A SYSTEM FOR DRAWING SYNTHETIC IMAGES OF FORESTED LANDSCAPES

Timothy P. McDonald'

Abstract-A software package for drawing images of forested landscapes was developed. Programs included in the system convert topographic and stand polygon information output from a GIS into a form that can be read by a general-purpose ray-tracing renderer. Other programs generate definitions for surface features, mainly trees but ground surface textural properties as well. The package can be used to design logging cut unit boundaries that are less visible from a given viewpoint. Images created using the system may also be suitable for showing the public how a given prescription might appear if implemented. The package was used to make images showing the potential for using strip clearcutting as a means of mitigating the visual impact of harvests on steeply sloping ground. An analysis of the package indicated that there was a potential to improve its data handling and user interface components.

INTRODUCTION

Appearance is the most convincing source of information available to the public in evaluating the status of a landscape (Nassauer 1992). Silvicultural treatments on a landscape can leave unintended visual impacts that persist for a number of years. These impacts can often be avoided or reduced by careful planning but it is difficult to incorporate consideration of the visual effects of a prescription into the planning process. This has led to the development of software tools that give feedback to trcatmenr designers on how decisions might affect the appearance of a landscape. Pictures derived from these tools have been found to be useful in both planning prescriptions, and in gauging the public's reaction to a proposed treatment before it is implemented (Bergen and others 1992).

Perhaps the most succesful computer terrain visualization tool developed to date has been Vantage Point (Fridley and others 1991), developed at the University of Washington Cooperative for Forest-Systems Engineering. Vantage Point is a highly integrated environment for creating visual simulations of forested landscapes. It has the capability to incorporate topographic and stand-related information into a rendering system to produce high-quality images showing landform, trees, roads, and cut areas. The software also has domain-specific knowledge of how trees grow and can use this to generate time-sequenced images of a series of treatments.

Although Vantage Point is a powerful design tool, it is currently available only for a specific workstation-class platform. Other tools are more widely available. SmartForest, for example, developed at the University of Illinois Landscape Architecture Department's Imaging Lab, runs on a variety of Unix-based workstations. UVIEW is PC-based and is a geographic analysis software package developed by the USDA Forest Service that includes some landscape rendering capabilities. At present, however, these alternatives are designed primarily for purposes other than landscape visualization and their output quality cannot duplicate that of Vantage Point.

The need exists for a visualization tool that provides output renderings of acceptable quality, and that is available for use on a variety of hardware/operating system platforms. Such a system should be inexpensive, be flexible enough to interface with any number of data sources, particularly GIS, and should be relatively simple to operate. This paper reports on the development of such a system that, although it does not duplicate the full functionality of Vantage Point, it does produce high-quality renderings of forested landscapes.

DESIGN OBJECTIVES

The visualization system was conceived as a means of achieving two related functions: (1) to provide feedback to logging engineers on the visual impacts of cutblock boundary changes when designing treatments, and (2) to create images that could accurately convey to the public' the results of the engineer's design decisions. As a design tool, producing images should be fast and simple enough so that a sufficient number of alternatives can be evaluated within a treatment design cycle. The system should also interface smoothly with a GIS to easily access topographic, unit boundary, and stand data. As a tool for informing the public on treatment decisions, output images should be realistic with respect to boundary placement, residual stand density, landform, and other visual cues an observer might use in judging the scenic beauty of a landscape.

The primary development objective for the system was that it should be reasonably simple to use and available for as wide an array of computer platforms as possible. This required designing to a "least common denominator" and development efforts therefore concentrated mainly on functional aspects as opposed to user interface concerns. Although the user interface was not considered a primary feature of the system, the various components were kept fairly consistent in their usage. Finally, because there was a great deal of previously developed software freely available, it was decided to use public domain programs to as great an extent as possible.

[1] Research Engineer, USDA Forest Service, Southern Research Station, Auburn, AL

SYSTEM DEVELOPMENT

From a software development standpoint, the most difficult part of creating a landscape viewing system is building the rendering engine, the program that calculates the interaction between "lights" and "objects" in a virtual "world," and creates a depiction of the scene based on its results. Fortunately, several rendering systems have been developed and are in the public domain. Of those available, the POVRAY ray tracer is available for nearly every computer platform that exists, is actively used by thousands worldwide, and is under continuous development by a group of very talented people. POVRAY is a powerful ray tracer that uses a very general scripting language as its input, includes many standard and prebuilt objects and textures, can model a large set of graphics primitives and do constructive solid geometry with most (for example, do intersections or unions between two objects), and can model atmospheric attenuation and particle-induced scattering (haze, smoke, or fog). Because of its flexibility, power, and availability, POVRAY was chosen as the rendering engine of our landscape visualization system. The remaining components of the system were designed to provide the data necessary to drive the rendering engine and included tools for terrain and object (for this application, tree) modeling.

Terrain data is available in many digital forms, most commonly as U.S. Geological Survey Digital Elevation Models (OEM's). A DEM is a matrix of elevations arranged in row echelon, and is sometimes referred to as a heightfield. Heightfields are discrete in nature, but for locating objects on its surface we need a method of interpolating between the grid points. There are, again, multiple means of interpolating heightfields, but we chose the triangulated irregular network (or TIN) as our preferred method. TIN's can significantly reduce the amount of data required to represent smooth surfaces, they are continuous (except along triangle intersection lines) and use planar facets, making it simple to calculate surface object locations. Since most topography data exists as DEM's, however, a method was needed to convert this type of data to a TIN. Again this problem has been solved and the tools placed in the public domain. A package known as scape, developed by Paul Heckbert and Michael Garland at Carnegie Mellon University, was chosen to fulfill this task in our system (Garland and Heckbert 1995).

Rendering trees requires some notion of where on the landscape trees should and should not be, and what they should look like. Our contributions to the visualization package were in developing tools to integrate the various components, and in developing methods for efficiently drawing and placing trees on the landscape. Two assumptions were made concerning tree locations in developing the methods: (1) trees were assumed to be objects uniformly distributed within any given polygon, and (2) locations of individual trees were assumed to be independent of all others regardless of species or age. With this model, only a density and polygon boundary are necessary to generate object locations. Assigning an object type (tree, for example) and object characteristics (species and age) to locations is done separately and can make use of positional information in classifying them. Species may vary with elevation, for example, and methods are provided to assign characteristics based on height above a reference plane.

SYSTEM USAGE

Table 1 lists a sequence of steps necessary to create a rendered image of a landscape, and the programs used in each step. The system is intended to be used in conjunction with a GIS for maintaining stand, DEM, and polygon data. We are currently using the GRASS GIS for data storage, and translation utilities have been developed to convert output data into intermediate formats understood by the system programs. Using another GIS would require development of a similar set of utilities.

DISCUSSION

The visualization package was designed to meet two particular goals: that it provide a design tool for logging engineers to evaluate prescriptions for their visual impacts; and that it produce images that accurately convey a sense of how a given prescription would look if implemented. Testing of the system by outside users has not been

Table I-Steps and tools used in creating a rendered landscape image

Step	Software	Input requirements	output
1. Data assembly	GIS		DEM, view points, stand polygons, tree species and age info
2. Create TIN	scape	raster DEM	List of triangle vertices
3. Add vertex normals	normcalc	TIN	TIN with vertex normals, in intermediate format
4. Add trees	lt cutpoly	TIN, stand polygons, # of trees	List of x, y, z coordinates tree locations
5. Convert TIN to POV-Ray	gs2pov	TIN, surface texture Info	POV-Ray object with textures assigned to triangles
6. Assign species, age to trees	trees2pov	Tree types, sizes	A list of "tree" objects for inclusion in POV-RAY
7. Render	POV-Ray	Control script, tree definitions	24-bit color image of landscape

conducted so it is difficult to draw conclusions on how well the system meets these goals. Some preliminary observations based on a trial study by our research unit, however, indicate that it can potentially do both given that the user is aware of specific problems.

Data from a study on alternative silvicultural prescriptions in upland hardwood management were used as a test case. An objective of the study was to measure the difference in visual impact due to prescription based on viewer preferences. Because of space constraints. the prescriptions could not be implemented on a large scale. It was felt that viewer response to the actual treatment plots would not accurately reflect their feelings about large-scale implementation of the prescriptions, so the visualization system was used to produce images showing the treatments applied across a large hillside.

The study site was in northern Alabama near the Bankhead National Forest. A sequence of images was created to duplicate the view from an observation point across a narrow valley looking toward the hillslope where the treatments were installed. Figures 1 through 3 are a series of images created using the visualization system and show the hillslope in an uncut state. with a 72-acre clearcut. and with 43 acres removed as three 150 ft-wide strips following the terrain contour.

Suitability as a Design Tool

Based on experience in developing the images in figures 1 through 3, problem areas were identified that might limit the utility of the visualization system. One was the lack of true integration between the system components. Although this was not generally a problem, lack of integration forces the user to be familiar with the workings of several independent programs and the data formats, and idiosyncracies. of each. An example of this type of problem was the inconsistency in how scaling of data was handled by the various system programs. For example, the TIN generation tool takes as input a DEM that is a form of raster. A raster format is assumed to be on a uniform grid and, therefore, no x,y coordinate data are included. This means. however, that scaling information is lost-the output TIN covers an mXn grid, where m and n are the number of rows and columns of the input DEM. This shortcoming is easily fixed by simply multiplying the output TIN x and y coordinates by the DEM scale. It does illustrate, however, the degree of familiarity with the system that is required, and the need for the user to direct the flow of correct information between the system components.

Generating images using the system was a data-intensive process. For the 4 alternatives evaluated in our study, a total of 11 data files were needed: a control script to drive the renderer (~0.6K bytes in size), a file defining how trees were to be drawn (1.2K), plus images of trees (in our case, 6, each about 12K), the topography TIN (600K plus foreground and cut area texture image files, each about 15K), and lists of tree locations (unique to each image, ranging from 3970K to 4540K in size—49,759 to 56,850 trees). Although the system required about 5000K of data

Figure l-Example application of the visualization system. Rendered view of a hillside in an uncut state.

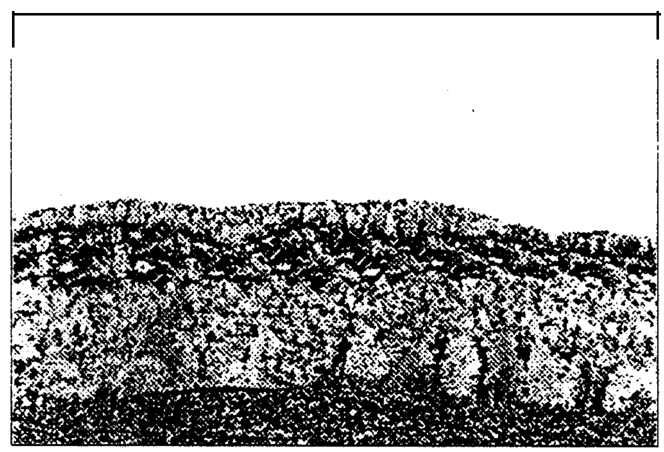

Figure 2-Identical view as in figure 1, but with a 72-acre clearcut imposed.

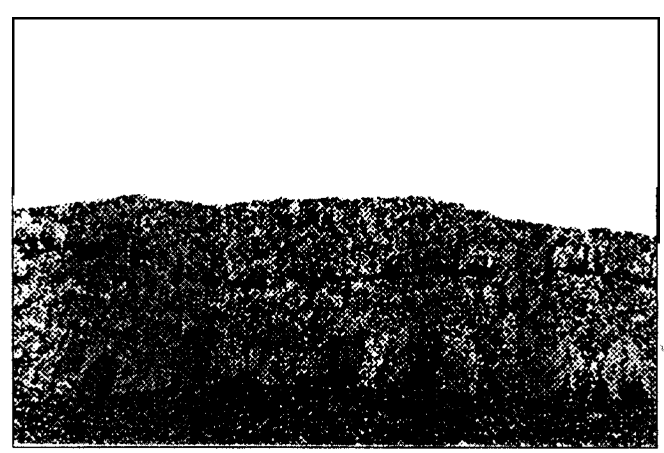

Figure 3-Identical view as in figure 1, but with a 43-acre strip clearcut imposed.

to create a single image, no integrated tools were provided to manage it. Again, this was not a limiting problem, but required methodical documentation of the steps taken to keep track of file locations.

The use of a general-purpose rendering engine meant that there was a significant amount of time spent in learning to use it. A degree of familiarity with the syntax and capabilities of POVRAY is needed to create the script file defining viewpoints, camera direction, lighting, and other object characteristics. This can only be gained from reading the large amount of documentation supplied with POVRAY, and by actually using it.

Creating realistic trees was the most difficult part of using POVRAY. Trees themselves are complex 3D objects, and it was difficult to capture that complexity in such a way that enough trees could be included in the scenes without exceeding the memory constraints of the computer system. Initially, trees were modeled as cylindrical trunks with'a 3D polygonal canopy. It was found, however, that only about 3,000 of these trees could be placed in the scene before the virtual memory capacity of our workstation was exceeded. The circumstances in our test case required about 50,000 trees be included, so another type of tree had to be used. Our solution to this problem was a transparent, zero-thickness box with a picture of a tree painted on it. POVRAY had no problem handling 57,000 of these "flat" trees (-47M bytes virtual memory peak usage), and they appeared as visually complex as a real tree as long as they were viewed from an adequate distance. Images with trees in the foreground, however, might require a different approach.

As a design tool, application of the visualization system was constrained primarily by the familiarity of the user with POVRAY. Once the details of the rendering process were worked out, time on the part of the user to create an image was on the order of a few hours from start to finish. If the design goal is selecting among alternative cut unit boundaries, the image creation process is greatly simplified after the first, and the time per picture is reduced to essentially the time to render. Rendering times for an image like those in the figures averaged a little over 60 minutes on a SUN workstation at 600 x 400 resolution. Increasing resolution had the most dramatic effect on rendering times-a 2048 x 1536 image took 9 hours 36 minutes to complete. Adding atmospheric scattering effects increased rendering times by a factor of 3.

Quality of the Images

Bergen and others (1993) compared observer response to computer-generated and photographic images of a particular viewshed. Their results indicated that, as long as elements that are important in forming an opinion about a scene are preserved, computer-generated images are suitable as a basis for gauging public opinion about a silvicultural treatment. The most important visual cue in these images was topography. Examination of the rendered images and several photos taken from observation points showed that there were some small differences between the generated terrain and the actual. The differences were mainly visible along the ridge top, with the rendered scene showing more relief than was truly present. This was likely due to inaccuracies in the input DEM, but this has not been verified. Accuracy of the trees in the rendered images could also be improved. As stated before, trees were "painted" on transparent boxes to create the scenes. The tree images used in the process were more like clip-art, and greater realism in the final renderings would be possible if scanned photos of canopy-grown hardwoods were used.

LITERATURE CITED

Bergen, S.D.; Ulbricht, C.A.; Fridley, J.L.; Ganter, M.A. 1932. Methods and tools for incorporating visual analysis into the forest harvest planning process. 1992 international winter meeting sponsored by the American Society of Agricultural Engineers. ASAE Paper 927517. St. Joseph, MI: American Society of Agricultural Engineers.

Bergen, S.D.; Ulbricht, C.A.; Fridley, J.L.; Ganter, M.A. 1993. The validity of computer generated graphic images of forest landscapes. 1993 international winter meeting sponsored by the American Society of Agricultural Engineers. ASAE Paper 937501. St. Joseph, MI; American Society of Agricultural Engineers. 16 p.

Fridley, J.L.; McGaughey, R.J.; Lee, F.E. 1991. Visualizing engineering design alternatives on forest landscapes. 1991 international winter meeting sponsored by the American Society of Agricultural Engineers. ASAE Paper 917523. St. Joseph, MI; American Society of Agricultural Engineers. 7 p.

Garland, Michael; Heckbert, Paul S. 1995. Fast polygonal approximation of terrains and height fields. Tech. Rep. CMU-CS-91-181. Pittsburgh, PA: Computer Science Department, Carnegie Mellon University.

Nassauer, Joan Iverson. 1992. The **appearance** of ecological systems as a matter of policy. Landscape Ecology. 6(4): 239-250.