Real-Time Rendering and Animation of Vegetation

Ralf Habel
Institute of Computer Graphics and Algorithms
Vienna University of Technology

***Abstract*— Vegetation in all its different forms is almost always part of a scenery, be it fully natural or urban. Even in completely cultivated areas or indoor scenes, though not very dominant, potted plants or alley trees and patches of grass are usually part of a surrounding. Rendering and animating vegetation is substantially different from rendering and animating geometry with less geometric complexity such as houses, manufactured products or other objects consisting of largely connected surfaces.**

In this paper we will discuss several challenges posed by vegetation in real-time applications such as computer games and virtual reality applications and show efficient solutions to the problems.

I. Introduction

Vegetation in computer graphics can be roughly categorized into the field of modeling the growth of a plant by generating its geometry, and the field of modelling the appearance and behavior of plants in an environment. Though real plants all basically use the same processes to grow, a plethora of methods can be applied to generate plants at various ages, ranging from fractals [1], L-Systems [2] and procedural approaches [3] to full simulations of ecosystems [4], among others. To display and to animate this generated geometry interactively, specialized representations, lighting and shading techniques together with animation or simulation methods are applied to incorporate the non-geometric attributes of vegetation. Of course, both fields are strongly connected since the environment impacts the growth of a plant [5]. Also, geometric representation and lighting or shading techniques are heavily dependent on each other since geometric attributes need to be transported by the representation in order to have them available for shading.

Though the generation of plants has received more attention than other aspects of vegetation, only the combination of accurate geometry, appearance and dynamic behavior results in a convincing result. Especially under real-time conditions, all facets of displaying vegetation pose significant problems, which makes interactive rendering and animation of vegetation one of the biggest challenges in real-time graphics.

II. Challenges

The term vegetation is a broad term, covering structures such as lawns up to complete landscapes covered with a forest. Many computer games and virtual reality applications are already very realistic, though most lack a realistic display of plants and trees due to their inherent complexity. Especially for trees and grass, many standard acceleration and simplification methods cannot be applied. This results in severe compromises in the realism of their appearance compared to other parts of a scene. There are several reasons why vegetation is more difficult to display than other objects:

A. Light Interaction

Vegetation is not only complex in geometry, also the light interaction of leaves or grass blades is highly intricate. A leaf for example usually consists of different layers and is strongly structured, which has a profound impact on both the reflectance and translucency of leaves, an integral part of the light interaction of vegetation. Additionally, many leaves differ not only between species but also in their light transport on the front and back, depending on the nature of the surface, and no general assumptions can be made.

B. Geometric Complexity

Concerning trees and treelike plants, it is possible to use a full geometry representation on current hardware, though only a limited amount of polygons can be spent on each branch and leaf depending on the corresponding size and shape, also limiting the number of branches to a few thousand and the number of leaves to a few ten thousand.

A tree in full geometry representation poses challenges to create realistic animations under the given real-time constraint since every branch and leaf is perceived as a separate part and thus needs to be treated separately. The structure of a tree consists of a complex hierarchy of branches to which leaves are attached, all of which interact with a turbulent wind field, and every part of the tree must react consistently to wind in order to achieve a realistic and convincing animation of a complete tree.

III. Leaf Rendering

The rendering of leaves in commercial applications such as games and virtual reality simulations is usually avoided completely by representing trees as billboards or billboard clouds. This means that there are no separate leaves and the textures used for the billboard are rendered with standard methods, not taking care of special attributes of leaves. This can already produce somewhat good results if the textures are generated so they benefit the appearance of leaves [6]. But this approach does not reproduce the behavior of leaves in light and can therefore provide only very limited realistic results.

Leaves have a very complex interaction with light and only few assumptions can be made since there is a large variety of leaves. They differ not only in shape and color, but also in surface attributes, ranging from highly glossy surfaces due to thick wax layers to completely diffuse surfaces due to micro hairs. Also, leaves usually show very different light interactions on the adaxial and abaxial side. But the most defining attribute of leaves differentiating them from other surfaces is their translucency, which becomes very apparent in direct sunlight when seen from the unlit side (see Figure 1).

Fig. 1. Leaves in sunlight.

Another research area where light-leaf interaction is important is remote sensing, which is usually done by satellite or radar. In order to derive values such as vegetation covering of a landscape, health of plants, water containment of plants, etc., from measurements, accurate models of reflectance, translucency and general light transport inside plants or canopies are required to extrapolate such data. Though those models are targeted to derive biophysical and agricultural properties, they can also be applied to computer graphics. An extensive overview of optical properties in the context of remote sensing can be found in [7].

A realistic leaf can not be modeled using standard methods due to the intricate light-leaf interaction, and specialized methods have to be applied to render convincing vegetation since an important part of the appearance is dominated by the scattering of light inside a leaf. Real-time graphics only tries to model the appearance of objects so fully accurate models that predict the light transport are not required and using measured data to reproduce the appearance without an exact knowledge of the internals is sufficient to display highly realistic results.

Scattering of light is a wide field in computer graphics, ranging from scattering in gaseous structures such as clouds or fog to scattering in fluid and solid material such as milk, marble, skin or leaves. In fluid and solid materials which reside inside a non-scattering medium, usually air, the scattering can be described with a BSSRDF [8]. Compared to a BSDF, the incident light can be at a different position than the exitant light, making a BSSRDF an 8-dimensional non-local function. This high dimensionality poses a computational problem which can only be solved exactly by path tracing. Practical methods reduce the dimensionality, compromising on the accuracy of the solution or deriving analytical expressions for special cases.

Concerning real-time rendering, subsurface scattering is an active research area with many results. Examples are skin subsurface scattering [9], scattering in more general lighting conditions [10] or deformable models [11]. Although this field can be seen as a complete sub-area of real-time rendering, only a few publications propose techniques that specifically deal with realistic leaf rendering.

Many properties of a leaf such as local thickness, optical density or internal structure have an essential impact on its appearance. These values are usually not generated synthetically but measured, so data sets have to be created that a model can be fitted or verified to.

A. Measurements

A realistic result can be achieved by measuring, since nature contains many small imperfections which are automatically captured. In this case, the surface of a leaf is fully reproduced so all structures on its surface, including any bending and bulging are incorporated, resulting in a very realistic appearance. The acquisition setup allows generating high-resolution maps (smaller than $1mm$) using a very simple process and off-the-shelf scanning hardware, so even the smallest details, which have an essential impact on both the reflectance and translucence, are captured.

The devices used are a 3D scanner operating at an effective resolution of 0.1 mm (Minolta VI-910), a digital camera (Canon EOS 20D) with fixed exposure time, two 1,000 Watt light sources with large box diffusers, and an easy to construct fixing frame for the leaf. The large diffusers are used to approximate hemispherical illumination, which is required for capturing the albedo, removing any directionality in the illumination.

The leaf is sampled by first taking a 3D scan, then the scanner is replaced with a diffuser and the albedo is recorded using the camera. To capture the translucency, the front diffuser is switched off and a picture is taken with the back diffuser on. The same procedure is done for the back side of the leaf.

For postprocessing, standard tools are applied. Maya was used to create a simplified mesh and to generate highly detailed normal maps [12] and displacement maps for both sides. Further, the thickness map is generated by subtracting the displacement maps, normalized to a user-defined maximum thickness which can also be measured directly on the leaf. The normal maps are not bound to a specific geometry but can be mapped to different geometric levels of detail (see Figure 2). Figure 3 shows a complete data set generated using this measurement method. In comparison to the acquisition setup by Wang et al. [13], the per-pixel BRDF or BTDF data is not captured, requiring a custom-built linear light source device in order to measure both sides. Spatially varying roughness or specular intensity according to measurements cannot be encoded, though hand-produced modulations of BRDF and BSDF parameters are still possible. On the other

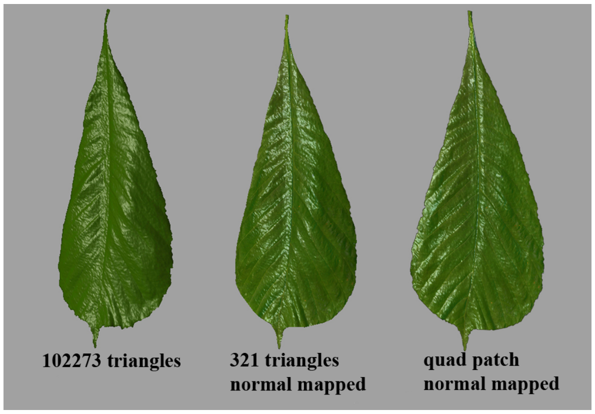

Fig. 2. The scanned geometry, normal-mapped simplified geometry and the normal map on a quad patch. The highlights have been exaggerated for visualization purposes.

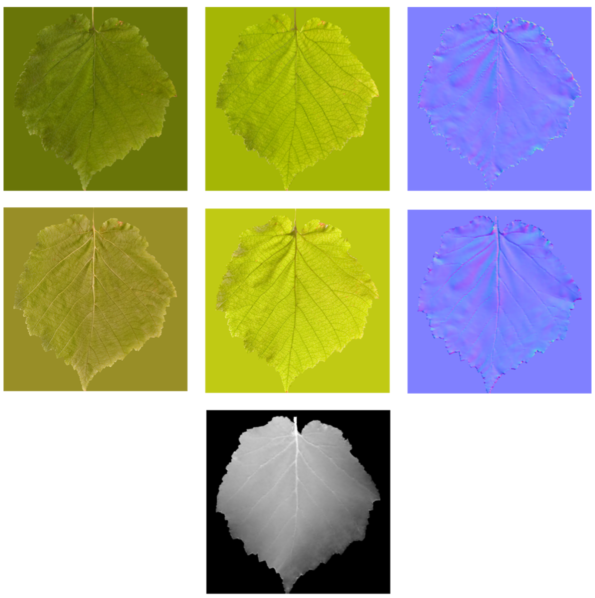

Fig. 3. A complete data set of a leaf, consisting of albedo (left), translucency (middle) and normal map (right) for both sides and a thickness map (bottom).

hand, high-resolution normal maps are created, which causes highlights to be placed more accurately according to the high-frequency structure of the leaf. This is not the case for the method proposed by Wang et al. [13].

B. Reflectance

The structure of a leaf is mostly perceived in its specular reflectance properties due to direct sunlight illumination, revealing its high-frequency structures, which are the most prominent features in the illumination and need to be modeled correctly to achieve a realistic leaf rendering. There is a huge variety of leaf BRDFs, ranging from velvet-like due to micro-hairs to highly specular caused by a thick waxy layer. In most cases, the front of a leaf has broad specularity whereas the back of a leaf is diffuse.

Following Bousquet et al. [14] and Wang et al. [13], the Cook-Torrance shading model [15] is used for the front side of the leaf. A simple diffuse model is applied to the back side of the shown leaves. This is not a general limitation, the reflectance attributes of each measured leaf should be examined to match them for a faithful reproduction. Other BRDFs such as Blinn [16] or Schlick [17] are also good choices depending on the physical accuracy required or other factors such as editability or parameter tuning.

As for the parameters of the Cook-Torrance BRDF, the measured and fitted specular coefficients of Bousquet et al. [14] are used. Their measurements define a range of $n = 1.2 - 1.7$ for the effective refractive index and $\sigma = 0.078 - 0.5$ for the roughness, covering highly specular leaves (e.g. *Laurel*) to nearly diffuse specular lobes (e.g. *Hazel*). Figure 4 shows leaves with different parameter configuration.

Fig. 4. Quad patches shaded with highly specular (top) and almost diffuse (bottom) reflectance, and with a directional light at steep (left) and grazing angle (right).

C. Translucency

One of the main insights of the shown technique is that while subsurface scattering has only negligible impact on the appearance of the light-facing side of a leaf, it is the dominant factor for the opposite side. Figure 5 demonstrates the difference between a simple, yet state-of-the-art translucency model based on a diffuse BTDF, and a BSSRDF approach. As opposed to the reflective part of the leaf, where high-frequency features are conveniently modeled using a normal map, the same approach should be taken to model the high-frequency surface variations in the translucent part compared to a simple diffuse shading model using the geometric normal of the leaf. By including these variations, depending on the incident light angle, the leaf appears either smooth at steep angles or shows the influence of the high-frequency details from bulges and veins at grazing light angles (see Figure 5).

The main features taken into account by the BSSRDF model are self shadowing of the leaf before the light pen-

Fig. 5. Physically based leaf translucency (top) with light at different angles from steep (left) to grazing angles (right) in comparison to the standard diffuse translucency model (bottom).

etrates into the leaf interior, variations in leaf thickness, and variations of the reflectance properties over the light-facing leaf surface. These effects lead to variations in the amount of light entering the medium and scattering towards a specific point to be shaded on the opposite leaf surface. Note that the presented model is local to a leaf, and therefore light variations due to shadows from other leaves or similar only modify the resulting radiance, but do not enter into the subsurface scattering computations. These effects can be handled using standard real-time shadow algorithms.

A combination of both reflectance and translucency, showing the consistency of both can bee seen in Figure 6.

Fig. 6. Leaf on a tree showing fully consistent reflectance and translucency at grazing light directions.

IV. Tree Animation

Generally, animating a tree involves a number of components. First, a wind model describes the characteristics of the wind-tree interaction which is coupled with a dynamic system of some form that describes the reaction of branches and leaves to the applied wind force. Usually, the dynamic system incorporates a structural model to define the hierarchical organization. Structural elements then define how the results of the dynamic system affect the geometry of the tree, the bending of branches for example.

The prevalent structural element model for trees is a skinned or rigid skeletal joint system analogous to rigid or smooth skinning of characters([18], [19], [20], [21], [22], [23]). Many interactive methods simply avoid the problem of bending by not considering any form of deformation ([20], [21]). To achieve deformation, a skeletal structure is used to segment single branches in order to model the smooth bending of a branch. With segmented branches, a principal problem occurs since a high number of joints are needed to get convincing results. Additionally, if leaves are represented separately, each leaf requires its own joint.

Another approach to model the deformation of branches is to use a structural mechanics model ([24], [25], [26]). Structural mechanics is the computation of deformations, deflections, and internal forces or stresses within structures, either for design or for performance evaluation of existing structures. As this is the basis of many engineering sciences, a number of different and highly developed methods exist, though the simpler methods are sufficient to accurately model a branch, as the goal is to achieve a correct appearance and thus strong simplifications can be introduced without compromising the realistic appearance of a deformation.

A. *Wind Model*

To avoid a full fluid dynamics simulation, a stochastic approach from wind engineering can be applied as proposed by Shinya et al. [25] which is also used by Zhang et al. [19]. Instead of solving the Navier-Stokes equation, wind is modeled as a velocity field with longitudinal, lateral and vertical components. Each fluctuating component of the resulting velocity vectors is modeled by a stationary Gaussian stochastic process. The spatial-temporal properties of the components in the frequency domain are represented by the Cross-Power Spectral Density Matrix to model the coherence of the fluctuations where the FFT (Fast Fourier Transform) delivers the velocity field. Since the fluid simulation is replaced by an FFT, this approach is much faster and still can deliver realistic and physically based wind fields, because turbulent wind can be modeled accurately through a stochastic process. Stam [24] applies a similar model by filtering uncorrelated random velocity vectors for each branch in the frequency domain to achieve a correlation of loads on nearby branches.

B. *Heuristic Animation*

Heuristic models do not try to solve the animation problem by means of a simulation, but rather try to emulate the appearance of vegetation movement as efficiently as possible ([21], [27]) using noise functions to drive the animation.

This usually does not require dedicated structural elements, structural models or elaborate calculations but still can deliver acceptable animation. The simplest approach is to modulate the position of a vertex by a noise function, disregarding any correlation to the geometric structure. Additional weights, which define the strength and direction of the displacement, can introduce user-defined constraints and a high controllability, which makes this approach very artist friendly and can also emulate the bending of branches on a coarse level [27].

The key element for real-time performance is to *localize* all computations in a vertex shader, leading to so-called vertex displacement. This is often used to animate simple vegetation represented as billboards (e.g., grass) or billboard clouds (e.g. simple tree models). For full geometry models, this is not straight-forward.

The general approach is to expose all relevant information of the tree structure, i.e. the hierarchy of the structural model, to every vertex. Thus, the hierarchical deformations of all parent branches can be explicitly performed inside the vertex shader and no information needs to be propagated at runtime. This can be achieved by assigning each vertex an index into a texture that holds all necessary information. This means that every vertex within a branch has the same index. Additionally, as sub-branches can emanate at any position on its parent branch, the relations of each vertex to all parent hierarchy levels are required. To have this data exposed to each vertex, the normalized local coordinate $x \in [0..1]$ of the vertex is precalculated as a scalar per vertex, where x is along the principal axis of a branch. Also, the x-values at parent-branch connections are calculated and propagated down the branch hierarchy (see Figure 7). These values

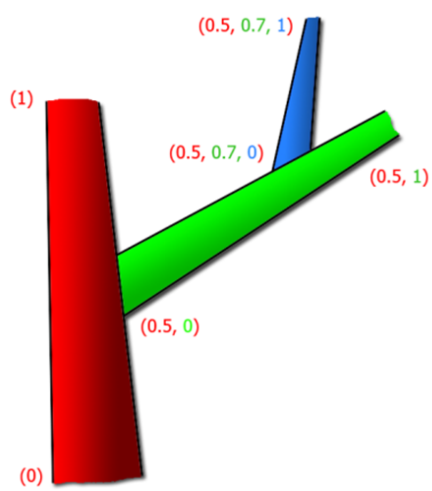

Fig. 7. $\vec{w}$ distribution of branches in a tree.

are stored with each vertex in addition to its own x-value. The shown trees have 4 hierarchy levels, so each vertex has a vector $\vec{w}$ of 4 values associated in addition to the branch index. A problem that occurs with hierarchical vertex displacement is that the used deformation model that defines the structural elements needs to be able to correctly transform the local coordinate axes between hierarchy levels, so tangent and normal transformations need to be available in order to transform local coordinate systems as well.

V. Beam Model

The model used for describing the geometry and physics of a branch as a beam determines how realistic branches swaying in the wind appear to the viewer. A common approximation for realistic animation systems is to model the beam as an elastic cylinder (uniform beam) using structural mechanics, and describe the deformation due to a uniform traversal force using a polynomial deflection function depending on the basic physical properties of the beam. However, uniform beams are not a good approximation for tree branches, as branches are not uniform beams but thin out (taper) at their free end, which has an essential impact on the bending behavior. This taper leads to the effect that tips are much more flexible than thicker parts, which is not accounted for in a uniform beam model. Also, the length needs to be taken care of to achieve a convincing deformation.

VI. Synthesizing Branch Motion

A tree interacting with wind is a highly complex dynamic system that is difficult to solve through numerical simulation while upholding the restrictions of hierarchical vertex displacement, not allowing access to previous states. To simplify the dynamic system, it is treated as an uncoupled system of harmonic oscillators per branch. The basis of this simplification is that an observer cannot judge the correctness of the response function of the highly complex dynamical system because the wind itself is not visible. Thus, the characteristics of the animation are mainly determined by the frequencies and amplitudes of branches and not by their exact functional values.

To avoid explicit integration of the equations of motion, *spectral methods* similar to [26] are used to synthesize branch motion to drive the branch deformation. The principal idea is to generate noise functions that obey the same frequency distributions as empirically observed data or results of a full simulation.

VII. Conclusion

The presented method allows efficiently animating and rendering highly detailed trees with a massive amount of branches and leaves in high quality. With the stochastic approach, there are no considerable costs and enough resources remain free for other calculations such as shading the tree or other parts of a natural scene.

The shown set of methods are confined to a vertex shader using hierarchical vertex displacement, leveraging the performance of GPUs and also making it easy to integrate into existing frameworks.

In summary, the presented methods provide a simple and efficient way for high quality animation. There are no elaborate precomputations required and all parameters can be changed interactively, both on a high level such as the damping coefficient of a level, and on a very low level such as the physical properties of an individual branch.

VIII. Acknowledgements

This research was funded by the Austrian Science Fund (FWF) under contract no. P21130-N13.

Fig. 8. Detailed animated tree.

REFERENCES

[1] P. E. Oppenheimer, "Real time design and animation of fractal plants and trees," in *SIGGRAPH '86: Proceedings of the 13th annual conference on Computer graphics and interactive techniques*. New York, NY, USA: ACM, 1986, pp. 55–64.

[2] P. Prusinkiewicz, A. Lindenmayer, and J. Hanan, "Development models of herbaceous plants for computer imagery purposes," *SIGGRAPH Comput. Graph.*, vol. 22, no. 4, pp. 141–150, 1988.

[3] P. de Reffye, C. Edelin, J. Françon, M. Jaeger, and C. Puech, "Plant models faithful to botanical structure and development," in *SIGGRAPH '88: Proceedings of the 15th annual conference on Computer graphics and interactive techniques*. New York, NY, USA: ACM, 1988, pp. 151–158.

[4] O. Deussen, P. Hanrahan, B. Lintermann, R. Měch, M. Pharr, and P. Prusinkiewicz, "Realistic modeling and rendering of plant ecosystems," in *SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques*. New York, NY, USA: ACM, 1998, pp. 275–286.

[5] W. V. Haevre, F. D. Fiore, P. Bekaert, and F. V. Reeth, "A ray density estimation approach to take into account environment illumination in plant growth simulation," in *SCCG '04: Proceedings of the 20th spring conference on Computer graphics*. New York, NY, USA: ACM, 2004, pp. 121–131.

[6] A. Kharlamov, "Next-generation speedtree rendering," in *GPU Gems 3*, H. Nguyen, Ed. Addison Wesley, July 2007, ch. 4.

[7] S. L. U. Stephane Jacquemoud, "Leaf optical properties: A state of the art." in *8th Int. Symp. Physical Measurements & Signatures in Remote Sensing*, 2001, pp. 223–232.

[8] F. E. Nicodemus, J. C. Richmond, J. J. Hsia, I. W. Ginsberg, and T. Limperis, *Geometrical considerations and nomenclature for reflectance*. USA: Jones and Bartlett Publishers, Inc., 1977.

[9] D. L. Eugene d'Eon, "Advanced techniques for realistic real-time skin rendering," in *GPU Gems 3*, H. Nguyen, Ed. Addison Wesley, July 2007, ch. 14.

[10] R. Wang, J. Tran, and D. Luebke, "All-frequency interactive relighting of translucent objects with single and multiple scattering," in *SIGGRAPH '05: ACM SIGGRAPH 2005 Papers*. New York, NY, USA: ACM Press, 2005, pp. 1202–1207.

[11] T. Mertens, J. Kautz, P. Bekaert, F. V. Reeth, and H.-P. Seidel, "Efficient rendering of local subsurface scattering," in *PG '03: Proceedings of the 11th Pacific Conference on Computer Graphics and Applications*. Washington, DC, USA: IEEE Computer Society, 2003, p. 51.

[12] J. Cohen, M. Olano, and D. Manocha, "Appearance-preserving simplification," in *SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques*. New York, NY, USA: ACM Press, 1998, pp. 115–122.

[13] L. Wang, W. Wang, J. Dorsey, X. Yang, B. Guo, and H.-Y. Shum, "Real-time rendering of plant leaves," in *SIGGRAPH '05: ACM SIGGRAPH 2005 Papers*. New York, NY, USA: ACM Press, 2005, pp. 712–719.

[14] L. Bousquet, S. Lacherade, S. Jacquemoud, and I. Moya, "Leaf BRDF measurements and model for specular and diffuse components differentiation," *Remote Sensing of Environment*, vol. 98, pp. 201–211, 2005.

[15] R. L. Cook and K. E. Torrance, "A reflectance model for computer graphics," *ACM Trans. Graph.*, vol. 1, no. 1, pp. 7–24, 1982.

[16] J. F. Blinn, "Models of light reflection for computer synthesized pictures," *SIGGRAPH Comput. Graph.*, vol. 11, no. 2, pp. 192–198, 1977.

[17] C. Schlick, "An inexpensive brdf model for physically-based rendering," *Computer Graphics Forum*, vol. 13, pp. 233–246, 1994.

[18] Y. Akagi and K. Kitajima, "Computer animation of swaying trees based on physical simulation." *Computers and Graphics*, vol. 30, no. 4, pp. 529–539, 2006.

[19] L. Zhang, C. Song, Q. Tan, W. Chen, and Q. Peng, "Quasi-physical simulation of large-scale dynamic forest scenes," in *Computer Graphics International*, 2006, pp. 735–742.

[20] T. Sakaguchi and J. Ohya, "Modeling and animation of botanical trees for interactive virtual environments," in *VRST '99: Proceedings of the ACM symposium on Virtual reality software and technology*. New York, NY, USA: ACM, 1999, pp. 139–146.

[21] O. Shin, T. Fujimoto, M. Tamura, K. Muraoka, K. Fujita, and N. Chiba, "1/fβ noise-based real-time animation of trees swaying in wind fields," in *Computer Graphics International*, 2003, pp. 52–59.

[22] R. Zioma, "Gpu-generated procedural wind animations for trees," in *GPU Gems 3*, H. Nguyen, Ed. Addison Wesley, July 2007, ch. 6.

[23] F. D. F. William Van Haevre and F. V. Reeth, "Physically-based driven tree animations," *Eurographics Workshop on Natural Phenomena*, pp. 75–82, 2006.

[24] J. Stam, "Stochastic dynamics: Simulating the effects of turbulence on flexible structures," *Computer Graphics Forum*, vol. 16, no. 3, pp. C159–C164, 1997. [Online]. Available: citeseer.ist.psu.edu/stam97stochastic.html

[25] M. Shinya and A. Fournier, "Stochastic motion-motion under the influence of wind," *Comput. Graph. Forum*, vol. 11, no. 3, pp. 119–128, 1992.

[26] Y.-Y. Chuang, D. B. Goldman, K. C. Zheng, B. Curless, D. H. Salesin, and R. Szeliski, "Animating pictures with stochastic motion textures," *ACM Trans. Graph.*, vol. 24, no. 3, pp. 853–860, 2005.

[27] T. Sousa, "Vegetation procedural animation and shading in crysis," in *GPU Gems 3*, H. Nguyen, Ed. Addison Wesley, July 2007, ch. 16.