ICAT 2003
December 3-5, Tokyo, JAPAN

Interactive 3-D Visualization of Real Forests

Qizhi Yu[1),2)] Chongcheng Chen[2)] Zhigeng Pan[1)] Tianhe Chi[2)]

1) State Key Laboratory of CAD & CG, Zhejiang University, Hangzhou 310027, China
2) Spatial Information Research Center of Fujian Province, Fuzhou 350002, China

Abstract

Forest resource management systems and forest landscape visualization applications often need user-steered interactive displays of a forest landscape representing the underlying forest inventory database. We present an approach for dynamically modeling and real-time rendering of a real forest. A parameterized procedural tree model is given with a hybrid representation method and a specific level of detail algorithm for ensuring real-time frame rates. Forest inventory data, together with predefined tree parameter templates, drive the modeling engine to generate visual realistic 3-D tree models on the fly. We have evaluated our approach by applying it in a GIS-based forest resource management system to construct a virtual forest with immersive capability for navigation, query, and analysis.

Key words: Forest Landscape Visualization, Procedural Model, Plant Model, Levels of Detail, Virtual Reality

1. Introduction

In the field of forest resource management, the creation of visual tools of high quality represents a very important scientific and technical issue. 3-D visualization of forest landscape has been increasingly used in planning processes as means of communication between planners, clients and the public, in particular for the discussion of visual impacts of proposed changes in the landscape [6]. Our research work seeks both an appropriate modeling method and real-time rendering performance in data-driven forest visualization applications.

Forest landscape visualization system for decision-making support must construct the scenes effectively and accurately faithful to forest inventory data that are frequently changeable as the result of forest operations, growth, or fires *etc*. Therefore, dynamically modeling of trees is necessary for constructing forest scenes as automatically as possible. Moreover, sizes of the dynamically created tree geometric models must match those found in forest inventory database.

On the other hand, an essential way of experiencing landscape is moving through the landscape and perceiving it from different angles. Ideally, such a walkthrough needs to maintain a frame rate of at least 20 frames per second to avoid jerkiness. As it takes roughly hundred thousand triangles to build a convincing model of a single tree, an efficient representation method and a simplification algorithm must be applied to reduce the polygon count for a given frame to a reasonable level.

In this paper, we present a data-driven visualization system for forest scenes. Our contributions are the introduction of an appropriate method for dynamically creating visually realistic representation of individual trees in accordance with the size information found in underlying forest inventory database, an efficient representation of 3-D tree model and a specific simplification method suitable for real-time rendering.

At run-time, we transform forest inventory data into a set of tree parameters by modifying predefined parameter template of some species. Then, the new parameters drive the parameterized procedural model to generate a 3-D tree model maintaining the features of the species and matching given sizes including tree height, crown width, and DBH (Diameter at Breast Height).

Our solution for the interactive visualization of forests combines two methods of real-time rendering: the texture mapping and multi-resolution modeling techniques. To greatly reduce the number of polygons, a quadric texture map is used to represent a cluster of leaves. Level-of-detail is obtained by reducing the displayed number of leaves and stems and the section resolution of stems.

The remainder of this paper is organized as follows. After reviewing relevant previous work, in Section 3, we develop our tree model and show how to produce a multiresolution representation. Next section presents the details of modeling and rendering of a real forest. In Section 5, we show an application and results of our approach. We conclude with some discussion about future work in Section 6.

2. Related Work

2.1 Modeling and Rendering of Trees

Plant modeling has been widely investigated. Due to their complexity, plant models in computer graphics are commonly created using procedural methods, which generate intricate branching structure with a limited user input.

Several simulation-based procedural models have gained very nature tree structures [19,16]. While simulation-based model has advantages in modeling and simulation of development for biological purposes, [17] pointed out its several drawbacks one of which is that global characteristics of plant appearance are difficult to control.

Along a different way, a number of methods have been proposed for generating plants with the goal of visual quality without relying on botanical simulation [2, 8, 18, 22]. Weber and Penn [22] presented a parameterized procedure model that emphasized the overall geometrical structure of the tree. This class of approach provides a more direct and intuitive control of visually important aspects of plant form, and therefore is preferable in applications where visual output is of primary importance.

Many efforts have been put into efficiently rendering of trees. Oppenheimer [12] used polygonal cylinders to render large branches, replacing the small ones with lines. Weber and Penn [22] proposed a similar simplification approach by replacing cylindrical branch segments with lines, and leaf polygons with points or not rendering them at all. Deussen et al. [4] recently showed how rendering with points and lines could be used to render the plants in fairly complex outdoor scenes in real time.

The main advantage of using image-based rendering methods over 3D models is that it is computationally less expensive and rendering time is usually constant and not dependent on object complexity. Thus, higher frame rates can be attained. Several texture-based methods for reducing the geometric complexity of trees have been proposed. Max and Ohsaki [10] rendered trees from pre-rendered multiple-layer z-buffers. Jakulin [5] approximated a tree with slicings, each consiting of multiple layers of textured polygons. Meyer et al. [11] described a method for rendering trees based on blending of pre-computed hierarchical impostors.

If geometry is required, it is necessary to use multiresolution modeling techniques that reduce the polygons sent to the graphic system. Lluch [9] proposed a procedural multiresolution representation for plant and tree based on parametric L-systems. By introducing a metric that quantifies the ramification of the branches of the tree, the algorithm could produce LODs that preserve the visual structure of the tree. However, their work does not include simplification methods for leaves. Remolar et al. [20] proposed a foliage simplification algorithm that collapses two leaves at each iteration and obtains a new one, maintaining a similar area to that of the pair of collapsed leaves.

2.2 Data-driven Forest Visualization Systems

To visualize forest landscape accurately, a number of forest landscape visualization systems have been implemented which are based on the use of map information, forest inventory database and digital elevation model (DEM) [13,15,21]. Generating tree geometry model according to field data is an important and challenging problem in such systems. While many systems [7,15] adopt simulation-based procedural models, this approach can represent trees in different development phases but fail to accurately depict the size information of trees given in inventory data. Another method is to use descriptive models that capture plant architecture without simulating the underlying developmental process. When a family of geometric models is constructed to capture the key "postures" of a plant at different ages and with different high-level characteristics, we can obtain in-between geometries by interpolation [3]. Nevertheless, pre-design of lots of models is necessary but time consuming.

Interactive visualization has also been pursued in such systems by sacrificing visual reality. Smart Forest [13] allowed flexible real-time movement in virtual forest landscape. However, this ability is in effective use only in management mode, which illustrates landscape with simple geometric tree objects. To achieve a sufficient rendering speed, Lim [7] expressed trees by 2D plant images generated by a simulation-based model.

3 Individual Trees

3.1 Tree Structure Modeling

In this paper, the tree model is composed of a main trunk, several levels of branches, and leaves. We use a set of parameters to depict the structure and appearance characteristics, which have much in common with the method in [22]. Enlightened by [17], we extend the work in [22] by introducing parameterized curve to depict morphogenetic gradients that describe the distributions of features along the axes. Experiences show that curves are more intuitive and flexible than mathematical equations used in [22].

For trunk, the parameters mainly include length, maximum diameter, branching pattern, and distribution and quantity of first level of branches. Branch parameters are grouped according to their level. Parameters for each level mainly include length, maximum diameter, angle between parent branch or trunk, and distribution and quantity information for next level of branches or leaves. Trunk and branches are always not straight. We use parameterized curves to depict the characteristics of bend. Parameters of leaves include size and distance to the last level of branches. It should be emphasized that many parameters described above are not single values but parameterized curves by which we can depict morphogenetic gradients according to positional information.

In our model, each of structural parameters has a random factor that would be given as a parameter. Thus, using different random seed makes it possible to generate numerous trees each different yet fundamentally similar to each other.

The procedure of creating a tree structure is straightforward. Starting with the main trunk, the algorithm constructs the tree by generating the branches in a hierarchical fashion where all branches of the same level use the same generation parameters. The recursive proliferation of branch is limited by levels of the maximum level of branches. Finally, leaves or needles are attached on the last level of branches.

The above model has two advantages. First, it is easy to adjust the tree shape without losing the main characteristics of some species by modifying a few parameters. We can generate trees belong to a same species with different sizes by modifying a predefined parameter template for that species. The details will be shown in Section 4. Second, we invoke our modeling engine to generate the tree at run-time. The storage requirement is modest for each tree because only its parameters and the random seeds are stored.

3.2 Rendering

The representation of a 3-D tree model can be separated into two different parts: the stems, including main trunk and branches, and the leaves. In this paper, each of these parts has been treated in a different manner.

The surface of a stem could be considered as a generalized cylinder with a circular cross section of varying radius [1]. To generate triangle strips of the stem surface for final rendering, a finite number of cross sections are evaluated along the stem axis and connected together. Each cross section consists of a finite number of points. We also use photograph of bark as texture.

Fig. 1 Represent a cluster of leaves by one textured quadrilateral

It is meaningless and impractical to model and render individual leaves in an interactive landscape visualization system. For greatly reducing the geometry detail, we use a textured quadrilateral to represent a cluster of leaves (Figure 1). This representation enable us greatly reduce polygons. In addition, screen-aligned billboard technique is used to orient the polygon.

Fake lighting of leaves is achieved by letting leaves near the center of the tree appear darker than those on the edges. Leaves are considered near the center of the tree if they grow near the base of their parent branch.

3.3 Levels of Detail

A multi-resolution model represents an object using different levels of detail (LODs). The finest LOD represents the full resolution model. Coarser LODs represent lower resolution versions of the model suitable for faster rendering. In this paper, the rendering algorithm selects a tree's LOD as a function of the distance from the tree to the observer.

For branches, the automatically computed LODs start with branches generated by the modeling engine and produce a user controlled number of discrete LODs where each new LOD is a reduced version of its predecessor. Reduction occurs by not only eliminating the least important branches but also reducing the cross-sectional resolution of those that remain. The criterion for evaluating the importance of branches is: the branches in a lower recursive level are more important than those in higher recursive levels; in the same recursive level, the longer branch is more important than the shorter one.

The leaves number is reduced when a lower LOD leaf set is created from an existing set. Meanwhile, in coarser LOD, leaves sizes are increased to represent original tree without loss leafiness. Figure 2 shows results of the LOD algorithms.

Fig. 2 Results of the LOD algorithm. a) 1,758 triangles for leaves. b) 372 triangles for leaves. c) 44 triangles for leaves.

For the use of levels of details, one may not neglect the issue of how to stage the two successive representations. Instead of simply switching two models, we achieve smooth transition by blending two neighboring LODs, which eliminates the "popping" that occurs in typical LOD transitions.

4 Visualization of Forests

4.1 Required Data

Four main sources of data are used to create the scene for a given forest landscape: tree parameter templates for each involved species, forest inventory data, a forest stand map, and a digital elevation model (DEM).

A tree parameter template mainly includes a set of parameters that could be used to create a tree model representing some species through the parameterized procedural model algorithm introduced in Section 3. At run-time, by properly adjusting a predefined parameter template for some species, we could generate tree models with desired sizes while having the recognizable characteristics of that species. We have developed an interactive modeling tool for effectively designing tree parameter templates. As parameters are changed, tree could be previewed immediately. At last, not only a set of parameters but also the size information of corresponding tree model is saved in a file called tree parameter template.

In present-day, forest inventory database are generally managed not in units of individual trees, but forest stands. A forest stand is a group of trees that have similar structures and are in the same growth stage. In this work, a forest stand table contains information on the area of a stand, together with the dominant species, average height, average crown width, average DBH and density of the dominant trees in the stand. A forest stand map expresses the boundaries of each stand.

Simply stated, DEM is an array of elevation points. DEMs can be created from contour lines and spot heights. DEMs also can be created by applying photogrammetric processes to overlapping stereo image pairs obtained from satellites with adjustable viewing geometry.

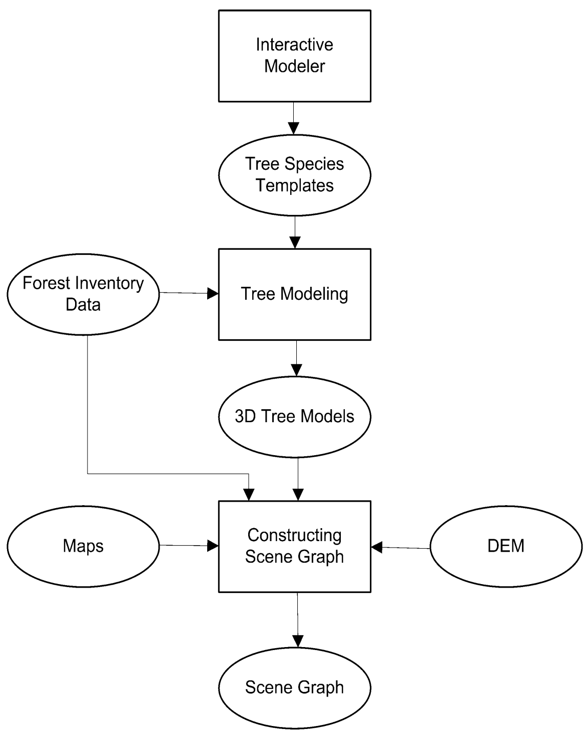

Fig. 3 Pipeline of constructing forest scenes.

4.2 Scene Graph

The modeling pipeline for forest visualization is given in Figure 3. We construct forest scene by transforming forest inventory data of every stand into a scene graph one by one. Every stand is added to the scene graph as a group node that has a common parent group node representing forest.

Finally, the model database is organized as a scene graph. The scene graph is a bounding volume hierarchy that supports hierarchical culling techniques. At run-time, the scene graph is traversed recursively and user-defined functions are called to process each node.

4.3 Distribution of Trees

To perform stand visualization, spatial information including tree location is needed. Since this information is not included in the inventory information, tree locations for stand visualization are generated using a random algorithm. Individual trees are populated in the stand according to the intensity value. After the uniform grids have been computed, a random factor is applied within certain limits to the grid node coordinates ($x \pm deltaX$, $y \pm deltaY$) during the allocation of the trees. The height of the base of a tree object equals terrain elevation at the location of that tree.

4.4 On the Fly Generation of Tree

For each forest stand, we firstly load an appropriate pre-designed tree parameter template according to dominant species. Then, we adjust parameters in template to generate geometry model for this stand according to size parameters in forest stand inventory data including DBH, tree height and crown width. In following equations, a variable with subscript stand represents corresponding value in forest stand inventory data; a variable with subscript template represents corresponding parameter in template.

For each forest stand, we firstly load an appropriate pre-designed tree parameter template according to dominant species. Then, we adjust parameters in template to generate geometry model for this stand according to size parameters in forest stand inventory data including DBH, tree height and crown width. In following equations, a variable with subscript stand represents corresponding value in forest stand inventory data; a variable with subscript template represents corresponding parameter in template.

To generate a model with a given DBH, the following $scale_{radius}$ is used to scale the radius parameters of trunk and branches:

$$scale_{radius} = DBH_{stand} / DBH_{template}$$

Now we adjust the length of trunk and branches for generating a tree model with a given height and a given crown width. Different methods are used according to branching patterns.

For monopodial branching tree (where apical meristems are the primary creators of new stems and leaves), its height is linear with the length of trunk and independent with length of branches. Therefore, we scale the length of branches in parameter template with:

$$scale_{branch\_length}=crown\_width_{stand} / crown\_width_{template}$$

and scale the length of trunk with:

$$scale_{trunk\_length} = heigh_{stand} / heigh_{template} \quad (1)$$

For sympodial branching tree (where each apical meristem dies, and an axillary meristem creates the next stem and leaves), tree height is dependent with both trunk length and branches length. Firstly, we still use Eq. (1) to scale branches length for getting desired crown width. Then, we compute heightnew, the height of the tree model generated with the adjusted parameters in template. At last, we replace the length of trunk in template by:

$$trunk\_length = trunk\_length_{template} + (height_{new} - height_{stand})$$

With those new parameters, we could invoke the modeling engine to create a geometry model faithful to forest stand inventory data. At this point, we also create texture maps that represent far trees. It is accomplished by render to texture function in OpenGL extension.

In order to gain a more realistic and nature forest scene, for every stand, we generate several tree models with the same tree parameters but different random seeds. Other trees in this stand are rendered as instances of those models

4.5 Rendering Optimization

Though forests of only a couple of trees may be visualized by rendering individual trees in a loop, once the number of trees in a scene increases beyond fifty, modifications are required to maintain real-time frame rates. As switching graphics mode and graphical attributes involves a performance penalty, the data should be presorted by type and mode if possible to avoid changes in the state of the image generator. We group leaves of all trees by their texture and draws those groups one by one. The same method is applied to trunks and branches.

Texture compression is also used in this system, which enables a reduction of texture data by one-fourth to one-sixth its original size while maintaining the image quality of the original artwork. Therefore, texture image can be stored with significantly less memory hence improve overall system performance.

By the usage of above optimizations it is possible for the user to move in real-time through such a virtual forest and thereby viewing it from an arbitrary position.

5 Application

We have applied the method described above in VFIS (Visual Forest Information System), a GIS-based 3-D forest resource management system aimed at constructing a virtual forest with immersive capability for navigation, query and analysis. VFIS was developed using ESRI ArcObject that comprises an object-oriented geographic data model and an integrated library of software components. Forest inventory information, forest stand maps and DEMs were stored and managed in a relational database management system by ESRI ArcSDE, a spatial database engine.

Our implementation of the visualization subsystem is written in C++, using OpenGL and OpenSceneGraph [14] libraries. We ran our tests on a PC with a 2.4G Pentium 4 processor, 512M RAM and an NVIDIA GeForece4 display adaptor.

VFIS allows user to select an interested area on 2-D map and jump into the corresponding 3-D scene. Figure 4 shows a screenshot of moving through in virtual forest. The frame rate is typically 25-35 Hz. Figure 5 is a screen shot of showing an example of querying forest stand information in a 3-D forest scene.

6 Conclusion

We have presented an approach for data-driven interactive visualization of forest landscape with applications in forest resource management. In this system, geometric models of individual trees are generated dynamically faithful to forest inventory data. With the hybrid representation for 3-D tree models and specific LOD algorithm, our system can provide real-time frame rates to ensure user-steered interactive displays.

Our work is but an early step in the development of techniques for automatically creating and real-time rendering data-driven forest scenes, and the presented concepts require further research. We are developing a paging algorithm that will allow the system to visualize arbitrarily large forest database.

It is our hope that the approach presented in this paper may also be applicable to other GIS-based forest simulation systems where real-time rendering 3-D tree models is necessary.

Acknowledgments

We would like to acknowledge contributions from Yuqin Su, Chengfang Song, Quanfeng Zhu, Jianwei Li, and Song Shi. Special thanks go to Mi Liu for providing the convenience of accessing related literature. This research was sponsored by Preliminary Project of 973 Program (No. 2002CCC01900) in China.

References

1. J. Bloomenthal. Modeling the Mighty Maple. *Proceedings of SIGGRAPH '85*, pp. 305-311,1985.
2. F. Boudon, P. Prusinkiewicz, P. Federl., C. Godin, a nd R. Karwowski. Interactive design of bonsai tree mo dels. *Proceedings of Eurographics 2003*, to appear, 20 03.
3. O. Deussen, P. Hanrahan, M. Pharr, B. Lintermann, R. Měch, and P. Prusinkiewica, Realistic Modeling and Rendering of Plant Ecosystems. In *Proceedings of SIGGRAPH 98*, ACM Press / ACM SIGGRAPH, E. Fiume, Ed., Computer Graphics Proceedings, Annual Conference Series, ACM, 15–22, 1998.
4. O. Deussen, C. Colditz, M. Stamminger, and G. Drettakis. Interactive Visualization of Complex Plant Ecosystems. *Proceedings of the conference on Visualization '02*, pp. 219-226, 2002.
5. A. Jakulin. Interactive Vegetation Rendering with Slicing and Blending. Eurographics'2000 Short presentations, 2000.
6. E. Karjalainen and L. Tyrvainen. Visualization in forest landscape preference research: a Finnish perspective. *Landscape Urban Plann*. 59, 13-28, 2002
7. E. Lim and T. Honjo. Three-dimensional visualization forest of landscapes by VRML. *Landscape and Urban Planning*, 63, 175-186, 2003.
8. B. Lintermann and O. Deussen. Interactive modeling of plants. *IEEE Computer Graphics and Applications*, 19(1): 56-65, 1999.
9. J. Lluch, E. Camahort, and R. Vivo. Procedural Multiresolution for Plant and Tree Rendering. In *Proceedings of the 2nd international conference on Computer graphics, virtual Reality, visualisation and interaction in Africa*, pages 31-38, 2003.
10. N. Max and K. Ohsaki. Rendering Trees from Precomputed Z-Buffer Views. *Eurographics Workshop on Rendering 1996*, pp. 165-174, 1996.
11. A. Meyer, F. Neyret, and P. Poulin. Interactive Rendering of Trees with Shading and Shadows. *Eurographics Workshop on Rendering*, 2001.
12. P. Oppenheimer. Real Time Design and Animation of Fractal Plants and Trees. *Proceedings of SIGGGRAPH '86* (Dallas, Texas, August 18-22, 1986). In Computer Graphics Proceedings, Annual Conference Series, 1986, ACM SIGGRAPH, pp. 55-64, 1986.
13. B. Orland. Smartforest: 3-D interactive forest visualization and analysis. In *Proceedings of Decision Support-2001*, Resource Technology 94, Toronto. American Society for Photogrammetry and Remote Sensing, Bethesda, Maryland, 181-190, 1994.
14. OpenSceneGraph. http://www.openscenegraph.org, 2003
15. L. Perrin, N. Beauvais, and M. Puppo. Procedural landscape modeling with geographic information: the IMAGIS approach. *Landscape and Urban Planning*, v 54, n 1-4, p 33-47, 2001.
16. P. Prusinkiewicz, and A. Lindenmayer. *The Algorithmic Beauty of Plants*. New York, Ed. Springer-Verlag, 1990.
17. P. Prusinkiewicz, L. Mundermann, R. Karwowski, and B. Lane. The use of positional information in the modeling of plants. In *Proceedings of SIGGRAPH 2001*, ACM Press / ACM SIGGRAPH, E. Fiume, Ed., Computer Graphics Proceedings, Annual Conference Series, ACM, 15–22, 2001.
18. W. T. Reeves and R. Blau. Approximate and probabilistic algorithms for shading and rendering structured particle systems. *Proceedings of SIGGRAPH'85* (San Francisco, California, July 22-26, 1985), in Computer Graphics, 19, 3 (July 1985), pages 313-322, ACM SIGGRAPH, New York, 1985.
19. I. Remolar, M. Chover, O. Belmonte, J. Ribelles, and C. Rebollo. Geometric Simplification of Foliage. *Eurographics'02 Short Presentations*, pp. 397-404, 2002.
20. C. Ulbricht. Vantage Point: *An Interactive Landscape Simulation Tool for Engineering Design*. MSc Thesis, University of Washington, WA, 1994.
21. J. Weber and J. Penn. Creation and Rendering of realistic tree. In *SIGGRAPH 95 Conference Proceedings*, pages 119-128, 1995

Fig. 4 Moving through the forest, rendered with typically 25-35Hz.

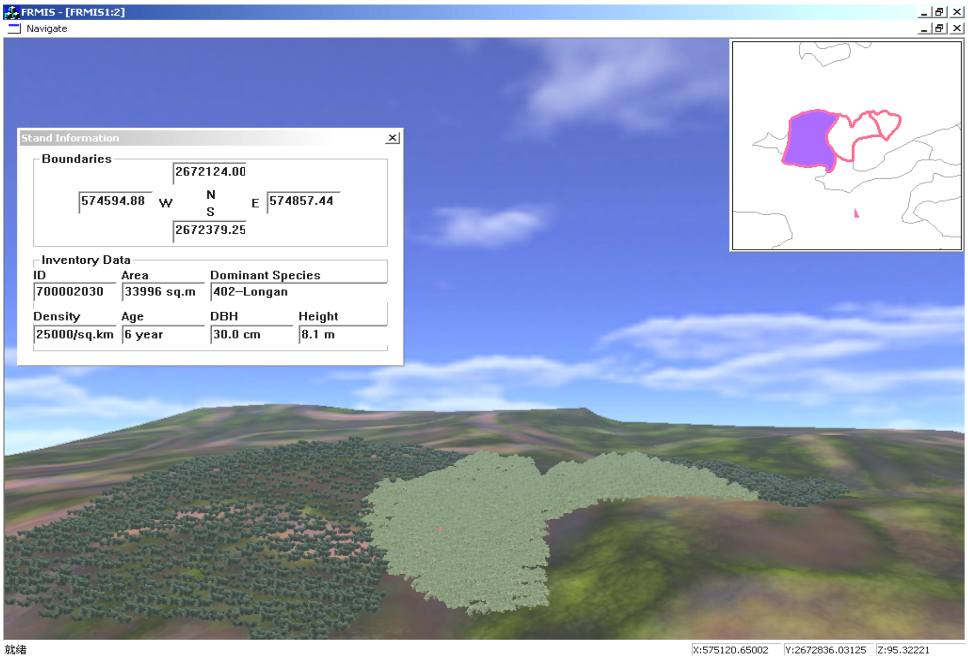

Fig. 5 Screenshot of VFIS. (The eagle eye window highlights boundaries of three interested stands in the map. The left dialogue shows forest inventory data of the stand currently queried, cyan area in eagle eye window.)