VirtuaForest: A Tool for Simple and Versatile Tree Data Visualization

Joel Fichter fichterJL@my.hiram.edu Computer Science Michael Benedict BenedictM@hiram.edu Environmental Studies Louis Oliphant OliphantLT@hiram.edu Computer Science

Hiram College 11715 State Route 700 Hiram, OH 44234

ABSTRACT

Hiram College conducts long-term vegetation monitoring to investigate changes in the structure and composition of the forests at the College's J.H. Barrow Field Station. This effort includes sampling tree dimensions data in fifty by twenty meter plots, with the goal of developing vertical and horizontal profiles of the forest structure. These structure data are traditionally depicted for visualization by hand-drawing a representation of the trees in each plot. This process is tedious and timeconsuming. This paper describes a Java3D application that allows users to input such tree data and have the program quickly and accurately create visualizations from multiple perspectives.

1. INTRODUCTION

Hiram College, located in a heavily forested part of Northeast Ohio, owns and operates the James H. Barrow Field Station, which contains one of the few remaining old-growth Beech-Maple forests in the state of Ohio. Given its conservation relevance, in 2009 the Environmental Studies Program, in partnership with the Biology Department, began long-term monitoring aimed at understanding the dynamics in the old-growth and surrounding successional forest stands as well as contributing to conservation-oriented management. Forest sampling follows the Vegetation Carolina Survey (CVS)





protocol[8, 9] and is done in several permanently marked fifty by twenty meter plots. These plots are further subdivided into ten separate ten meter by ten meter squares, or "modules", which are numbered and used to specify a tree's location in the plot as shown in Figure 1. Faculty and students in the field take measurements of various attributes of the trees in these plots. Data collected about each tree includes attributes such as height, trunk diameter, and position in plot, which are then recorded using a standardized form. These are then used to draw a graphical data representation of the trees within the plot to which it corresponds from two perspectives: a top down view and a side view. The students tasked with creating these drawings must painstakingly measure out the exact dimensions of each tree before drawing it in by hand from each perspective. When recording a plot that contains dozens upon

dozens of trees, this can easily become a daunting task.

The primary goal of this project was to create a program using Java3D that would perform this function automatically, creating a graphical representation of any given plot with accuracy comparable or superior to that of the hand-drawn method. The standardized nature of the tree data collection made this achievable, however not every institution carrying out this type of forest research uses the same plot design in the tree sampling method. One of the secondary goals of this project was to use a coordinate system general enough that many different data formats could be adapted to fit it, and yet not require an excessive amount of conversion to work with the program.

In addition to these goals, we also attempted to include features that would make the resulting graphical representation easier to read and therefore more useful to researchers studying the layout of each plot. The most notable of these is the inclusion of a 3D view mode that allows the user to navigate the plot in 3D space and look at the trees from any angle they desire, something that would not be possible using the traditional pen and paper method.

Finally, a constant emphasis was maintained on keeping the program as simple and as easy to use as possible, so that it would not require a great deal of time or training to use effectively, regardless of the user's level of experience with computers.

2. RELATED WORK

There are several programs which have similar functions to the one featured in this paper. One related technique is to use a Geographic Information System (GIS) to store and manage data collected by a Light Detection and Ranging (LIDAR) technology, and using this data to create a 3D representation of a forest[4]. This approach can be used over a wide area, does not require manual data collection, and creates a fairly accurate representation of the forest. On the other hand, the technology required is prohibitively expensive and specialized, and the focus is on creating a generally accurate model over a wide area on a large scale, not necessarily reproducing the minute details required for more local applications.

Another, similar method uses GIS in conjunction with aerial photography, but this limits the perspectives of the representation to an aerial view as it cannot capture the structure of the forest beneath the upper canopy[2].

Graphics based approaches have resulted in extremely realistic forest scenes, but use predefined tree models and are not focused on analyzing forest data[3].

The closest analog to the program described in this paper is a tool known as the Stand Visualization System (SVS)[1]. SVS was developed at a research station of the United States Department of Agriculture (USDA) Forest Service, and uses data collected from stands of trees to create graphical representations of those stands. Features include the ability to differentiate between different species of trees, as well as including representations of underbrush, other types of foliage, fallen logs and debris, and other types of objects. However, the interface is laden with options that may not be useful to many research based enterprises, and data entry is a rather time-consuming process due to the wealth of options that are available. In addition, the size and shape of the plot are rather restrictive given the various plot shapes used by various institutions. While there are many features that could prove useful, learning how to operate all of them could prove to be time consuming for the average user

The goals of this program are to provide a tool for electronically rendering forest structure data in a side view (vertical profile), a top-down view (horizontal profile), and in a 3D view. While less complex than the aforementioned examples, it is a useful instrument that has broad utility for forest ecology and forestry applications in the academic, industry, government, and community settings.

3. BACKGROUND

Forest profile diagrams provide information about the vertical and horizontal spatial structure of trees within a section of the forest. Vertical profiles have been traditionally used by ecologists to depict the vertical structure and complexity of forest stands[5] and continues to be widely applied in forestry practices[6].



Figure 2: Hand-drawn vertical (A) and horizontal (B) profiles for Old-Growth Beech-Maple forest, James H. Barrow Field Station, Hiram College. [7]

The vertical profile (Figure 2.A) presents a side view that allows for the visualization of individual tree heights, the height of individual tree crowns, the distribution of tree species among vertical strata (e.g., canopy, subcanopy, understory), the width of

individual tree crowns, and the presence of gaps along the length of the sampled area. The horizontal profile (Figure 2.B) presents a top-down view (i.e., a ground plan) of the entire sampled area that includes the mapped position of individual tree bases, the dimensions (width and depth) and relative positions of individual tree crowns (including overlapping patterns), and the presence of gaps within the sampled area. Because forest profiles are developed using measurements taken in the field, they can be drawn to scale to provide an accurate representation of spatial patterns in the vertical and horizontal dimensions.

The data collected on each tree in the plot is as follows: the number of the module the tree occupies (from one to ten, Figure 1), the species of the tree, the trunk's diameter at breast height (DBH, measured in centimeters at 1.4 meters height), the height of the top and base of the tree's crown (meters), the horizontal length to which the crown extends from the trunk in each of four cardinal directions (meters), the length (the tree's distance along the centerline that bisects the plot lengthwise, measured in meters), and the depth (the tree's perpendicular distance from the centerline within its module, measured in meters). This setup for recording tree position is markedly different from the standard 2D pair coordinate system in that whether a tree is above or below the centerline is not indicated by whether its depth is positive or negative but the number of its module. If the above the centerline. module is the displacement is in an upward direction, if the module is below the centerline, it is the opposite (see Figure 1). Negative depth values are not used to indicate which side of the centerline a tree is on. Instead, negative depth values are used to indicate the position of trunks of trees situated outside of the plot, but which still have branches intruding over the plot which causes them to be measured

for cover with the rest of the plot's trees. Additionally, field data for each tree is paired with a small sketch that represents its general shape and architecture. Because there is no way to accurately or easily translate this sketch into information that would be useful to the program, this information was omitted from the program design.

In both side and top down views, the trees are drawn to scale with small acronyms that allow the reader to unambiguously determine a tree's species, even in the black and white drawings. A tree which is obscured by another is drawn using dotted lines to indicate that it has been concealed from view, but when a large number trees in a small area are blocked off it becomes very difficult to determine the relative depths of individual trees, as they all have dotted outlines. The species of the trees also posed a problem. One of our primary goals is to develop an application that can be used by institutions in any geographic location, therefore we have taken into account the possibility that other institutions may have species that are not present in Hiram's tree plots. For this reason, the list of possible tree species is kept flexible and modular.

4. CREATING THE TREE VISUALIZER

We have created a tree visualizer program in the Java programming language using the Java3D library for rendering the different graphical representations of the plots. We have created several Java classes to represent the data and different aspects of the program. We will explain these Java classes.

4.1 Creating the View, Plot, and Tree classes

Our application displays multiple kinds of views using the same data set. To achieve this, the architecture of the program uses a View class that is constructed using the Java3D library and is capable of displaying trees in three different modes (3D, side, and top down). Figures 3 & 4 show top down and side views that have been created using actual plot data. The Plot class contains the data for all trees in the plot, and is accessed by the View class. Whenever the user desires to edit the data in the plot or load a new plot altogether, it is the Plot class that changes.



Figure 3: Side view of plot generated by VirtuaForest



Figure 4: Top down view generated using the same data as Figure 3

We created a separate Tree class that would be used by the Plot class to keep track of each tree. The Plot class holds Tree objects in an array list so that any number of trees can be added to a given plot.

The task of actually rendering the scene is split between the Plot and Tree classes. The Plot class contains the method used to draw the lines of the plot on which the trees would appear, based on the length and

width parameters it is given. These parameters are not currently able to be modified by the user, but in theory they would allow a plot of any dimensions to be used, although this functionality has not yet been thoroughly tested. All of the methods used to draw the tree crowns and trunks are contained in the Tree class. Each Tree is capable of drawing itself for each of the different modes, using separate methods for drawing the trunk and crown in Top Down, Side, or 3D views. The Plot class iterates through the array list of Tree objects, calling the appropriate draw methods for each one, and passes the resulting Shape3D objects up to the View class.

4.2 The User Interface

After developing the mechanism for displaying trees in 3D, we created a simple user interface. We used NetBeans to arrange the interactive elements in the user interface, but most of the work involved creating a custom table model that displays the data held in the Plot class, and that allows the user to edit the data for each Tree.

Originally the plot graph was displayed in a separate window, but while working on the user interface we realized that since most users would want to compare the information in the table with the graph the layout would work better having all elements in the same window. We created a Jpanel in the interface window and used that to display the plot. We also added a filtering system that allows the user to specify which trees should and should not be displayed based on various criteria. The only two filters existing right now are for trunk diameter and overall height, but filtering for other attributes such as individual species could easily be implemented using modified versions of the existing code. The current version of the user interface can be seen in Figure 6.

4.3 Saving and Loading

After establishing a user interface, we added a method for saving and loading tree data. For the application to be useful we included a way to load and store tree data to a file. We made this part of the program so that it could read .csv files created in Excel or similar spreadsheet software. In addition. spreadsheets are widely used for entry and analysis of field research data. We also added the capability to save to a .csv file so that changes made to the data within the program were reflected in the file. This would allow users to experiment by giving trees different attributes or correct data entry errors within the program without having to modify the original file directly.

We also modified the way the program kept track of the various species of trees. Users in different regions need a program that can recognize the local species of trees, some of which may not exist in Northeast Ohio. By default, the program does not initially contain a list of tree species. When the load method is reading in a set of data, it checks the species of each tree it loads to see if that species name has already been encountered, and if it has not, it adds the name of that species as a key to a hashmap of Species objects, which contain data about how trees of that species would be rendered visually. In this way, the program updates its list of species each time data for new tree species are loaded. rendering the program useful in any region.

All of the saving and loading methods are implemented in the Plot class, where values read in from the .csv files can be directly used to create new Tree objects that populate the plot's array list.

4.4 2D Viewing Modes

We focused on the 2 dimensional views because they would likely be the most used

features of the program. This involved rendering the same data used for the 3D view, but with several changes: the camera reoriented to reflect a top down or side view, the projection mode changed to orthographic rather than parallel, the 3D models replaced with Line Arrays and flattened spheres/cylinders used to simulate 2D shapes. The methods used for drawing these views were created alongside the preexisting methods for creating the 3D tree renderings within the Tree class. Which set of methods were called depends on a variable in the View class, which is set by controls manipulated by the user in the interface.

During the development of the 2D viewing modes we tested the coordinate system of the program more rigorously. We received a larger and more comprehensive set of test data that was used to tweak certain aspects of how the program calculated each tree's position and shape. By comparing the output of the program to the existing hand drawn profiles for that test data, we were able to verify that the program would deliver the same output as the hand drawing method given the same data. However, this required some of the raw data for tree positions to be converted so that it would work with the program's coordinate system, as the original data lists each tree's coordinates relative to the module it is in, while the program uses absolute coordinates and does not take the module into account. While this makes more work for the user entering the data, it is also more flexible as not all institutions use modules when recording tree data. Theoretically, any institution could use this bv simply converting their program coordinate system to the absolute one used by the program. Also, in the future one of the goals for this program is to allow it to accommodate multiple data entry schemes, so that the type of coordinates used by Hiram College (i.e., CVS protocol) could be entered without modification as long as the corresponding data format is selected as the data is being loaded.

Due to the nature of how VirtuaForest renders trees, it does not represent the architecture of individual trees with the same level of detail that can be included in hand-drawn profiles (Figure 2.A) in the vertical profile rendering. While that information may be useful for certain forest ecology investigations, its applicability depends on the aim of the study. In Hiram College's forest monitoring efforts tree architecture is not essential; coarse field sketches are made for general characterization of tree architecture in hand-drawn vertical profiles, but no additional data such as detailed branching patterns are collected. absence of tree-architecture Thus, the information in this program does not negatively affect its usefulness for our monitoring program. In fact, the program was developed with our monitoring needs in mind; its functions either currently meet those needs, or have the capability of being expanded to address them in the near future.

Other features added to the 2D viewing modes included a series of radiating lines in the top down mode that stretched from the center of the trunk to the outer edges of the crown. This fixed one of the problems with the hand drawn top down view, which was that when multiple trees had overlapping crowns, it was often difficult to tell which trunk belonged to which crown, and because the species label was written next to the trunk, it was also difficult to determine the exact species of each tree without consulting the data. The radiating lines made it immediately clear which crown belonged to which trunk, even when many crowns were overlapping each other (see Figure 5).



Figure 5: Detail of top down view demonstrating how radial lines indicate which trunk belongs to which crown

To further indicate the species of each tree, we added a set of rendering attributes to the Species class that could be edited by the user. The user could select a species from the list of all tree species found within the given data set, and for each one alter properties such as line thickness, line pattern, and fill color. We hope to add the ability to save species appearance configurations, which would allow a single appearance standard to be copied and used throughout an entire institution, which would minimize the confusion regarding how individual species of tree are displayed.

Finally, We added the species acronym display to the top down and side views, causing the first two letters of each species two-word name to be displayed next to the trunk of any tree of that species. The species acronym can also be edited as part of the species appearance, which allows the user to avoid having two different species whose names both start with the same two letters to be assigned identical acronyms.

5. FUTURE WORK

Although the most basic features of the Tree Rendering Program are complete, there is still a great deal we would like to add. As the project went on, we focused more on the 2D views, so there are many improvements that could be made to the 3D viewing mode, which at the moment is rather rudimentary. In general, all viewing modes could benefit from graphical enhancement designed to make the trees more realistic looking. One feature that would make the program more useful is the ability to save a particular view of the tree graph as an image. This would greatly assist researchers who wish to include illustrations of their data in papers and posters. More generally, many improvements could be made to the user interface, both aesthetic and functional, that would make it more convenient and user friendly, while leaving the basic functionality of the program intact.

Data on the same plot of trees is collected periodically. In future work we plan to implement an animation of this periodical data. We also plan on creating basic statistical tools to aid the researcher in finding patterns among the growing trees.

With this project, we have created a basic tool for visualizing plots of trees. While there is still much to be done we feel the program is to a point where others can use it and provide us with feedback and suggestions for improvement.

The tool is available for download from:

http://cs.hiram.edu/~oliphantlt/tree_program.

6. REFERENCES

[1]McGaughey, R. J. Visualizing forest stand dynamics using the stand visualization system. *Proceedings of the 1997 ACSM/ASPRS Annual Convention and Exposition*. April 1997.

[2] Dunbar, M. D., Moskal, L. M., and Jakubauskas, M. E. 3D Visualization for the Analysis of Forest Cover Change. *Geocarto International*, Vol. 19, Issue 2, 2008.

[3] Yu, Q., Chen, C., Pan, Z., and Li J. A GIS-Based Forest Visual Simulation System. *Proceedings of the International Conference on Image and Graphics*, 2004.

[4] Sumerling, G. Lidar Analysis in ArcGIS 10 for Forestry Applications, January 2011

[5] Newton, A. C. *Forest Ecology and Conservation: A Handbook of Techniques.* Oxford University Press, New York, 2008.

[6] Oliver, C. D. and B. C. Larson. *Forest Stand Dynamics*. John Wiley and Sons, Inc., New York, 1996.

[7] Benedict, M., Matthew Hils, Brittany Blum, Olivia Hess, Jemika Kastee, Timothy Luttermoser, Regeant Panday. Long-Term Forest Monitoring at James H. Barrow Field Station, Hiram, Ohio. Ohio Natural History Conference, Ohio Biological Survey, 2011.

[8] Peet, R.K., T.R. Wentworth and P.S. White. 1998. A flexible, multipurpose method for recording vegetation composition and structure. Castanea 63:262-274

[9] The Carolina Vegetation Survey. Sampling methodology and data management. http://cvs.bio.unc.edu/methods.htm, undated.



Figure 6: The VirtualForest user interface, including plot display, height/diameter filters, and editable plot data table