Tree and Plant Volume Imaging
—An Introductive Study towards Voxelized Functional Landscapes

M. Jaeger[1, 2], J. Teng[2]

[1]*jaeger@liama.ia.ac.cn, CIRAD AMAP, TA 40/PS2, 34398 Montpellier Cedex 5. France*
[2]*jteng@nlpr.ia.ac.cn, LIAMA, CASIA, B.P. 2728, 100080, Beijing, China*

Abstract

Virtual vegetation, and more precisely its geometrical representation and visualization ask both high memory space and high computing time costs.

The main contribution of this paper is to propose an open, fast, efficient 3D plant exhaustive spatial representation from tree/plant skeleton discretization. This voxel based representation can carry topological, geometrical and functional information.

Two single application examples are presented: biomass volume estimation and visualization. Diameter retrieval is exposed leading to a classical exhaustive voxel plant description in a constant time process. Including discrete terrain voxel representation, volume rendering shows the interest of the method on small landscapes for fast visualizations purposes.

Perspectives point out that this approach could be a key feature for functional landscape simulations.

Keywords: computer graphics, natural phenomena, plant, tree, landscape, 3D discretization, volume rendering, discrete ray tracing, transfer functions, voxel

1 Introduction

The context of the presented work is structural and physiological landscape simulation. In the challenging field of vegetation modeling, geometrical representation is a high cost process [1]. In agronomy and forestry, exhaustive plant 3D representation is required for any anisotropic study involving physiology with spatial interactions and competition [2, 3]. Such is the case of homogenous crop plantation when spatial environment parameters are varying or when fine variation analysis is required [4]. Interactive complex natural scenes are also a growing challenging demand, where plant behavior must remain realistic and have fast response to external user specifications [1,5].

This Study is supported by LIAMA Projects 98-01"GreenLab" and 01-08 "VPaleo"

Unfortunately virtual plant geometrical representation and visualization have high complexity compared to its topological or formal representation, and no physiology study can be led without minimal support of geometry. Of course geometrical complexity decimation or degraded representation is an active research area for fast multilevel-based visualization [1, 6, 7, 8], but for many applications, such works are difficult to upgrade when plant geometry has to carry functional data or environmental data.

Virtual plant representation must thus integrate its environment, allow spatial interactions, be able to express an exhaustive 3D geometry, be highly interactive, allow multi-level views (from plant organ to canopy), be a support for functional simulation and fast visualization. The ground idea of the presented approach is to use the space itself as a support for vegetation scene simulation and visualization.

1.1 Related works and justification

This idea is in fact well known in 2D; many works were done using maps coming from Geographic Information Systems (GIS), or competition diagrams. A survey of such techniques can be found in Ervin and Hasbrouck's book [9]. In the field of interactive vision, use of texel in natural scenes was tested. Initially described by F. Neyret [10], single plant is converted to texels and then pasted on one terrain surface. Complex billboards consisting in plant slicing views can also be considered as a similar technique [11]. But homogeneous discrete space (voxels), is seldom used. One can cite Ned Greene [12] on a voxel based plant growth model, and some other authors like F. Blaise [13] using a rough discrete space for collision detection purpose.

Our approach consists in discretizing the full landscape in order to use the inherent voxel topology as interface for plant/soil/light interaction and allow fast constant time visualization.

On one hand, this approach makes sense since technology development makes huge RAM on low cost PC available. One Gb RAM allows the representation of a scene of 10000 square meters with 0.1m resolution. On the other hand, new developments in Greenlab project[14, 15] on single detailed plant visualization are based on implicit surface approaches (see J. Bloomenthal pioneer's work [16]). And an efficient, classical way to render those surfaces is based on a regular space subdivision isosurface drawing, known as "marching cube" [17, 18]. GreenLab plant models do also involve both structural and functional approaches, where functional parts (wood volume, leaf area) is related to visible elements. Encoding voxels elements with such data has therefore the double interest of storing functional data and enhancing the geometrical information for rendering. Moreover, volume image rendering gained in the past decade interesting new approaches and techniques[19, 20]. And finally we have developed, generic and applied image volume tools [21] focused on large volume data set and fast rendering for forestry purposes.

1.2 The data process pipeline

The landscape simulation flow is organized on the support of volume image space as follows:

```
Define, allocate and initialize voxel space to NULL
Loop:
    read and discretize terrain file            (step 1)
    read and discretize virtual plants          (step 2)
```

```
        retrieve functional parameters from voxel space          (step 3)
        upon request :
                do diameter retrieval   (step 4.1)
                do visualization        (step 4.2)
                stop simulation
        send functional parameters to growth engine and terrain engine        (step 5)
    read external data (temperature, rain...),   update terrain file and virtual plants (step 6)
```

In this introductive study, work is strictly focused on the plant to voxel conversion, data retrieval and volume rendering: steps 2, 3, 4.1, 4.2 in the loop described above. More precisely, as defined in Fig.1, vegetation is defined by a set of individual plants those are converted to voxels from a list of exhaustive tree components or from a rough single shape.

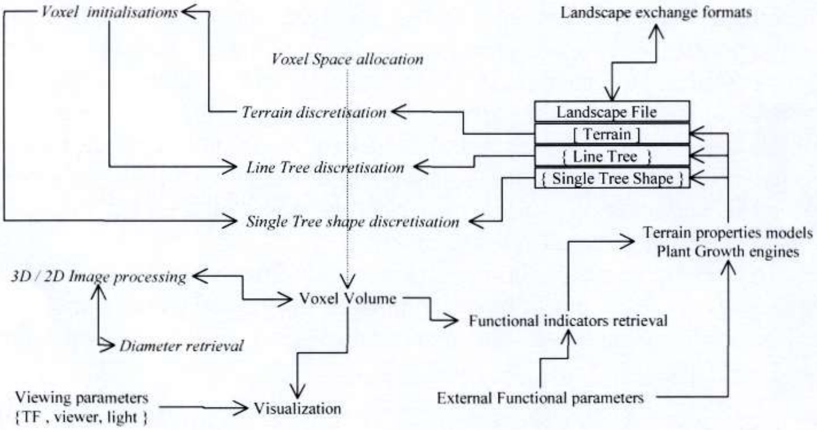

Fig.1 Landscape simulation overall chartflow

Following section is dedicated to tree data volume discretization, while next one illustrates plant data retrieval, the next section is dedicated to fast volume landscape visualization and the last section concerns performances.

2 Tree Data and Volume Discretization

The idea here is to convert the plant skeleton instead of an exhaustive geometrical boundery representation (BREP) such as polygon list or parametric patches. Plants are supposed to be described by a list of components, organized or not, with basic geometrical aspects (position, orientation, size), and basic characterization (wood/leaf). Two levels of description are tested: exhaustive simulated tree outputs (Bionatics AMAP™ or GreenLab generated plants [14,22,23]), compatible with L-system techniques [1,2,5] and crude simplified tree description like cylinder for trunk and basic shape for crown, compatible with usual forestry single tree characterization [24,25,28]. Note that the topology of plants (if exists) will be lost by the process of discretization. This information can be kept by the plant simulation process, which is clearly outside the scope of this study.

2.1 Tree data

Exhaustive tree description hold on AMAP™ simulated trees is based on plant "LineTree" as defined earlier by the author [22] based on P. Oppenheimer data structure [26]. LineTree is an unorganized display list defined from a list of components. Each component contains "topologic" data (bearing branch, age,...) and a 4x3 geometric matrix. Tree geometry and rendering need therefore instantiation of various geometrical primitives (BREP shapes) transformed by the affine operator corresponding to the encoded 4x3 matrix.

Here, each component will be described by a single line that characterizes its main direction (with its length). Applied to whole plant, this can be seen as an exhaustive tree skeleton.

2.2 Tree data voxel conversion

Two volume data sets are generated. The first one is defined from LineTree component labels, i.e. voxel value refers to a component index such as branch, leaf, flower. The second volume refers to functional parameters of the plant, i.e. voxel value is a grey value that express volume of woody parts or surface of leaves. In practice, we do encode diameter value for branch and surface area for leaf. A branch (or a portion of it) is then described by its origin, its end and both diameters at ends; and a leaf by its origin, its end (of main direction) and its surface. So, each component is described by a couple $(x1, y1, z1, val1)$, $(x2, y2, z2, val2)$ which defines a 3D grey level line.

Technically, discretization is perform by a classical 3D Bresenham line drawing [27] in an one-pass process. Any line segment (corresponding to a component axis) is discretized with grey level interpolation (standing for start and end diameters) which values are defined by:

$$GreyLevel\_value = Cd * diameter \tag{1}$$

$$\text{where } Cd = 2^{\text{number of encoding bits}} / Max\ (diameters).$$

We set $Cd = 16$, so diameter range is 1/32 to 128 cm for 4096 grey values. In case of conflict (several components in the same voxel), voxel with the highest value wins. Algorithm is so similar to 3D Bresenham drawing with Zbuffer interpolation. Discretization process CPU time is not significative (see below for performances). The result of the discretization process is illustrated in Fig.2 with single plotting (no illumination). Color is defined from component nature in the labeled volume while grey level illustrates diameter of axes or leave surface. Therefore trunk appears in light color while thin branch axes appear darker.

Fig.2 Label and grey level line tree discretisation

This discretization policy has several side effects. Several organs may share the same voxel leading to bias in functional data retrieval and rendering. Inner voxel participating proportion could be encoded to solve this. Concerning woody parts (branches), note that the voxel diameter encoding can be considered as such a technique. If *scale* is the metric resolution of a voxel (usual value range is from 3cm to 15cm), any voxel with corresponding diameter higher than *scale* is "full" and should then not be shared with other organs; otherwise, sharing should be done. Technically, use of content lists inside each voxel will bring a high level of computation time and memory space increase. Nevertheless, an efficient low cost approach to test would consist in the use of only 2 contributions respectively concerning leaves and axis (active and passive organs in terms of functioning). Inner voxel value could also encode a resulting function of the contibution distribution (i.e. not just the maximum value). Another side effect of the process comes from the algorithm that generates aliased 3D lines. That effect can be solved using a 3D grey level antialiased line drawing procedure. These improvements are not done yet but should be hold on in the future.

2.3 Other voxel conversion of usual tree description

Process described above can be used for any tree or plant description that can be expressed from line extrusions. Classical tree description in forestry is thus limited to a single path carrying trunk diameter and then crown diameter; a single cone shape tree will be defined by two aligned consecutive lines. Other usual single crown shapes can also be expressed easily and used to generate landscapes as shown in Fig.10, making the approach compatible with many forest plantation or silviculture scenario simulators: Capsis [24], SVS [28] ...

Note also that this approach can be used for any branching shape defined by a curved skeleton or set of points with extrusion functions. This includes in particular extruded generalized cylinders, and implicit surfaces defined by 0D or 1D primitives (including soft objects or metaballs). Once the linetree discretization is performed, both discrete grey level and label spaces can be used for a wide range of applications.

3 Functional Parameters Retrieval from Voxel Space

In this section, we will only focus on plant production retrieval, i.e wood volume and leaf area, both essential input and output simulated parameters of advanced growth models sensitive to environmental conditions[2,13]. Total leaf area retrieval from voxel space is an easy process that will not be detailed here (sum grey levels corresponding to leaves).

3.1 Computing the wood production from voxel space

Single branch component ($X1, Y1, Z1, D1$), ($X2, Y2, Z2, D2$) as described in previous section can be considered as a cone frustum which volume is:

$$Vc = 1/3\ \Pi\,(d1^2 + d1\ d2 + d2^2)\ .\ ((x2-x1)^2 + (y2-y1)^2 + (z2-z1)^2)^{0.5} \qquad (2)$$

Full wood volume of a plant can be estimated by the sum of component contributions (we do suppose that self intersections are not meaningful).

Branch volume retrieval from grey level volume data can thus be evaluated as follows:

$$V = Cv\ \Sigma(Vox[i][j][k])^2 \qquad (3)$$

where

$$Cv = \Pi\ .\ \text{scale}\ .\ (Cd)^{-2}$$

and scale stands for voxel metric resolution.

Label volume data can be used as a mask to consider only woody parts and/or focus on a given specific plant. Interest of this approach is that this volume estimation seems to be quite stable when space resolution decreases despite a over estimation coming from low diameter values. Nevertheless, this study has to be enforced; by using floating point voxel encoding, relation to average length, relation to number of plant components for better value retrieval procedure and discretization effects compared to cone frustum explicit voxel conversion. Volume estimation can so be processed on a huge number of plants at the same resolution level. Another way to evaluate the wood production is to retrieve the diameter in the volume grey level data set by 3D dilation process.

3.2 Geometrical diameter retrieval

An intuitive and basic idea is then to propagate diameter (size) information to neighbour voxels. This morphological grey level dilation operation is high time consuming; it leads to compute on the full grey level volume a distance field function from grey values. As underlined by Jones and Satherley[29], distance field computation approximation can be done in a single recursive pass. In our approach current diameter (grey level value) is propagated to its d-connected and i-connected neighbours, with the appropriate grey level decrease. Basis of the implemented algorithm (in 2D) is given hereby:

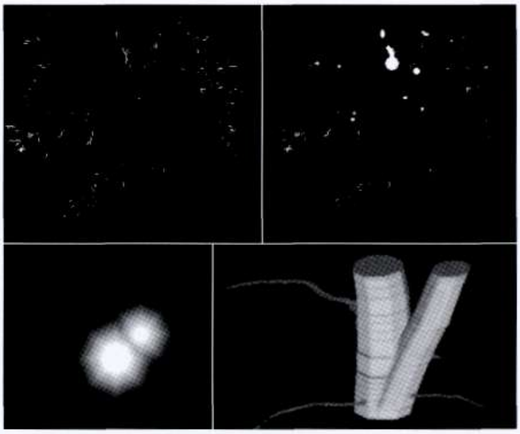

Fig.3 Diameter retrieval. Topleft:original image. Top right: diameter restoration. Bottom: close up near branching. Slice grey level and 3D reconstruction

```
//     Voxel[i][j] stands for grey level voxel value at adress (i,j,[k])
//     dg is the d-connex grey level decrease,         ig is the i-connex grey level decrease
dg = decrease; /* = scale . Cd    */
ig = sqrt(2) * decrease;
// loop on the image          (inner loop on i ), $k^{th}$ slice of the volume
//       j = 0 to yres–1                i = 0    to xres-1
                Voxel[i+1][j]            =   Max (Voxel[i+1][j], Voxel[i][j] - dg) ;
                Voxel[i+1][j+1]    =   Max (Voxel[i+1][j+1], Voxel[i][j] - ig);
// Similar work is then done by reversing the loop on the $k^{th}$ sliceof volume
       // by    reverse way onY_axis and on    X
       // by exchanging X_axis and    Y-axis, reverse way on Y axis
       // by exchanging X_axis and    Y-axis, reverse way on X axis
```

The effect of this process is shown in Fig.3. Extension of the algorithm to 3D increases the process time without significant gain of quality. Despite the low complexity and the low amount of computation perform on each voxel, this process is time consuming (see Section 5 for figures). And side effects (d-connect and i-connect direction prior and truncated integer operations) make process not suitable for volume estimation. At this time, this approach is therefore only suitable for visualization purposes. But it can be used as a quick and dirty implicit surface visualization, if defined from 0D and 1D skeletons.

4 Plant, Tree and Landscape Volume Visualization and Rendering

Volumes data sets, processed or not, can be efficiently used for visualization by classical volume rendering techniques. All rendering examples shown here were perform by C2000 in house volume image platform and all functions are written without use of any graphical library or use of any hardware capabilities. Visualization is supposed to be a requested step consecutive to terrain and vegetation volume discretization as defined in Section 1.2.

4.1 Terrain discretization and filling, plant discretization

Terrain model was defined in this study from a grey level image generated by Terragen© freeware[30]. First, grey levels corresponding to altitudes were written in the volume data set. Then volume below terrain surface was filled and volume label data set was updated. This process is quite fast (see next section). A set of plants (LineTrees or single shapes) are then discretized. Plant position, here randomly chosen, is then updated in altitude according to existing altitude in the volume data sets. Finally, for each tree component, geometrical transformation (scale, rotation, translation) is applied before discretization process (see 2.2).

4.2 Image based rendering

An easy way of visualization is to plot each non-empty voxel while reading the volume data set. This quick and dirty image based "like" rendering can be improved by introduction of Zbuffer, various colors according to volume label, simple illumination (diffuse reflection) and depth cueing post process. These techniques are illustrated below in Fig.4. Gradient for illumination is computed by a classical local differentiation in grey level volume data set[19]. This scene of 320 plants is build from 2 million components. Images are generated in a few seconds while same scene consisting in 135 million polygons take near 3 minutes using an NVIDIA compatible 128 Mb Card. Visual impact is quite convincing despite the fact that diameter retrieval was not performed (see thin main axis stems on right view).

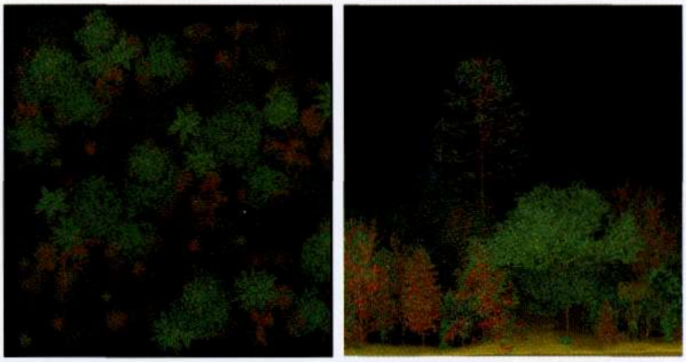

Fig.4 Forest label volume visualisation. Side view includes terrain elevation. Colors are affected according to line tree components. Depth cue process is added

This approach can also be applied after diameter retrieval process (Fig.3 bottom and Fig.5).

Fig.5 Grey level threshold effect on common poplar rendering

On these examples, a volume isosurface rendering technique, dividing cube[17] was used. By changing potential value of the surface to draw, i.e. threshold grey level values (diameter), various levels of details can be defined. Such approaches are voxel content driven, they have linear complexity to number of filled voxels (multiplied by the subdivision factor for Dividing cube, given by the voxel projection size on final image).

4.3 Towards realistic visualization

Volume ray tracing is a classical way to render discrete volume[19]: rays are sent from eye position thru final image pixel centers to the volume data set. Final pixel color value is set by the highest contribution (MIP), the average value (AVE), a fixed value (similar to isosurface), or the sum of all traversed voxels (transfer function).

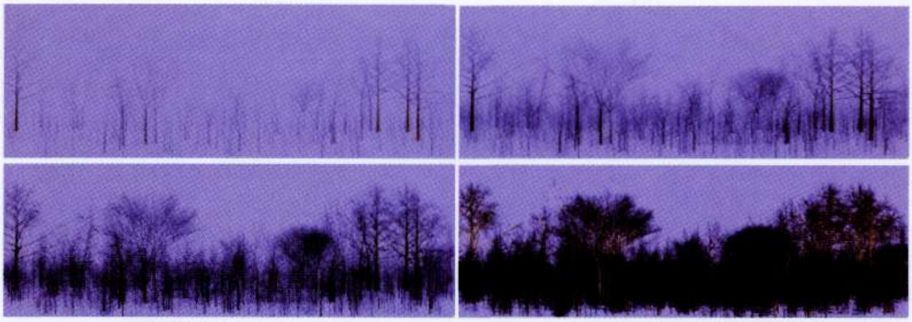

Fig.6 Ray traced wood forest rendering. Transparency is used in transfer function for LOD effect

In this last case, various transparencies and colors are given according to different grey levels. We did implement MIP, AVE and transfer function algorithms with a 3D line Bresenham voxel traversal based algorithm. Algorithm is written front to back, ray path may stop when voxel tranversal reaches final pixel contribution. Fig.6, Fig.7 and Fig.8. illustrate ray traced transfer function volume rendering examples. The Lambertian diffuse illumination model previously cited is used. Normal is computed for any voxel that contributes to final image by differentiation of grey level on neighbour voxels.

In these approaches, complexity is linear in number of sent rays multiplied by average ray path length. In practice, if rough illumination is satisfactory, rendering time is shorter for complex scenes (ray traversal stops when voxel opacity is high).

Fig.7 Volume rendering examples using discrete volume ray casting with transfer function. Left to right: poplar tree, oak tree, wood forest. All line tree plants were generated by AMAP-Genesis^{TM}

Note finally that rendering computation time could drop down by the use of specific hardware or by a texture based hardware volume rendering [31, 32]. Such techniques should be tested in the future.

5 Performances

Implemented algorithms were tested on C2000[21] in house image volume tool on 700 MHz PC with 256 RAM memory (Linux). As shown in Table 1, for each scene, number of trees, number of components, number of axis (componants dedicated to diameter retrieval), voxel space resolution, allocated RAM, CPU time for: initialisation and terrain discretization, tree discretization, wood volume estimation were reported. Total CPU time stands for a full cycle as described in Section 1.2 including rendering and full volume data sets disk storage procedures.

Table 1 Performances obtained for a full simulation loop(CPU time in 10^{-3} sec)

Name	Prunus	Poplar	Poplar	Oak	Oak	Forest1	Subwood	Forest2(*)
Figures	2 and 5	7 left	7 left	7 mid.	7 mid	4	6	7 right
Nb of Tree	1	1	1	1	1	317	1661	600
Components	7093	8941	8941	52859	52859	2141636	5108927	5143224
Nb of Axis	5281	7262	7262	52859	52859	1461348	2841346	3547620
Resolution (X,Y,Z)	400 357 438	170 150 408	220 220 544	400 400 200	385 370 473	600 600 190	780 780 105	1000 1000 330
Needed RAM in Mo	178	29	75	91	192	195	182	944
CPU Init, Terrain	-	-	-	40	-	360	300	937
CPU Discretization	30	30	30	130	220	6140	12740	61625
CPU Volume	480	80	210	11610	530	960	950	-
CPU Diameter	22700	3640	9260	11610	24370	26500	25030	125141
CPU Total	26950	4250	11630	13210	30170	39710	41500	280156

(*)Figures of this column were obtained on another system CPU 2GHz, 1Gb RAM, Windows2000

Rendering is discrete volume ray tracing with transfer functions except for Fig.4 (image point rendering) on a standard 512 by 512 image.

Process complexity can be analyzed as follows:

Discretization : $C = O$ *(nb of componants * Volume Space)*
Relief : $C = O$ *(Volume Space) (for constant altitude map size)*
Volume estimation, Diameter retrieval $C = O$ *(Volume Space)*

Extensive trials were done varying voxel resolution as shown in Fig.8.

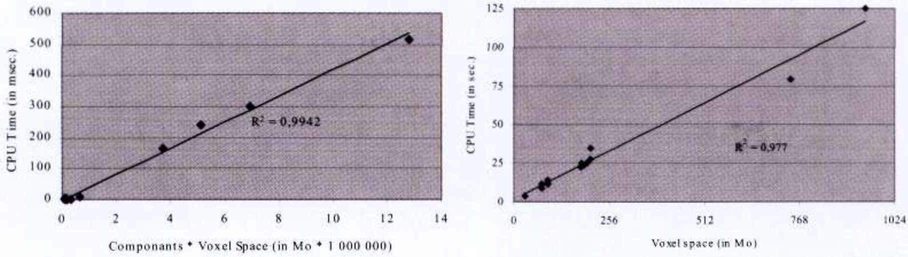

Fig.8 CPU time for plant volume discretization and diameter retrieval process. Left: LineTree discretization process CPU time(in 10^{-3} sec) versus number of tree components and voxel space. Tree components range is 5.10^3 to 5.10^6, voxel space range is 30 to 210 Mb. Right: Diameter retrieval process CPU time (in sec) versus voxel space. Note the high time cost of this process

Diameter retrieval process is the most expensive process, followed by the plant discretization when the scene is complex (when number of components gets near $C_3/C_1 \sim 5.10^6$) as shown in the following equation:

$$Total\ Time = C_1 \,.\, Nb\_Components \,.\, Voxel\ space + (C_2 + C_3 + C_4) \,.\, Voxel\ space\ (R^2 = 0{,}9993)$$

where $C_1 = 28.10^{-6}$ *stands for discretization ,* $C_2 = 5$ *stands for volume estimation,*
$C_3 = 130$ *stands for diameter retrieval,* $C_4 = 15$ *stands for rendering and storage*

Volume retrieval tests from voxel space where performed on two synthetic objects (single cylinder and cone) and on a simulated oak tree (Fig.7 middle) by changing resolution from 512^3 to 32^3.

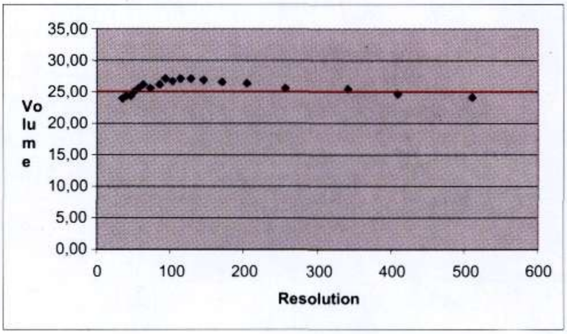

Fig.9 Volume estimation from voxel space variation versus resolution. Red line is volume computed from line tree components

Results on synthetic single components are very stable, results on simulated tree is shown below in Fig.9 and is quite satisfactory but should be analysed closely.

6 Conclusion and Perspectives

In this preliminary study, we show that discrete 3D volume image technique can be an interesting spatial support for landscape simulation purposes.

Skeleton of virtual simulated plants can be converted to a labelled and a grey level voxel space. Single axis diameter encoding in voxels makes this conversion powerful : volume and diameter retrieval can then be done in a constant time with sufficient accuracy. Exhaustive as well as simplified plant representations can be used and integrated in scenes with solid terrain. However, further studies should be hold on the line tree voxel discretization, volume estimation and diameter retrieval process on both accuracy and speed aspects. Use of fine distance field function estimation should be developed as well as development of efficient algorithms with the current morphological approach. Such investigations may be extended to innovative implicit surfaces rendering techniques.

We also show that image based or volume ray tracing rendering can be used to obtain fast and realistic images : complexity is function of voxel resolution and is weakly linked to plant data structure. The obtained results seem to be promising in terms of rendering speed, basic illumination, component classification and even level of detail. For close views, beside classical troubles of volume rendering discretization, fine rendering of axis and leaves has to be improved. Use of several transfer functions, indexed on the label volume data set, should lead to more realistic "poly-chromatic" pictures (Fig.10).

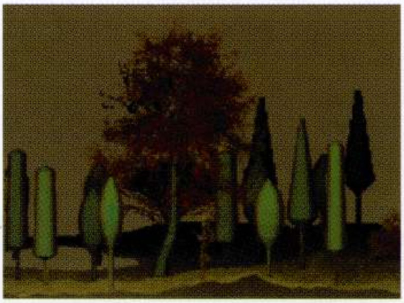

Fig.10 Volume rendering with terrain on exhaustive and single shape trees. Note side effect due to single transfer function ("green" trunk on oak)

Leaf geometrical area retrieval is another feature to be developed for functioning purposes (light computations for instance) and visualization. Inherent capabilities of LoD/LoV were poorly developed. They could be focused on three points: - plant LoD description (mixture between exhaustive and simplified representations of skeleton), - efficient use of production information in voxels, - use of volume resolution changes.

Significative gain should moreover be reachable when rendering from the use of texture hardware capabilities.

Finally full software interfaces and integrations with growth engines, soil moisture content models, climate files are to be boosted, tested and validated on study cases.

References

1 O. Deussen, P. Hanrahan, B. Linterman, R. Mech, M. Pharr, P. Prusinkiewicz. Realistic modeling and rendering of plant ecosystems. Proceedings of SIGGRAPH'98, pp. 275-286, July 1998.

2 W. Kurth, B. Sloboda. Tree and stand architecture and growth described by formal grammars. II. Sensitive trees and competition. J. Forest Science., 45, 53-63., 1999

3 F. Sillion and C. Soler. SOLEIL. Application of the radiosity method to the physiological simulation of plant growth. P. 21-22, ERCIM News No. 44, January 2001

4 P. Wernecke, G. Buck-Sorlin, W. Diepenbrock. Combining process with architectural models: the simulation tool VICA. SAMS, Systems Analysis Modelling Simulation special issue vol. 39, pp. 235-277, 2000.

5 P. Prusinkiewicz, M. James and R. Mech. Synthetic Topiary. Proceedings of SIGGRAPH'94, pp.351-358, July 1994

6 I. Remolar, M. Chover, O. Belmonte, J. Ribelles, C. Rebollo. Geometric simplification of foliage. Proceedings of Eurographics'02. ISBN/ISSN 1017-4565, Saarbrücken (Germany), pp. 397-404, 2002.

7 D. Marshall, D. S. Fussel, and A. T. Campbell. Multiresolution rendering of complex botanical scenes. In Proceedings of Graphics Interface 97, pages 97-104, May 1997.

8 J. Hammes. Modeling of ecosystems as a data source for real-time terrain rendering. Springer Velag, LNCS 2181, p. 98. Digital Earth Moving First International Symposium, Proceedings of DEM 2001. p. 98, 2001.

9 S. M. Ervin, H. H. Hasbrouck. LANDSCAPE MODELING: Digital techniques for landscape visualization. McGraw-Hill Professional Publishing (c) 2001 ISBN: 0-07-135745-9

10 F. Neyret, Synthesizing Verdant Landscape Using Volumetric Textures Eurographics Workshop on Rendering. Porto, Portugal, June 1996

11 A. Jakulin. Interactive vegetation rendering with slicing and blending. Proceedings of Eurographics 2000.

12 Ned Greene. Voxel space automata: Modeling with stochastic growth processes in voxel space. Proceedings of SIGGRAPH '89, pages 175-184, July 1989.

13 F. Blaise, J.-F. Barczi, M. Jaeger, P. Dinouard, P. de Reffye. Simulation of the growth of plants – Modeling of metamorphosis and spatial interactions in the architecture and development of plants. Cyberworlds. Springer-Verlag, Tokyo, p. 81-109, 1998

14 B.G. Hu,Ph. de Reffye.Plant growth modeling and visualization. Proceedings of Euro-China Co-operation Forum on the Information Society, April 16-20, Beijing China, 101-110, 2002.

15 H.P. Yan, J. F. Barczi, Ph. de Reffye, B.G. Hu. Fast algorithms of plant computation based on substructure instances. International Conferences in Central Europe on Computer Graphics, Visualization and Computer Vision, 3(10), 145-153, 2002.

16 J. Bloomenthal. Modeling the mighty maple. Proceedings of SIGGRAPH'85, pp.305-311, Jul. 1985

17 H.E. Cline, W.E.,Lorensen, S. Ludke, C.R. Crawford,.B.C. Teeter. Two Algorithms for the 3-Dimensional Construction of Tomograms.In MedicalPhysics,15,3,320-327,(May/June1988).

18 J. Bloomenthal. Poligonzarion of Implicit Surfaces Computer-Aided Geometric Design, 594. pp.341-355, 1998

19 A. Kaufman (ed.), "Volume Visualization",IEEE Computer Society Press, 1990.

20 H. Pfister at al, The transfert Function Bake-Off. Visualization Viewpoints. In: IEEE Computer Graphics and applications, Vol. 21, No. 3. pp. 16-22, May/June 2001.

21 M. Jaeger, J.-M. Leban, S. Chemouny, L. Saint André. 3D stem reconstruction from CT scan exams. Proceedings of 3rd Workshop IUFRO WP S5.01-04. Biological improvement of wood properties, pp. 399-409. september 1999.

22 P. de Reffye, C. Edelin, J. Françon, M. Jaeger and C. Puech. Plant models faithful to botanical structure and development. Proceedings of SIGGRAPH'88, pp. 151-158, July 1988.

23 http://www.bionatics.com

24 F. de Coligny, P. Ancelin, G. Cornu, B. Courbaud, P. Dreyfus, F. Goreaud, S. Gourlet-Fleury, C. Meredieu, L. Saint-André. Capsis : Computer-Aided Projection for Strategies in Silviculture : Advantages of a shared forest-modelling platform. In: A. Amaro, D. Reed and P. Soares

(edsModelling Forest Systems. CABI Publishing, Wallingford, UK, 2002, on press.

25 C.L. Brack. Measuring trees, stands and forests for effective forest management. Computer-based course resources for forest measurement and modeling (FSTY2009) at the Australian National University. http://www.anu.edu.au/Forestry/mensuration/home.htm (1999, 2000, 2001).

26 P. E. Oppenheimer, Real time design and animation of fractal plants and trees. Proceedings of SIGGRAPH'86, pp.55-64, July 1986.

27 J. E. Bresenham. Algorithm for computer control of a digital plotter, In: IBM Systems Journal, vol.4, no 1, pp. 25-30, 1965.

28 SVS. The Stand Visualization System. http://faculty.washington.edu/mcgoy/svs.html.USDA Forest Service. PNW Research Station.

29 M.W. Jones, R.A. Satherley. Shape representation using space filled sub-voxel distance fields. Proceedings of International Conference on Shape Modeling & Applications Genova, Italy, pp. 316-326, May 2001

30 Terragen. http://www.planetterragen.btinternet.co.uk/

31 J. Kniss, P. McCormick, A. McPherson, J. Ahrens, J. Painter, A. Keahey, C. Hansen. Interactive Texture based volume rendering for large data sets. In: IEEE Computer Graphics and Applications. July/August 2001 (Vol. 21, No. 4). pp. 52-61.

32 C. Rezk-Salama, K. Engel, M. Bauer, G. Greiner, and T. Ertl. Interactive volume rendering on standard pc graphics hardware using multi-textures and multi-stage rasterization. In Proc. SIGGRAPH/Eurographics Graphics Hardware Workshop, 2000.