Implementing SmartForest forest visualization tool on PC environment

Uusitalo, J.[1] & Kivinen, V.-P.[2]

[1] Faculty of Forestry, University of Joensuu, BOX 111, FIN-80101 Joensuu, Finland, jori.uusitalo@forest.joensuu.fi
[2] The Forest Visualization Laboratory, Department of Forest Resource Management, University of Helsinki, BOX 24, FIN-000014 University of Helsinki, Finland, veli.kivinen@helsinki.fi

Abstract

A forest visualization tool can be implemented on a large scale only when it can be integrated with other forest planning systems such as growth simulators, GIS, etc. Since the forest management systems in Finland are mainly running on the PC environment, there will be a growing need for advanced PC-based forest visualization tools. Concurrently, PCs equipped with appropriate graphic accelerators are gradually achieving the efficiency needed to run advanced visualization software. The Forest Visualization Laboratory at the University of Helsinki has an ongoing project intended to convert SmartForest, originally developed at the Imaging Systems Laboratory, University of Illinois, to the PC environment. This paper describes the PC version of SmartForest and the process of converting UNIX-based into PC-based SmartForest.

1 Introduction

The capacity to apply data visualization in forestry is improving rapidly. Escalation in computing power, development of computer graphics and image editing tools enables construction of forest visualization or even virtual forest with sufficient fidelity has been brought within the constraints of almost any university research project or commercial application. Today, personal computers (PCs) equipped with appropriate graphic accelerators are gradually achieving the efficiency needed to run advanced visualization software. As the graphic performance of PCs accelerates, it is obvious that more complicated forest visualization systems, even virtual reality (VR) solutions, will be integrated with existing forest management tools and spatial databases.

The potential of applying data visualization to various forest management tasks has already been demonstrated in numerous papers. Forest visualization can play an important role in assessing the effects of pests (Orland et al. 1990), evaluating the visual effects of various forest harvest practices (Heasley & McNamara 1990, Orland et al. 1992, Thuresson et al. 1996), judging the visual quality of the landscape, assisting wood procurement managers in industry to buy stands that meet market needs (Uusitalo et al. 1997). As stated earlier (Uusitalo et al. 1997) a forest visualization tool can be implemented on a large scale only when it can be integrated with other forest planning systems such as growth simulators, GIS, etc. Since the forest management systems in Finland are mainly running on the PC environment, there will be a growing need for advanced PC-based forest visualization tools.

SmartForest, developed at the Imaging Systems Laboratory, University of Illinois, meets the requirements needed for an advanced forest visualization tool in many respects. The user can view forest areas from any user-defined location, “walk” between the trees, assess the visual character of the forest with realistic tree and ground textures and interact with the underlying data. SmartForest, written in C and using OpenGL library and X-windows was originally been developed for use only in computers using the UNIX operating system. The program and it’s origins are described elsewhere (Orland 1994, Uusitalo, Orland et al. 1997).This paper describes the PC version of SmartForest and the process of converting UNIX-based into PC-based SmartForest.

2 SmartForest features

SmartForest enables a variety of ways to move in a forest setting. In most cases the user moves the cursor within the view while holding different mouse buttons to achieve longitudinal, rotational, and vertical movement. Having reached a new position, each of the location and height parameters can be adjusted further by either more keyboard input or by dragging slide bars. The location may also be defined by a separate map window (Figure 1) by clicking the desired viewing point. The map window, overlaid either by shaded relief or stand polygons, facilitates a quick and easy orientation to the forest by showing the location and the direction of view with an arrow-like indicator.

To enhance interaction speed, a number of compromises have been made. First, the program displays a reduced number of trees while moving. On ceasing movement the remaining trees are drawn. Another means of enhancing interaction speed is to reduce the “horizon” in the image. A less extensive landscape will draw more rapidly, allowing faster interaction until a point of interest is reached at which time the horizon can be extended to give a complete landscape-scale view.

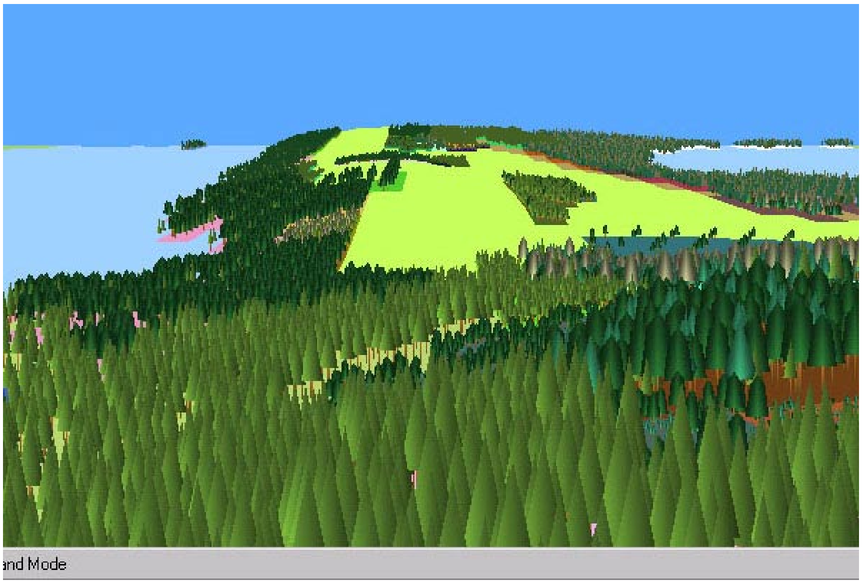

Fig. 1. The SmartForest Management mode. Trees are presented as simplified icons to faciliate quick analyses.

SmartForest comprises two different modes: management mode and landscape mode. Management mode is a simplified presentation of the real forest conditions that helps the manager quickly and efficiently query and analyze the various characteristics of the forest stands and single trees (Fig. 1). Reduction of the realism enhances the moving capabilities, speeds the analyses and enables the user to observe detailed information about large forest areas at one view. Landscape mode is a realistic one-view presentation of the real world. Trees are presented as texture-mapped objects and the ground is wrapped with realistic 2-D ground images. Landscape mode facilitates the evaluation of the visual effects of different forest harvest practices. The desire for realism reduces the speed, which however should not be a problem, since landscape analysis is usually restricted to a couple of crucial viewpoints (Fig. 2.)

SmartForest can be used to visually classify trees according to many different tree characteristics. This color-classification enables the manager to efficiently envision the especially advantageous characteristics of a stand or forest holding. For procurement officers and mill-owners these features may be critical in determining the feasibility of particular purchasing or harvesting plans.

The strength of SmartForest lies in its capability for providing a tool for several types of forest management task. One of the major recent developments has been improvement

Fig. 2. The SmartForest Landscape mode. Trees are presented as texture-mapped objects and ground is wrapped with realistic ground textures.

of texture mapping, allowing for better landscape management. The program is now able to vary ground textures by site type. Moreover, to eliminate graininess of the ground textures appearing in front of the viewer, the size of the texture maps of each 30 x 30 m grid has had to increase to 512 x 512 pixels. This has, of course, markedly increased the response times, causing a lot of frustration with large horizon views. One way of speeding up the response times may be using the same texture map with different pixel sizes depending on the distance between viewpoint and each texture map. In any case, the latest improvements have markedly increased the fidelity of views, facilitating judgement of the visual quality of landscape.

3 System development tools

The SmartForest system comprises three types of program element:

- Graphical user interface programmed in Microsoft Visual C++
- Routines for file operations, navigation and visualization written in C
- 3D graphics by OpenGL

Microsoft Visual C++ is one of the most popular development environments for 32-bit Windows platforms (Win32), including Windows NT and Windows 95. Writing applications with the Visual C++ system is based on the Microsoft Foundation Class

Library (MFC) and on several fully integrated Windows-hosted development tools such as wizards and resource editors. The class library is a set of C++ classes that make up an application framework by encapsulating a large portion of the Win32 application programming interface (API). The Visual C++ tools enable programmers to visually design resource-based user interface elements and to easily connect these elements to code.

The message handlers of the application framework make calls to functions for navigation, visualization and file handling. All these functions are implemented in C since most of them have been ported from the original Unix-based version of SmartForest written in standard C language.

SmartForest is an OpenGL application. That is, SmartForest uses the OpenGL graphics and modeling library to draw a 3D forest environment. OpenGL is an industry standard 3D graphics API developed and optimized by Silicon Graphics, Inc. (SGI), and supported and further developed by many vendors, including Microsoft, IBM and Digital. OpenGL contains approximately 120 commands and functions that are used to draw graphics primitives such as points, lines and polygons in three dimensions. In addition, OpenGL supports lighting, shading, texture mapping and other special effects such as blending and fog. OpenGL is used for a variety of purposes, from CAD engineering and architectural applications to highly complex flight simulators.

4 Conversion process

The process of porting SmartForest from a Unix platform into the PC/Windows environment was facilitated by two important facts. First, the Unix version is written in C, an extremely portable programming language compatible with most environments for Windows development, including C++ and 4GLs (‘fourth-generation languages’) like Delphi and Visual Basic. Second, the original source code consists of several separate modules each of which is responsible for a certain task such as reading data inputs, navigating through the forest and drawing the scene as texture-mapped. This modular structure made it much easier to understand and reuse the code.

The conversion process was started by building an SDI (Single Document Interface) shell application with AppWizard, a code generator included in Visual C++. Though the skeleton application created by AppWizard is a full-fledged Windows program, it does not automatically support OpenGL graphics. In order to run OpenGL under Windows, the initial AppWizard shell application had to be provided with the mechanisms that associate OpenGL drawing commands to a Windows window (rendering context, pixel format, etc.).

In the second phase, the code for handling information inputs (an elevation map file, a stand file and tree list files) and initializing data structures was customized for the PC environment. Each relevant source code file was examined row by row and all the bugs and illogicalities discovered were fixed, and the X-Windows/Motif-related code blocks were eliminated or replaced with Windows-specific code. At the same time, the code was generally refined by improving the support for handling anomalous situations, known as ‘exceptions’, and by building procedures for processing program arguments.

Since the X-Window System differs completely from MS Windows, the whole graphical user interface of SmartForest had to be reconstructed for the PC environment. In the PC version, the Forest Window, the main window of SmartForest, is a normal frame window created by AppWizard while all the other windows, including the Map Window and the Control Panel Window, are either modal or modeless dialogs. Unlike the Unix version, dialog boxes, menus, icons, strings and other UI components are not created programmatically during the execution of the SmartForest PC version but are all resources loaded from the executable file into the memory only when those components are used for the first time. Using resources saves system memory, simplifies the maintenance of the program and helps develop different language versions from the program.

In the final phase of the conversion process, the functions for building a 3D forest scene and handling user interactions (e.g. sliding the mouse to move through the forest) were cleaned and localized into PC architecture and then linked together with the rest of the application. The result of the linkage was the pilot version of SmartForest that runs on both the Windows NT and Windows 95 operating systems.

5 PC vs. Unix in speed

After the initial tests, all the presumptions that the PC runs advanced visualization software in efficiently have to be revised. The PC version seems to be even faster than its predecessor written for Unix although some code has been reprogrammed more efficiently. Response times with texture-mapped forest seem to be basically dependent on the quality of the graphic accelerator and texture buffer. Tests with a typical PC with a P233 MMX processor, 64 Mb RAM and Matrox Mystique graphic accelerator with OpenGL support provided response times equal to the cheapest UNIX-computers with support for texture mapping, but 10–100 times faster than to similar class UNIX-computers without support for texture mapping. In a typical case rendering the forest with our P233-equipped PC varies from a couple of seconds (horizon 3 %) to some minutes (horizon 100 %).

6 Conclusions

Although new technologies are capable of bringing forest visualization or even virtual forest applications onto forest managers’ desks, there are still several obstacles to be removed. We still have insufficient knowledge about the advantages and disadvantages that the use of these advanced technologies could provide. Consequently, proper validity and utility analyses of forest visualization should follow development of forest visualization techniques. This will give us further vital feedback for future development.

Acknowledgements

We wish to thank prof. Brian Orland and his laboratory for guidance, encouragement anf technical backing during this work as well as Dr. Roderick McConchie for proofreading our English.

References

Heasley, J. & McNamara, J. 1990. Simulating the Visual Effects of Timber Harvest Practices. In: Proc. Resource Technology 90 – Second Symposium on Advanced Technology in Natural Resource Management. Washington, D.C. p. 39–47.

Orland, B. 1994. SmartForest: a 3-D interactive forest visualization and analysis system. In: Proc. Decision Support -2001. Combined events of the 17th annual Geograpic Information Seminar and the Resource Technology '94 Symposium. Toronto. p. 181–190.

Orland, B., Daniel, T.C., Lynch, A.M. and Holsten, E.H. 1992. Data-driven Visual Simualtion of alternative Furture for Forested Landscapes. In: Proc. IUFRO – Integrating Forest Information over Space and Time. Canberra. p. 368–378.

Orland, B., LaFontaine, J. & Daniel, T.C. 1990. Alternative Futures for Forested Landscapes. In: Proc. Resource Technology 90 - Second International Symposium on Advanced Technology in Natural Resource Management. Washington, DC. p. 48–57.

Thuresson, T., Näsholm, B., Holm, S. & Hagner, O. 1996. Using digital image projections to visualize forest landscape changes due to management activities and forest growth. Environmental Management. 20(1): 35–40.

Uusitalo, J., Orland, B. & Liu, K. 1997. Developing a forest manager interface to SmartForest. In: Proc. GIS '97. Vancouver, February 17–20, 1997. p. 231–235.

Uusitalo, J., Orland, B. & Liu, K. 1997. A forest visualization interface for harvest planning. In: Proc. ACSM/ASPRS/RT Annual Convention. Seattle, April 7–10, 1997. p. 204–215.