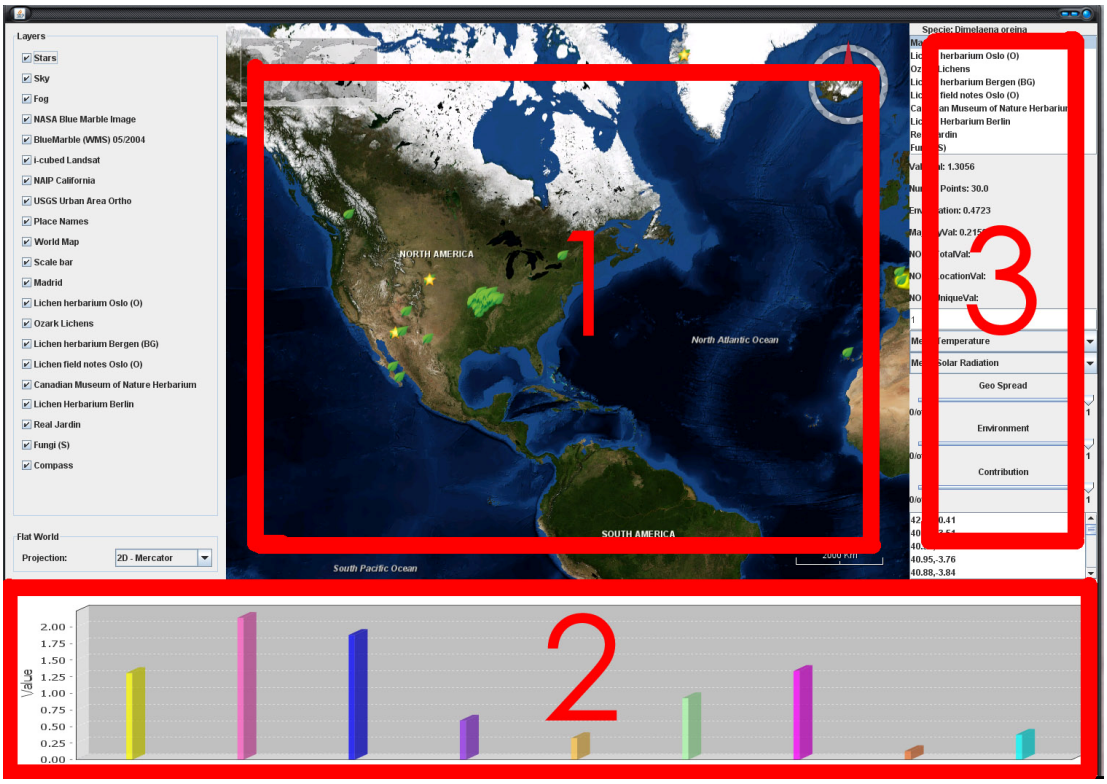


Figure 3: (1) The WorldWind display. (2) The dynamic bar chart that instantly reflects user changes to the value measure algorithm. (3) The collections statistics and slider bars for the three value criteria.

6.2: The Role of User Interaction

The results of comparative collection analysis must make logical sense to users. The graphical user interface (GUI) should effectively convey the rank of each collection. A naive GUI approach could feature a static ranking, unchangeable through user interaction, for each collection. However, a static ranking is detrimental to the goal of comparative collection analysis. We want users to not only see the rank,

but also to understand why a species collection is ranked in a certain way. Users gain insight into their data by not only seeing the results, but by also interacting with the value measure algorithm to better comprehend the value score. Through interaction, users can assess which factors contribute the most or least to the overall value score.

User interaction is achieved through the use of weights, in conjunction with GUI 'slider bars'. As previously shown in the provider value measure equation; w_1, w_2 and w_3 are weights that range from 0 to 1, and are multiplied by the location, environment, and contribution measures, respectively. The weights affect the influence that each value measure criteria has on the final value score. Each slider bar adjusts the weight of its respective value measure from 0 to 1. As a slider bar is adjusted, the dynamic bar chart and geographical displays are updated in real-time to reflect changes to the overall value score. This interaction allows users to focus on those parts of the value measure of interest to them. For instance, if a user is only concerned with environmental diversity, they can use the sliders to turn off the location and contribution measures. In this manner, users can explore the data to determine strengths and weaknesses of multiple collections. Through customizable interaction, users not only view, but also understand the results of the value score.

User interaction further benefits from real-time interactivity with the application. Researchers and scientists will most likely use comparative collection analysis of this type as a reference tool. Therefore, the application must perform its analysis and display the results quickly. When talking with a museum curator we learned of another open-source geospatial visualization program called GeoDa that

can not handle input data sets over 25,000 points, and with data sets over 10,000 it becomes very sluggish. In contrast, our application can effectively analyze several tens of thousands of species occurrence points. For instance, it takes less than 3 seconds to construct two quadtrees in order to analyze all three criteria of an occurrence data set that consists of 100,000 occurrence points (A 3Ghz quad-core computer with 4Gb ram, and our quadtree set with a maximum capacity of 15.). The speed of analyzing very large data sets is made possible by our quadtree and fast geospatial calculations.

6.3: Achieving Real-time Interaction

Real-time interaction is achieved through the speed up gained by using our quadtree and slight modifications to WorldWinds icon render, explained further in subsequent paragraphs. The application takes only seconds to load a species data set; thereafter the user experiences no ‘input lag’ with the application. Input lag is the time required for an application to respond to user input; generally input lag less than 1 second is acceptable for an application of this type. The input lag is minimized by performing the value measure calculations while a data set is initially loaded. Thus, when a user modifies a criterion of the value score, the application does not need to re-traverse the quadtree and recalculate the value score; instead the application updates the dynamic bar chart and geographical visualizations to display effects of the

user's action. There is only input lag when the user changes the environmental layers; in this case a new quadtree must be built and analyzed for a new environmental space.

The quadtree data structure also enables fast node retrieval upon a user query. When a user selects a group of occurrence points, the application can rapidly query and display more information about the selected points. As mentioned earlier, the speed of node retrieval is attributed to the fact that every node stores its index within the dynamic array, allowing for rapid retrieval of all the information associated with relevant occurrence points.

As of this writing WorldWind 1.4 was not able to efficiently display a large number of icons. For example, 2,000 occurrence point icons would restrict the display to running very slowly, about 5-10 frames per second. However, with some modification to the WorldWind icon render, we were able to allow it to display a large number of icons (3,000 icons) at 20-23 frames per second. Our modification involved view based culling of icons. That is, any occurrence icons that resided outside the current viewable sectors of the display were not rendered. Additionally, another modification enabled the ability to draw icons in a large batch in order to save OpenGL state switching. For the purpose of informative visualization, it was not beneficial to display more than 3,000 occurrence point icons at a time. Therefore, on very large data sets exceeding 3,000, only a sample of the occurrence point data is displayed. Together our quadtree data structure, value algorithm, and WorldWind icon renderer modifications allows the user interface to achieve real-time interactivity.

Chapter 7: Evaluation

7.1: Data Set and Test Subjects

A survey was administered to test and evaluate the usefulness of our comparative collection analysis. For the purpose of evaluation, we used the occurrence data set of *Dimelaena oreina* (Ach.) Norman, a lichen species, downloaded from the GBIF online repository on January 10th 2009. The data set contained a relatively clustered distribution of 203 occurrence points from Spain, Sweden, North America, and Northern Russia among other areas. Those specimen-based occurrence points for the species came from nine biological collection institutions. The data with its widely dispersed and clustered occurrence points nicely exercised the various dimensions of the ranking system. In contrast, we also used GBIF occurrence data for the weed species *Argentina anserina* (L) Rydb. This contained 2,340 occurrence points from seven institutional collections. This data set contained over 10 times the number of points and had a much less clustered distribution than the lichen data set.

Our application is targeted toward users who have a background in museum collections. As such, the test subjects included scientists and curators with a strong background in managing biological collections. There were 5 test subjects, 3 men and 2 women, from the University of Kansas Biodiversity Institute. The test subjects areas of expertise were quite varied, those areas included: ichthyology, mammalogy,

entomology, and botany. Testers were either curatorial or collection management staff.

7.2: Results

The survey was administered as follows: First, the test subject was given a brief explanation of the purpose of the application. Next, a concise explanation was given of the GUI controls of the application and how to use them along with a brief explanation of each value measure. The test subject was then free to explore and manipulate the application in an undirected manner and asked to vocalize any thoughts, questions, or concerns.

We evaluated whether the results of comparative collection analysis were useful. Four of the five participants confirmed that the results of our application made logical sense. They agreed and understood why certain species collections were ranked near the top or bottom. Furthermore, one test subject, with a background in botany had prior professional knowledge of reputable herbaria collections, confirmed that our application did in fact correctly rank the top collection in the lichen data set based on his knowledge of the field. The overall reactions to comparative collection analysis were positive. The participants echoed that our application was beneficial for collaboration and provided a method to document strengths and justify future collection priorities. Other participants said that our application was useful for finding gaps in geographical coverage, as well as providing an incentive for curators to make

their collection data available online. Furthermore, three test subjects expressed their desire to use the tool on their own collections, as there is currently no such way to assess these qualities of a species collection.

There were also negative reactions to the application. A concern expressed by the first three test subjects involved the coloring of icons and collections. The colors among icons were difficult to distinguish. For that reason, we adjusted the colors, and the following two test subjects made no mention of difficulty distinguishing among colors. Participants expressed another concern: they thought it would be helpful to rank collections on a genus and family level. This concern could be easily addressed, because instead of importing one occurrence data file we could just import multiple occurrence data sets to build a quadtree. Another repeated concern was that it would be helpful to have other criteria for comparison. Each test subject suggested different additional criteria, which served to reinforce our previous notion that researchers in specialized biological disciplines have different priorities on important criterion. Fortunately, our framework for comparative collection analysis is extensible to easily handle additional attributes specific to a biological domain, as there are no restriction on the number of attributes which can be added. As a result, all participants appreciated the ability to weigh the different criteria as they saw fit. Finally, they all agreed that the three criteria we have defined (location, environment, and contribution) are criteria they would also use as a standard to rank biological collection holdings.

The evaluation of comparative collection analysis proved successful. Based on test subject feedback, the application correctly ranked collections. The three value criteria also proved to be more than sufficient for ranking biological species collections. Test subjects also understood how and why collections were ranked, and they could also foresee many uses for comparative collection analysis. The user interface also provided an intuitive method to adjust weights for all three criteria. Additionally, our application quickly processed a large number of occurrence points, and no test subject mentioned ‘input lag’ or ‘slow response’ of the application. The results of our comparative collections analysis method delivered insightful aspects into the value of ranking species collections, and illustrated how our approach is an effective ranking and analysis tool.

Chapter 8: Conclusions

8.1 Summary

We have described the design and development of several novel techniques. The first is the method of "branch bypassing" for quadtrees which speeds traversal time by skipping over superfluous paths within the quadtree created from clustered data sets. Branch bypassing also preserves the structure of the quadtree because it neither removes nor deletes nodes from the underlying data structure. We also developed an extensible framework for evaluating point data sets, and created a standard general value measure to rank biological collections on three criteria: geospatial spread, environmental attributes, and 'unique contribution'. User testing further reinforced the usefulness and benefits of our application in the biological collections area. Furthermore, we developed a fast geospatial spread approximation which exploits a quadtrees spatial hierarchal aggregation to quickly operate over large sets of data. Finally, we developed a real-time interactive application to allow users to explore and understand how and why collections are ranked amongst each other.

8.2 Conclusions

We have proposed a framework to create an interactive environment, suitably intuitive, easy to use, and fast that allows massive data collections to be efficiently rated/ranked according to standard criteria. We have found that the quadtree structure, in conjunction with branch bypassing, effectively provided a fast and efficient way to

calculate geospatial spread, while concurrently working with large data sets. The geospatial spread calculations are also extensible to other applications that assess geospatial spread on large sets of data. Our framework also provided a first-of-its-kind useful value measure based upon three standard criteria in order to rank biological collections specimen holdings. Furthermore, branch bypassing proved to be a fast operation, because it neither deleted nor added nodes to the underlying dynamic data structure, to increase traversal time of the tree while preserving the original quadtree data structure. Branch bypassing can also be generalized to any region-based quadtree. The results of our application will lead to a richer description of collection holdings of interest to researchers in biological sciences, resource planning, development, and collaboration.

8.3 Future work

Though the results given in this paper show the success of comparative collection analysis, there is room for possible future directions with the application. The environmental impact of the species sightings could be further investigated by overlaying various maps on to specimen occurrence data. For instance, modern land use, land change maps or remote sensing data overlaid on top of historic species distributions could prove highly valuable for a museum to know that they may have the last or only collections of species from a habitat that no longer exists. The application would also benefit from the ability for users to reduce their comparisons to specific areas of the globe. Additional value measures could be incorporated, such

as preservation method, date collected, etc. These kinds of analyses, which can only be done with cross-collection comparisons, and which can only be done efficiently and interactively with desktop geospatial visualization tools, will be highly valued by biodiversity collections, researchers, students, curators and administrators.

References

[1] “Early History of GBIF”. Global Biodiversity Information Facility. 4 April. 2004 <http://www.gbif.org/GBIF_org/facility/history_gbif>

[2] Bantchev, Boyko. Representing Trees. In *Mathematics and Education in Mathematics.* Pages 193-196. 2007

[3] Beyer, H. L .“Distance Between Points Tool”. *Hawth's Analysis Tools for ArcGIS*. 2004. http://www.spatialecology.com/htools/pntdistbetw.php

[4] Black E. Paul, ed. Algorithms and Theory of Computation Handbook, CRC Press LLC, 1999, "quadtree complexity theorem", in Dictionary of Algorithms and Data Structures., U.S. National Institute of Standards and Technology. 17 December 2004.

[5] M. de L. Brooke. Why museums matter *Trends in Ecology & Evolution*, Volume 15, Issue 4, 1 April 2000, Pages 136-137

[6] Kennedy, Michael. Introducing Geographic Information Systems with ArcGIS. Wiley, 2006.

[7] *LifeMapper Services*. April, 2001.< http://lifemapper.org/services/lm2/layers>

[8] Dinesh P. Mehta, Sartaj Sahni. Handbook of Data Structures and Applications. *Chapman & Hall/Crc Computer and Information Science Series.* Page 19-3, October 2004.

[9] H. Samet. Data structures for quadtree approximation and compression. *Communications of the ACM*, 28(9):973-993, September 1985. Also *University of Maryland Computer Science Technical Report* TR-1209, August 1982.

[10] H. Samet. The quadtree and related hierarchical data structures. *ACM Computing Surveys*, 16(2):187-260, June 1984.

[11]H. Samet. Multidimensional spatial data structures. In D. Mehta and S. Sahni, editors, *Handbook of Data Structures and Applications*, chapter 16. CRC Press, Boca Raton, FL, 2005.

[12] H. Samet. Neighbor Finding in Quadtrees Hanan Samet. 1981.

[13] H. Samet. Using quadtrees to represent spatial data. In H. Freeman and G. Pieroni, editors, *Computer Architectures for Spatially Distributed Data*, pages 229-247. Springer-Verlag, Berlin, West Germany, 1985.

[14] H. Samet, A. Rosenfeld, C. A. Shaffer, R. E. Webber. Processing geographic data with quadtrees. In *Proceedings of the 7th International Conference on Pattern Recognition*, pages 212-215, Montréal, Canada, July 1984.

[15] H. Samet, A. Rosenfeld, C. A. Shaffer.Use of hierarchical data structures in geographical information systems. *In Proceedings of the International Symposium on Spatial Data Handling*, pages 392-411, Zurich, Switzerland, August 1984.

[16] H. Samet, A. Rosenfeld, C. A. Shaffer, R. E. Webber. Quadtree region representation in cartography: experimental results. *IEEE Transactions on Systems, Man, and Cybernetics*, 13(6):1148-1154, November/December 1983.

[17] A. Rosenfeld, H. Samet, C. Shaffer, R. E. Webber. Application of hierarchical data structures to geographical information systems. *Computer Science Technical Report* TR-1197, University of Maryland, College Park, MD, June 1982.

[18] "WorldWind". *National Aeronautics and Space Administration*. 2004 http://worldwind.arc.nasa.gov/