Dynamic Simulation of Natural Environments in Virtual Reality

M.N. Setas*, M.R. Gomes*, J.M. Rebordão**

*INESC - Instituto de Engenharia de Sistemas e Computadores
Grupo de Computação Gráfica e Multimédia
mns@minerva.inesc.pt, mrg@minerva.inesc.pt

**INETI - Instituto Nacional de Engenharia e Tecnologia Industrial
Laboratório de Apoio às Actividades Aeroespaciais

Abstract

This paper presents a set of algorithms to model medium and large scale natural environments. The simulated environments are composed of vegetation dispersed on a topographic surface.

To determine the dynamic response of the vegetation to a wind stimulus a physically based method is applied. We have developed an interpolation technique that reduces the computational cost of this method, permitting real-time simulations.

We have also developed multiple resolution modeling techniques to represent the terrain and the vegetation, reducing the complexity of the rendered scenes. The use of such techniques not only maintains the visual realism of the scenes, but also allows real-time navigation, as is required for their integration in virtual reality systems.

Keywords: Virtual reality, computer animation, natural phenomena, physically based modeling, terrain modeling, vegetation modeling, wind simulation, modal analysis, texture mapping.

1 Introduction

Understanding the complex structure and organization of Nature has always been a challenge for science [1, 2, 3]. The infinity of shapes and behaviors present in our Universe makes it almost impossible to simulate natural environments or phenomena. Particularly in computer graphics, whose algorithms are supposed not only to be efficient, but also to produce realistic results, this goal is even more difficult to achieve.

This paper presents a number of methods and techniques, some of them never used in computer graphics, to model medium and large scale natural scenes.

The scene synthesis is accomplished by superposing on a terrain, represented by a stochastic field, one or more sets of vegetal species modeled by structural or impressionist methods and distributed according to point patterns.

The scene animation is based on a physically based simulation of the dynamic response of the vegetation to a wind stimulus. We have developed an interpolation technique that reduces the computational cost of this method, permitting the inclusion of plants in virtual reality systems as semi-autonomous agents, exhibiting dynamic behavior.

Moreover, to reduce the complexity of the simulated scenes, we use a simple but efficient partitioning scheme of the scenes and multi-resolution representations of their elements. This approach allows real-time navigation through the animated scenes, which is indispensable for their integration in virtual reality systems.

2 Scene Modeling

In the context of this paper a natural scene is understood as a set of vegetal elements superposed on a topographic surface, in a scale ranging roughly from 10m to 10km.

The use of stochastic fields for terrain modeling has produced interesting results [4, 5]. In this paper we introduce an efficient and flexible algorithm, called *Turning Bands Method* [6, 7], to generate 2D random fields.

Concerning the vegetation, we model its distribution on the terrain with point patterns generated with an algorithm

known as the *Quadrat Method* [8], generally used for statistical analysis of spatial dispersion. The vegetation itself is simulated with three methods based on structural [9, 10] and impressionist [2] models.

Hence, the process of modeling a scene is composed of three distinct phases: generation of one or more random fields to model the terrain; generation of one or more point patterns to model the distribution of vegetal species on the terrain and generation of 3D representations of vegetal species.

2.1 Terrain Modeling

To generate 2D random fields we have implemented the *Turning Bands Method* (*TBM*) [6, 7], whose application in computer graphics is unknown, but which has been used in many geophysical sciences like hydrology [7], and more recently in remote sensing to simulate medium and large scale data fields [11, 12]. These fields can represent realistic topographic surfaces, as well as their reflectance variability, which is usually relevant in vegetated environments.

The main advantage of the *TBM* is to transform a multidimensional stochastic process into the sum of a series of equivalent unidimensional stochastic processes, and consequently reduce the computational cost. So, instead of simulating the two-dimensional field directly, simulations are performed along several lines, using an unidimensional covariance function that corresponds to the desired two-dimensional one.

Another important advantage of this method is that it can be used to generate fields with different covariance structures [7, 13].

The method can be briefly described as follows (see Figure 1):

ALGORITHM 1 (*TURNING BANDS METHOD*)

1) Choose an arbitrary origin, O, in $\Re^2$;

2) Distribute uniformly on the unit circle L direction vectors, $\mathbf{u}_i$, corresponding to the lines along which the unidimensional stochastic processes will be generated;

3) Along each line i, generate a second-order stationary unidimensional discrete process having zero mean and a specified covariance function, $C_1(\zeta)$, where ζ is the coordinate along line i.

4) Project orthogonally onto line i those points of the field where we want to generate values, and assign to them the corresponding values of the unidimensional stochastic process.

For instance, if we consider the N^{th} point of the field, with location vector $\mathbf{x}_N$, the assigned value from line i is $z_i(\zeta_{N_i})$ where $\zeta_{N_i} = \mathbf{x}_N \cdot \mathbf{u}_i$ is the projection of the vector $\mathbf{x}_N$ onto line i.

5) Finally, after repeating this process for all the lines, assign to each point of the field a value that is a weighted sum of the values obtained in 4):

$$z_s(\mathbf{x}_N) = \frac{1}{\sqrt{L}} \sum_{i=1}^{L} z_i(\mathbf{x}_N \cdot \mathbf{u}_i). \quad (1)$$

The subscript s represents the term 'simulated' or 'synthetic'.

□

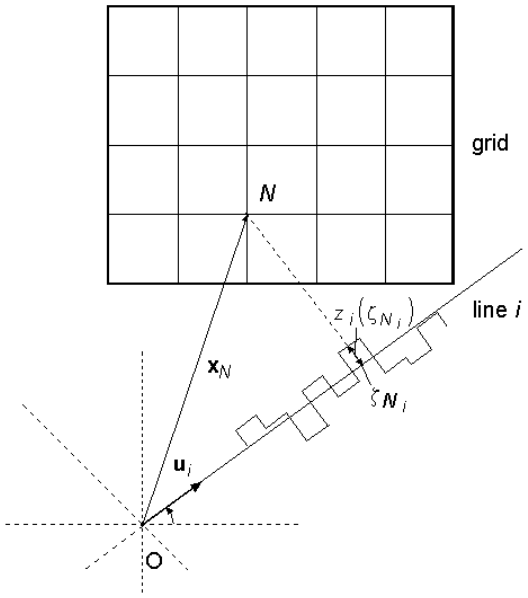


Figure 1: Representation of the *Turning Bands Method*.

The simulated two-dimensional covariance, $C_s(\mathbf{h})$, is related to the covariance of the unidimensional process, $C_1(\zeta)$, by the expression:

$$C_s(\mathbf{h}) = \int_c C_1(\mathbf{h} \cdot \mathbf{u}) p(\mathbf{u})\, d\mathbf{u}, \quad (2)$$

where c represents the unit circle, $\mathbf{h}$ is the vector connecting two points of the field, $p(\mathbf{u})$ ($1/2\pi$ for isotropic fields) is the probability density function of $\mathbf{u}$ (direction of the lines).

LINE PROCESS

According to Mantoglou and Wilson [7], the unidimensional stochastic process on line i can be generated by:

$$z_i(\zeta) = 2\sum_{k=1}^{M}\left[S_1(\omega_k)\Delta\omega\right]^{1/2}\cos\left(\omega_k'\zeta + \phi_k\right), \quad (3)$$

where ϕ_k are independent random phases uniformly distributed between 0 and 2π, $\omega_k = (k-1/2)\Delta\omega$ and $\omega_k' = \omega_k + \delta\omega$ for $k = 1,\ldots,M$. It is assumed that the unidimensional spectral density function, $S_1(\omega)$ (Fourier pair of $C_1(\zeta)$), is insignificant outside the interval $[-\Omega,\Omega]$. The discretization frequency is defined as $\Delta\omega = \Omega/M$, where M is the number of harmonics used. The frequency $\delta\omega$ is a small random frequency added to avoid the periodicity of the simulated process and is uniformly distributed between $-\Delta\omega'/2$ and $\Delta\omega'/2$, with $\Delta\omega' \ll \Delta\omega$.

Mantoglou and Wilson [7] showed that the unidimensional spectral density function, $S_1(\omega)$, is simply given by one half of the radial spectral density function, $f(\omega)$, of the two-dimensional isotropic process, multiplied by the variance:

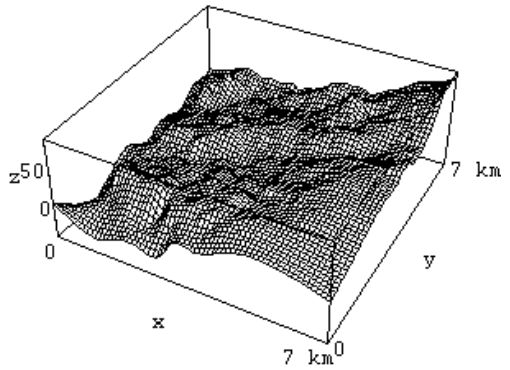
$$S_1(\omega) = \frac{\sigma^2}{2} f(\omega). \quad (4)$$

For example, the radial spectral density function for a fractal field can be written as [13]:

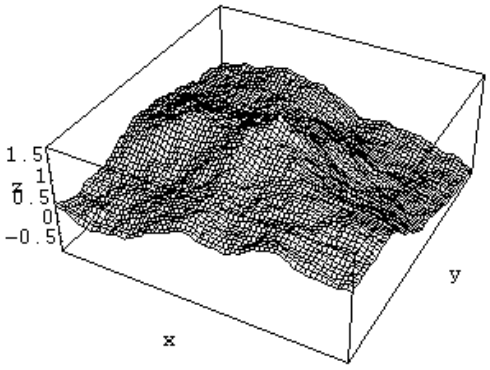
$$f(\omega) \propto \frac{1}{\omega^{7-2D}}, \quad (5)$$

where D is the fractal dimension.

Mantoglou and Wilson [7] present other radial spectral density functions and the corresponding covariance functions: Simple Exponential, Double Exponential, Bessel 1, Bessel 2, Telis and Spherical.



(a)



(b)

Figure 2: (a) Real terrain; (b) Simulated field with the *Turning Bands Method* (same fractal dimension as the real terrain, D = 2.15).

Using the *Levenberg-Marquardt Method* [14], which is a nonlinear least-squares algorithm, to test the fitness of each of the available theoretical models, we can characterize the covariance structure of real topographic data. Choosing the best covariance model, the *Turning Bands Method* can be used to generate synthetic fields with the same statistical properties of real terrains.

2.2 Vegetation Modeling

2.2.1 Point Patterns

The *Quadrat Method* [8], mainly used by ecologists and geographers, is a technique of pattern analysis that can be either applied to characterize or generate point patterns, according to a number of statistical models. The available statistical models permit the characterization or generation of random (Poisson), regular (Binomial) or clustered (Negative Binomial, Poisson-Poisson, Poisson-Binomial, Poisson-Negative Binomial) point patterns [8]. The Poisson, Binomial and Negative Binomial distributions are the most commonly used. Their probability density functions are, respectively:

$$P(r) = \exp(-v)\frac{v^r}{r!}, \quad (v = Q/N), \quad (6)$$

$$P(r) = \binom{n}{r} p^r (1-p)^{n-r}, \quad (7)$$

$(n \text{ integer}, 0 < p < 1, np = Q/N)$,

$$P(r) = \binom{k+r-1}{r} p^r (1+p)^{-k-r}, \quad (8)$$

$(k, p > 0, kp = Q/N)$,

where Q is the number of quadrats and N is the number of points.

ALGORITHM 2 (*QUADRAT METHOD*)

1) Superpose on the study area a regular grid of cells (quadrats). Generally, the number of quadrats, Q, is approximately the same as the number of points, N.

To generate a point pattern:

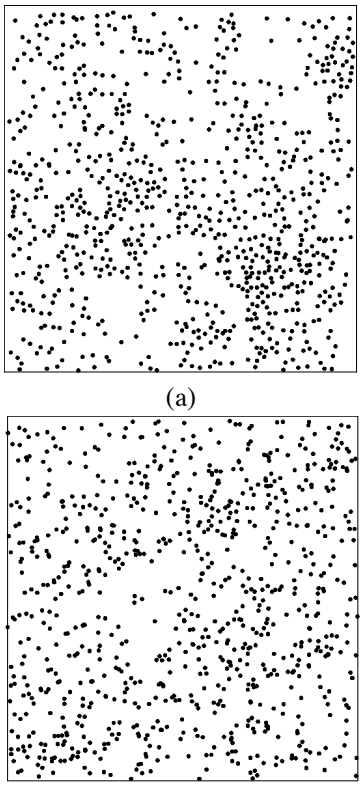
2) Calculate the theoretical frequency distribution, $QxP(r)$, which gives the amount of quadrats with r points.

3) Distribute randomly 1 point in each of the $QxP(r=1)$ quadrats, 2 points in each of the $QxP(r=2)$ quadrats, and so on.

To characterize a point pattern:

2) Determine the observed frequency distribution, counting the number of quadrats that contain r (0, 1, 2, ...) points.

3) Use an estimation algorithm, such as the method of the moments [8] or the method of maximum likelihood [8], to find the best theoretical model and its parameters. □



(a) (b)

Figure 3: (a) Vegetation distribution on part of the terrain of Figure 2(a); (b) Simulated point pattern with the *Quadrat Method*, (same spatial dispersion model as the real pattern - Poisson).

Through the analysis of aerial photographs or high-resolution remote sensing pictures we can characterize the spatial distribution of vegetal species in terms of its statistical properties. Using this method we simulate spatial arrangements of plants similar to those existing in Nature.

2.2.2 Vegetation Models

The visual quality of a synthetic scene that simulates a natural environment is obviously very dependent on the realism and accuracy of the vegetation models. In this work we have used an impressionist method similar to Gardner's [2] model, a structural and stochastic method based on Reeves' model [9], and a structural method based on Aono's model [10]. With these three methods we are able to simulate a wide variety of vegetal species.

To generate a set of plants of the same species we add a statistical variation to the simulation parameters to reproduce the natural variation of form and aspect.

3 Scene Animation

The dominant causes of motion in natural scenes are of climacteric origin. The wind is the most frequent cause of motion. We have included in our simulation the dynamic response of the vegetation to a wind stimulus.

The wind is modeled by a spectral synthesis algorithm, which takes into account its spatio-temporal characteristics.

The plants are considered as hierarchical structures of deformable uniform beams, whose motion is determined by modal analysis, commonly applied to the study of the dynamics of buildings [15, 16].

3.1 Wind Simulation

Stochastic approaches are normally effective for modeling natural phenomena. We simulate the wind with an algorithm imported from wind engineering [17], introduced in computer graphics by Shynia [18], that synthesizes 3D wind velocity fields using experimental power spectra and cross-spectra of wind and that has proved to be an efficient technique.

3.1.1 Velocity

There are several experimental models of the power spectra of wind for isotropic a homogeneous conditions [17]. We have used a simple model that well matches experimental data. According to the model, the power spectra for u (mean wind direction), v (horizontal direction

perpendicular to the *u* direction) and *w* (vertical direction) components are, respectively:

$$S_u = u_*^2 \cdot 200 \cdot (\nu / f) / (1 + 50\nu)^{5/3}, \quad (9.a)$$

$$S_v = u_*^2 \cdot 15 \cdot (\nu / f) / (1 + 9.5\nu)^{5/3}, \quad (9.b)$$

$$S_w = u_*^2 \cdot 3.36 \cdot (\nu / f) / (1 + 10\nu^{5/3}), \quad (9.c)$$

where u_*^2 is the variance of the *u* component, *f* is the temporal frequency in Hz, $\nu = fz/U(z)$, *z* is the height of the observation point from the ground, $U(z)$ is the mean wind velocity, generally given by the *logarithmic law*:

$$U(z) = \frac{1}{k} u_* \ln \frac{z}{z_0}, \quad (10)$$

where *k* is the von Karman constant (0.4), u_* is the friction velocity, z_0 is the roughness length, which is a measure of the soil irregularity. Typical values of z_0 range from 0.01m for low grass to 1m for a pine forest [17].

The cross spectra of *u*, *v* and *w* components, at two points $P_1 = (x_1, y_1, z_1)$ and $P_2 = (x_2, y_2, z_2)$, can be expressed by:

$$S_u^c = \sqrt{S_u(z_1)} \sqrt{S_u(z_2)} \exp(-\hat{f}_u), \quad (11.a)$$

$$S_v^c = \sqrt{S_v(z_1)} \sqrt{S_v(z_2)} \exp(-\hat{f}_v), \quad (11.b)$$

$$S_w^c = S_w(z_1) \exp(-\hat{f}_w), \quad (11.c)$$

with:

$$\hat{f}_u = f \sqrt{c_{1z}^2 (z_1 - z_2)^2 + c_{1y}^2 (y_1 - y_2)^2} \Big/ U(10), \quad (12.a)$$

$$\hat{f}_v = f \sqrt{c_{2z}^2 (z_1 - z_2)^2 + c_{2y}^2 (y_1 - y_2)^2} \Big/ U(10), \quad (12.b)$$

$$\hat{f}_w = 8f |y_1 - y_2| / U(z). \quad (12.c)$$

The exponential decay coefficients are empirically determined and have typical values between 2 an 10.

We use these spectra and cross spectra to synthesize 3D vector fields that represent the wind velocity. The algorithm is described below.

ALGORITHM 3

1) Initialize three complex data arrays ($wind_u$ (*f, y, z*), $wind_v$ (*f, y, z*) and $wind_w$ (*f, y, z*) of dimension *N* (power of 2)) with the cross spectra given, respectively, by 11.a, 11.b, 11.c.

2) Apply an FFT with *y* and *z* to reduce the spatial dependence to the frequency domain.

3) Multiply each complex component of the arrays by $\exp(i\phi)$, where ϕ is a Gaussian random variable uniformly distributed between 0 and 2π. Since the wind velocity field is real, a set of conjugate symmetry conditions for each array and for $1 \le i, j, k \le N/2$ must be satisfied:

$$wind[N-i][N-j][N-k] = wind^*[i][j][k], \quad (13.a)$$

$$wind[i][N-j][N-k] = wind^*[N-i][j][k], \quad (13.b)$$

$$wind[N-i][j][N-k] = wind^*[i][N-j][k], \quad (13.c)$$

$$wind[N-i][N-j][k] = wind^*[i][j][N-k]. \quad (13.d)$$

Moreover for $i, j, k = 1$ and $N/2$:

$$wind[i][j][k].imaginary = 0. \quad (13.e)$$

4) The wind velocity field is obtained by applying the inverse FFT to the three components of the three arrays.

5) Finally, normalize the velocity field with the specified variance.

□

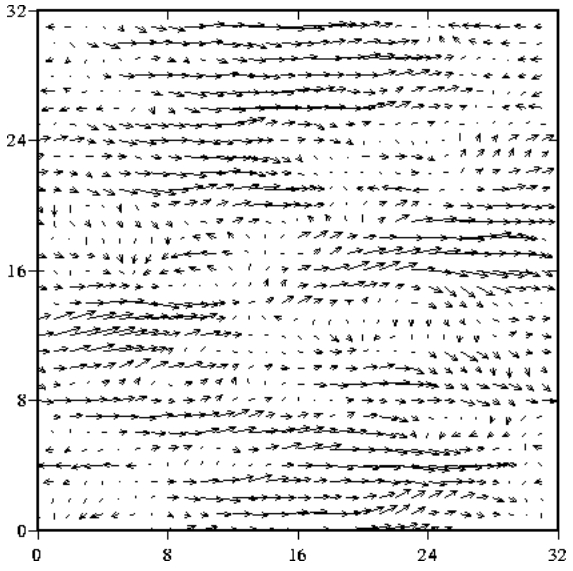


Figure 4: An example of a wind velocity field.

One assumption usually accepted, when dealing with turbulent phenomena, is the *Taylor's Hypothesis* which assumes that the turbulence travels at the mean velocity of the flow:

$$u(x, y, z, t) = u(x - t*U, y, z, 0), \quad (14)$$

where *u* is the wind velocity, *U* is the mean wind velocity, *t* is time and (*x*, *y*, *z*) is the space position. So, we can consider this problem as a 3D problem instead of a 4D one. Moreover, as this algorithm uses the FFT to

accomplish the spectral synthesis, it generates periodic fields. We take advantage of these two convenient facts to produce continuous animation sequences, using a single 3D wind velocity field.

3.1.2 Induced Forces

The wind model permits the generation of velocity fields, however, to determine the dynamic response of vegetation to a wind stimulus we need to know the forces induced by the wind. There are two major forces produced by a wind gust. The force in the direction of the flow is conventionally called drag, on the other hand, the force normal to the flow is generally called lift.

For a mean wind velocity U the drag and lift forces can be written as:

$$F_D = \frac{1}{2}\rho U^2 A C_D, \tag{15}$$

$$F_L = \frac{1}{2}\rho U^2 A C_L, \tag{16}$$

where ρ is the density of air, A is some typical dimension of the object (for a cylinder, for instance, the diameter), C_D and C_L are the drag and lift coefficients, respectively. Both the drag and lift coefficients depend on the shape of the object and on the angle of attack, that is the angle between the wind direction and the object [17].

3.2 Dynamics of the Vegetation

To simulate the behavior of a plant under the action of a generic force, which can be the drag and lift forces produced by wind or even the gravity force, the stems, trunks or branches are approximated by uniformly mass-distributed beams linked in a tree-type structure, and the leaves are seen as massless plates.

3.2.1 Branches

Modal analysis is a powerful method to determine the solution of vibrating systems, and in particular of damped forced oscillators. As beams exhibit an oscillatory behavior, modal analysis can be used to determine their dynamic response to external stimulus, as Shynia has already shown [18].

The essential operation of modal analysis is the transformation from the geometric displacement coordinates to the modal amplitude or normal coordinates. For a one-dimensional continuum this transformation is expressed as [15]:

$$v(x,t) = \sum_{n=1}^{\infty} \phi_n(x) Y_n(t), \tag{17}$$

in which $\phi_n(x)$ is a mode-shape function and $Y_n(t)$ is a modal amplitude function. This mode-superposition analysis states that any physically permissible displacement pattern can be made up by superposing appropriate amplitudes of the vibrating mode shapes of the structure. Furthermore, the solution is generally well approximated by the fundamental mode (n = 1), which contributes to the simplification of the problem.

For a cantilever beam with length L, the mode functions are given by [16]:

$$\phi_n(x) = A_1\Big[sin(\eta_n x) - sinh(\eta_n x) + \\ + \overline{D}_n\big(\cos(\eta_n x) - \cosh(\eta_n x)\big)\Big], \tag{18}$$

where:

$$\overline{D}_n = \frac{\cos(\eta_n L) + \cosh(\eta_n L)}{sin(\eta_n L) - sinh(\eta_n L)}, \tag{19}$$

and A_1 is an arbitrary constant. The product $\eta_n L$ is the solution of a transcendental equation [16]:

$$\eta_1 L = 1.875, \eta_2 L = 4.694, \eta_3 L = 7.855, \ldots, \eta_n L \approx \left(n - \frac{1}{2}\right)\pi$$

and the natural frequencies are:

$$\omega_n = \eta_n^2 \sqrt{\frac{EI_z}{\rho A}}, \tag{20}$$

where E is the *Young's modulus*, I_z is the moment of inertia of the cross sectional area (for a cylinder of radius R, $I_z = \pi R^4$), ρ is the density and A is the cross sectional area of the beam.

The modal amplitude functions for a damped system are solutions of the following equation:

$$\ddot{Y}_n(t) + 2\xi_n \omega_n \dot{Y}(t) + \omega_n^2 Y_n(t) = \frac{P_n(t)}{M_n}, \tag{21}$$

where n is the mode number, ξ_n is the damping ratio, $P_n(t)$ and M_n are the generalized load and mass, respectively. The generalized load is:

$$P_n(t) = \int_0^L \phi_n(x) p(x,t) dx, \tag{22}$$

where $p(x,t)$ is the load along the beam over time, and the generalized mass is:

$$M_n = \int_0^L \phi_n^2(x) m(x) dx, \quad (23)$$

where $m(x)$ is the density along the beam.

The solution of equation (21) can be calculated with the *Duhamel integral* [15]:

$$Y_n(t) = \frac{1}{\omega_{D_n} M_n} \cdot \int_0^t P_n(\tau) \exp\left(-\xi_n \omega_n (t-\tau)\right) sin\left(\omega_{D_n}(t-\tau)\right) d\tau \quad (24)$$

in which $\omega_{D_n} = \omega_n \left(1-\xi_n^2\right)^{1/2}$.

To simulate the dynamic response of a structure of branches to a wind stimulus it is necessary to calculate, for each time step and for each branch (if they are different), the solution of the *Duhamel integral* (expression (24)), multiply it by the corresponding mode-shape (expression (18)), and add these products for all the considered modes (expression (17)). To simplify the simulation we have considered only the fundamental mode (n = 1) and have also neglected the coupling force between two branches.

AN APPROXIMATE SOLUTION

To produce real-time simulations of a series of trees and plants, with hundreds or thousands of branches each, the animation process has to be optimized, particularly the calculation of the *Duhamel integral*. We present an approximation based on the interpolation of the dynamic responses of three orthogonal beams.

As referred to in section 3.1.1 the wind velocity fields generated by algorithm 3 are periodic, which means that the wind load applied to each branch and the corresponding dynamic response, after a transition period, are also periodic. Hence, if we pre-compute the modal amplitude (expression (24)) of three orthogonal beams, the modal amplitude of any beam can be determined, during the animation process, with a simple expression:

$$\vec{Y_{Beam}} = \left(\vec{Y_X} + \vec{Y_Y} + \vec{Y_Z}\right) - \left(\vec{Y_X} \cdot \vec{u_{Beam}} + \vec{Y_Y} \cdot \vec{u_{Beam}} + \vec{Y_Z} \cdot \vec{u_{Beam}}\right) \vec{u_{Beam}}, \quad (25)$$

where $\vec{Y_{Beam}}$, $\vec{Y_X}$, $\vec{Y_Y}$ and $\vec{Y_Z}$ are, respectively, perpendicular vectors to the generic beam and to the three orthogonal beams, whose lengths are the corresponding modal amplitudes. $\vec{u_{Beam}}$ is a normalized vector that represents the orientation of any generic beam. Basically, $\vec{Y_{Beam}}$ is equal to the sum of the projections of $\vec{Y_X}$, $\vec{Y_Y}$ and $\vec{Y_Z}$ onto a perpendicular plane to the beam's orientation.

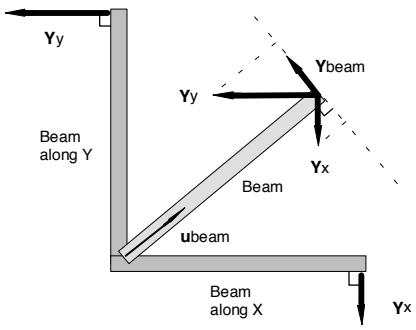


Figure 5: Interpolation scheme (depicted in 2D) used to determine the modal amplitude of any beam.

This approximation permits the animation of any tree-type structure of similar beams, after pre-computing the solution for only three beams. Depending on the complexity of the tree-type structures, this technique can be used to produce real-time simulations, allowing the inclusion of plants in virtual reality systems as semi-autonomous agents with a pre-defined behavior.

3.2.1 Leaves

We have developed an empirical method to animate the leaves. They are considered as massless plates with two perpendicular axes: the main one and the secondary one. When the wind is sufficiently strong the main axis tends to be aligned with the wind direction. Thus, its spherical coordinates, (θ_m, φ_m), can be determined by:

$$\theta_m = \theta_r + k_1(u).(\theta_u - \theta_r) + k_2(u).\delta_\theta, \quad (26.a)$$

$$\varphi_m = \varphi_r + k_1(u).(\varphi_u - \varphi_r) + k_2(u).\delta_\varphi, \quad (26.b)$$

where (θ_u, φ_u) is the wind direction and (θ_r, φ_r) is the main axis direction at rest. $k_1(u)$ and $k_2(u)$ are two generic functions (step functions, for instance), with values between 0 and 1, that depend on the wind velocity. δ_θ and $\delta\varphi$ are two random variables uniformly distributed between $-\Delta/2$ and $\Delta/2$, where Δ is a small angle.

The direction of the secondary axis is random, but is orthogonal to the main one.

4 Application to Virtual Reality

The computational power existing today is generally insufficient to have both visual realism and real-time immersion and interaction i virtual reality system. This fact has been strongly conditioning the kind of behavior that active agents have in virtual reality environments. In this paper we try to give a contribution in that direction. Our particular interest was to populate a natural outdoor scene with plants, mainly trees, capable of swaying under the influence of wind, and so, we have adapted the algorithms used for modeling and animation to permit their application in VR.

One first step to reduce the number of polygons sent to the graphics pipeline, as well as the number of agents (in our case plants) to animate, is to divide the scene into a regular grid of cells (voxels) and process only the data in the cells belonging to the instantaneous view volume.

A rule of thumb to subdivide the scene is to have approximately the same number of objects and cells.

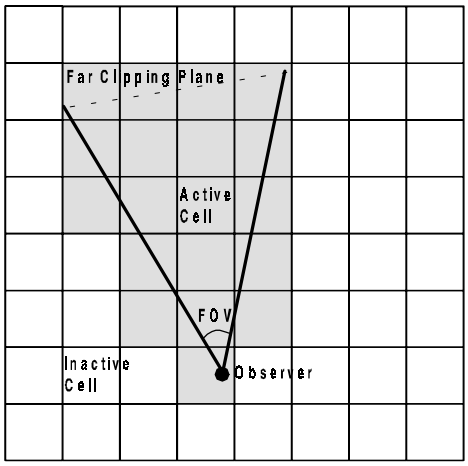


Figure 6: Scene subdivision.

Another way of improving the system's performance is to represent the elements of the scene with different levels of detail, depending on their distance from the view point.

4.1 Multi-Resolution Models

The use of multi-resolution modeling techniques is an efficient form of reducing the number of polygons passed to the graphics pipeline without losing visual realism, and has been successfully applied in virtual worlds [19, 20].

4.1.1 Terrain

Concerning the terrain we have used a simple pyramid representation to implement the multi-resolution of its elevation data.

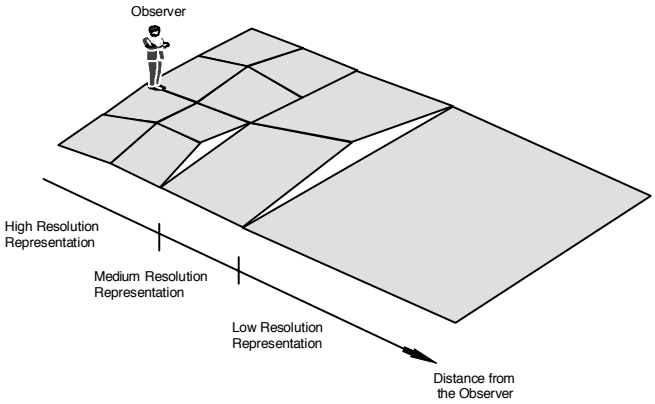


Figure 7: Multi-resolution representation of the terrain.

The distance thresholds where the changes of resolution occur are determined by trial and error. Each gap generated by a change of resolution (white triangles in Figure 7), when visible, must be filled with a triangle. Although this triangle is vertical, it is not noticeable since its vertices share the same normals as the adjacent polygons.

4.1.2 Vegetation

We have developed a method that uses three levels of detail to model vegetation and that is specially applicable to Reeves' stochastic trees [9].

The 3D high resolution representation is directly produced by the generation algorithm.

As Reeves' model uses a bounding volume (a cone or an ellipsoid) to limit the branches' growth, the medium resolution representation consists of a trunk (cone) and an envelope of leaves with the shape of the bounding volume. To increase the realism, this envelope is texture mapped with a translucent cylindrical projection of the 3D high resolution representation, simulating the foliage. The cylindrical projection is performed by ray-tracing, with a camera that spins around the plant.

The low resolution representation consists only of four non-textured polygons: two perpendicular rectangles to depict the trunk, and two perpendicular triangles (bounding volume - cone) or ellipses (bounding volume - ellipsoid) to depict the crown.

To obtain better performances we often only use the medium and low resolution representations. When following this approach we add some randomness to the crown's vertices (only in the medium resolution representation) to increase the boundary irregularity.

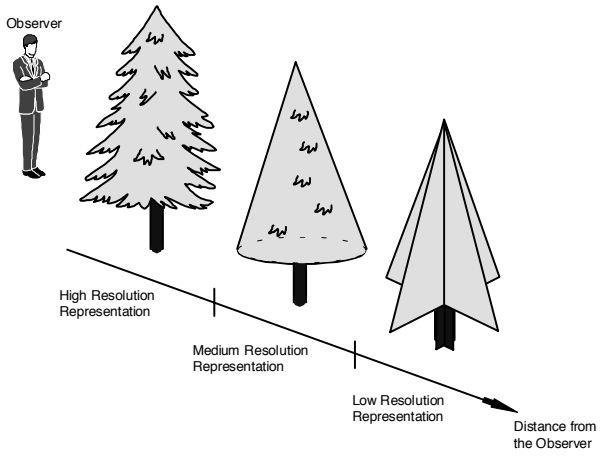


Figure 8: Multi-resolution representation of the vegetation.

4.2 Animation

The animation of the 3D high resolution representation is performed as described in section 3.2.

Concerning the medium resolution representation, its animation consists of two phases: the oscillation of the envelope of leaves and the deformation of the texture. To maintain visual coherence between the medium and high resolution representations the envelope of leaves oscillates as the main trunk of the high resolution model does. The deformation of the texture is obtained by generating a set of images of the high resolution representation in different positions of the animation sequence.

The low resolution representation, as it is used to depict trees considerably distanced from the observer, is not animated.

5 Results

We have obtained considerable gains in performance with the mentioned techniques.

Figure 9 represents the dependence of the frame rate with the complexity of the scene (number of trees). Using the scene partitioning scheme and the multi-resolution model, the frame rate (12.2 s^{-1}), for the more complex case (1000 trees), is approximately 13 times greater than the one obtained when using neither of those techniques (0.9 s^{-1}). In these particular tests the importance of the scene subdivision and of the different levels of detail seems roughly equivalent (frame rate of 5.2 s^{-1} using only voxels and 3.7 s^{-1} using only different levels of detail).

These tests were performed on an SGI Indigo IRIS XS24 with GL. The scenes had a flat terrain and the multi-resolution representation of the trees had only two levels of detail (without textures). The frame rates were measured with a 3 minutes' random path through the scenes.

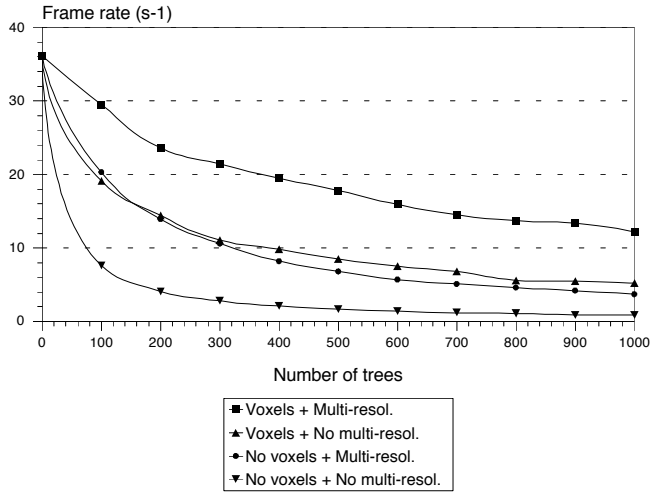
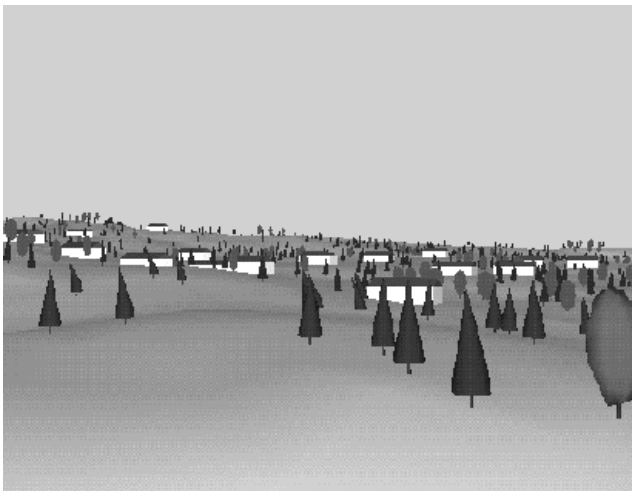
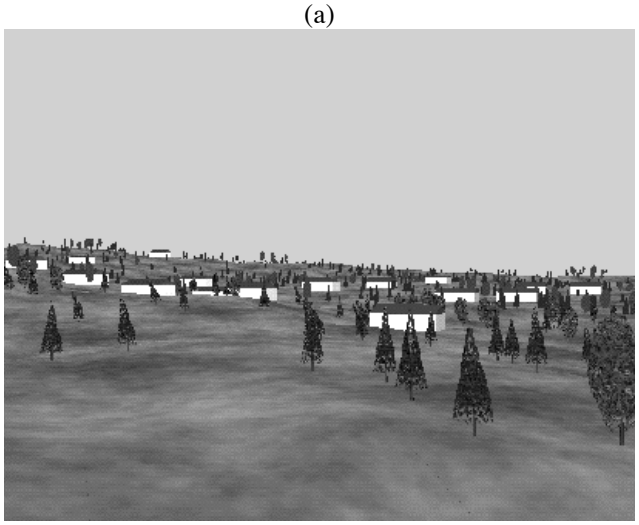


Figure 9: Frame rate versus Number of trees in the scene.



(a)



(b)

Figure 10: An example of a virtual scene. (a) Non-textured; (b) Textured.

The scene depicted in Figure 10 has over a thousand trees of two different species, with two levels of detail, and a few houses dispersed on a smooth terrain.

6 Conclusions

We have introduced two new methods for natural scene modeling. The *Turning Bands Method* to generate stochastic fields, that we use in terrain modeling, and the *Quadrat Method* to generate point patterns, that simulate the distribution of plants on the terrain.

We have used a physically based method to produce the animation of the modeled scenes under the influence of wind. A simplification of the method has reduced its computational cost, permitting the performance of real-time simulations.

We have also implemented a multi-resolution scheme for representing both the terrain and the vegetation, which have reduced significantly the amount of polygons sent to the graphics pipeline.

Hence, we have managed to model natural scenes whose elements exhibit a dynamic behavior, and we have improved the modeling and animation algorithms, allowing the integration of these scenes in a virtual reality system currently being developed in INESC.

Moreover, it has been shown that the use of stochastic and statistical algorithms together with physically based methods can produce interesting results in natural scene and phenomena modeling.

Acknowledgments

This work was supported by PRAXIS XXI.

References

[1] B. Mandelbrot, "The Fractal Geometry of Nature", W. H. Freeman and Co., 1982.

[2] G.Y. Gardner, "Simulation of Natural Scenes Using Textured Quadric Surfaces", Computer Graphics (ACM SIGGRAPH '84), **18**(3), pp. 11-20, July 1984.

[3] A. Fournier, "The Modelling of Natural Phenomena", Graphics Interface '89, pp. 191-202, 1989.

[4] M.F. Barnsley, R.L. Devaney, B.B. MandelBrot, H.-O. Peitgen, D. Saupe, R. Voss, "The Science of Fractal Images", Springer-Verlag, 1988.

[5] F.K. Musgrave, C.E. Kolb, R.S. Mace, "The Synthesis and Rendering of Eroded Fractal Terrain", Computer Graphics (ACM SIGGRAPH '89), **23**(3), July 1989.

[6] G. Matheron, "The Intrinsic Random Functions and Their Applications", Adv. Appl. Prob., **5**, pp. 439-468, 1973.

[7] A. Mantoglou, J.L. Wilson, "The Turning Bands Method for Simulation of Random Fields Using Line Generation by a Spectral Method", Water Resources Research, **18**(5), pp. 1379-1394, 1982.

[8] A. Rogers, "Statistical Analysis of Spatial Dispersion - the quadrat method", Pion Limited, 1974.

[9] W. Reeves, R. Blau, "Approximate and Probabilistic Algorithms for Shading and Rendering Structured Particle Systems", Computer Graphics (ACM SIGGRAPH '85), **19**(3), pp. 313-322, July 1985.

[10] M. Aono, T.L. Kunii, "Botanical Tree Image Generation", IEEE CG&A, **4**(5), pp. 10-34, 1984.

[11] M.F. Jasinski, P.S. Eagleson, "The Structure of Red-Infrared Scattergrams of Semivegetated Landscapes", IEEE Transactions on Geoscience and Remote Sensing, **27**(4), pp. 441-451, 1989.

[12] M.F. Jasinski, "Sensitivity of the Normalized Difference Vegetation Index to Subpixel Canopy Cover, Soil Albedo, and Pixel Scale", Remote Sensing of Environment, **32**, pp. 169-187, 1990.

[13] M.N. Setas, J.M. Rebordão "Modeling Anisotropic and Fractal Fields with the Turning Bands Method", submitted to IEEE CG&A.

[14] W.H. Press, B.P. Flannery, S.A. Teukolsky, W.T. Vetterling, "Numerical Recipes in C", Cambridge University Press, 1988.

[15] R.W. Clough, J. Penzien, "Dynamics of Structures", McGraw-Hill, 1975.

[16] A.A. Shabana, "Theory of Vibration - Volume II: Discrete and Continuos Systems", Springer-Verlag, 1991.

[17] E. Simiu, R.H. Scanlan, "Wind Effects on Structures: An Introduction to Wind Engineering", John Wiley & Sons, 1978.

[18] M. Shinya, A. Fournier, "Stochastic Motion - Motion Under the Influence of Wind", Eurographics '92, **11**(3), pp. 119-128, 1992.

[19] J.S. Falby, M.J. Zyda, D.R. Pratt, R.L. Mackey, "NPSNET: Hierarchical Data Structures for Real-Time Three-Dimensional Visual Simulation", Computer & Graphics, **17**(1), pp. 65-69, 1993.

[20] D.R. Pratt, "A Software Architecture for the Construction and Management of Real-Time Virtual Worlds", PhD Dissertation, June 1993.