Journal of Computational Information Systems 11: 5 (2015) 1837–1845
Available at http://www.Jofcis.com

Modeling Rippled Fractal Patterns of Leaves ⋆

Nan GENG, Shaojun HU, Zhiyi ZHANG*

College of Information Engineering, Northwest A&F University, Yangling 712100, China

Abstract

Rippled patterns of leaf edges are important factors to portray the natural beauty of plants. However, it is a challenging problem to model this phenomenon because of their irregularity. We propose an efficient hybrid method for generating rippled effects of leaf edges. First, we extract contours and veins from scanned images of leaves and triangulate leaf membrane using a refined Delaunay triangulation method. Next, we deform flat leaves based on a fast and stable cantilever beam model. Finally, we apply fractional Brownian motion (fBm) noise, which generated by inverse fast Fourier transform, to simulating wavy leaves based on their fractal features. We take a maize leaf as an example to demonstrate the effectiveness of our method. The results show that we can create a realistic wavy leaf only by controlling few parameters and provide a new way to model detail features of plant organs in virtual plants.

Keywords: Fractals; Rippled Patterns; Virtual Plants; Wavy Leaves

1 Introduction

For computer representation of plants, leaves are essential parts to portray the natural beauty of plants. Previously, many studies focused on simulating entire plants instead of individual plant organs [1, 2]. Recently, researchers begun to study the modeling of leaves and some modeling techniques are well developed especially for simulating venation patterns [3]. However, natural leaves show diversities such as rippled edges and curved surface even in the same species. Using interactive software, we can model a particular leaf with complicated shapes. The problem is the manually modeling thousands of leaves with a variety of shapes which can cover different wavy and curved situations. In this paper, we present a hybrid leaf modeling method that consists of a stochastic model and an improved cantilever beam model. The advantage of this method is the capability of modeling realistic leaves with wavy edges just by controlling few parameters.

⋆Project supported by the National High Technology Research and Development (**863**) Program of China (No. 2013AA10230402), the National Nature Science Foundation of China (No. 61303124), Fundamental Research Funds (QN2011135) from Northwest A&F University and the National Science and Technology Support Program of China (No. 2012BAH29B04).

*Corresponding author.
Email address: 815802490@qq.com (Zhiyi ZHANG).

DOI: 10.12733/jcis13578
March 1, 2015

2 Related Works

2.1 State of the art of leaf modeling

Representation of botanical trees in Computer Graphics has been studied by many researchers [4, 6, 6]. However, these studies mainly focused on modeling tree structures instead of tree organs such as bark, leaves, flowers and fruits. As the development of Computer Graphics, more and more studies begun to pursue representing detail information of trees. With regard to leaf modeling, some techniques including geometric based methods [5, 7], sketch based method [8], rule based method [3], and image based methods [9, 10, 11] have been proposed. Prusinkiewicz et al. [5] proposed a bi-cubic patches based geometric model to represent complex leaves surfaces. Later, they developed a leaf modeling method [7] by sweeping an open curve along an axis. Ijiri et al. [8] presented a freehand sketch system which can model leaves just by few strokes. This method is very intuitive and interactive for non-expert users. Runions et al. [3] focused on venation patterns of leaves and simulated development of veins by taking hormone distribution into account during leaf growth. Recently, some image based methods have been proposed to model leaves. Mundermann et al. [9] presented a leaf modeling method from a series of processes including extracting a leaf skeleton and triangulating membranes from scanned leaf image. Hong et al. [10] extended Mundermann et al. fs method by introducing generalized cylinder veins and a refined method for membrane triangulation. Quan et al. [11] extracted a generic deformable leaf model from a sample leaf, and used it to fit all the other leaf point clouds reconstructed from sparse images.

To represent rippled edges of leaves, we may use the method of [11] to obtain the point clouds of a real leaf; however, this method can only model a specific leaf by using a sophisticated reconstruction method. Likewise, it is possible to use methods of [9] and [10] to create the rippled edges of a leaf, while these methods need too much user interaction. Therefore, we propose an efficient method to model wavy leaves with few parameters. The closest models to our method are [9] and [10]. However, the most distinguish part of our method is the post-processing steps where we treat leaf veins as cantilever beams and control their deformation by setting external loads with turbulence.

Fig. 1: Photograph of a maize leaf

2.2 Fractal patterns of leaf edges

In a real world, we often observe rippled edges from leaves, flowers and lichens. Fig. 1 shows the photograph of a maize leaf with the rippled edges. Researchers have found that the wavy shapes of leaves and flowers are fractals where a pattern repeats on different scales [12, 13]. The

mechanism of the development of the rippled patterns is the more growth at the edges than the center of leaves. To represent this kind of ruffled phenomena, using fBm noise function is a natural choice because fBm function has the same feature of the fractals-self-similarity.

3 fBm Noise Function

To represent natural phenomena efficiently, Fournier [14] introduced fractional Brownian motion (fBm) noise to model and animate many natural phenomena. The fBm noise denotes a family of Gaussian random functions which can provide a host of time series. Brownian motion was observed in the movement of particles in an oil emulsion, and this motion showed a fractal character which exhibits self-similarity when rescaled using different anisotropic transformations. The major advantage of the fBm noise is the efficiency of animating the waving phenomena without solving time-consuming dynamic equations. Because of the low cost and self-similar properties, we can make use of fBm noise function to model the wavy leaves.

3.1 Generation of fBm noise function

Many approaches, including Poisson faulting, Fourier filtering, midpoint displacement, successive random additions and summing band-limited noises, have been developed to generate fBm noise function according to [15]. We choose Fourier filtering method in [16] since signal in frequency domain is more intuitive and easier to be analyzed than that in time domain. Thus, we first consider how to create the power spectrum of fBm noise function.

According to [15], a continuous spectrum S_f of one-dimensional fBm noise function can be denoted by:

$$S_f = C\frac{Nrand}{f^{\beta/2}} \tag{1}$$

where $Nrand$ is the normal distributed random number, C is a constant, f is the frequency ranging from $0 \to +\infty$ and β is the scaling factor. Fig. 2 shows different spectrums of fBm noise function with various β values.

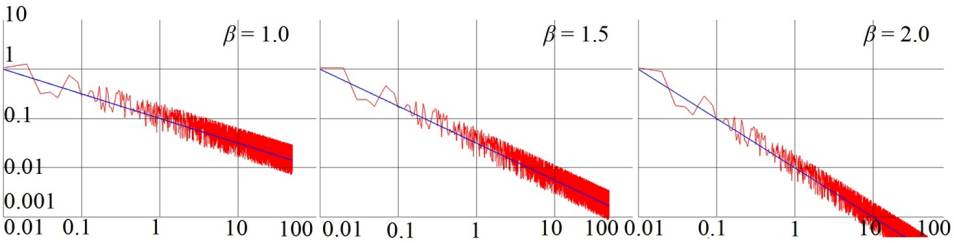


Fig. 2: One-dimensional spectrums with various β

The continuous fBm noise function $N(t)$ can be represented by the following inverse Fourier transform Eq. (2):

$$\begin{aligned} N(t) &= \int_0^{+\infty} S_f e^{i(2\pi ft + rand)} df \\ &= \int_0^{+\infty} S_f cos(2\pi ft + rand) df + i\int_0^{+\infty} S_f sin(2\pi ft + rand) df \end{aligned} \tag{2}$$

Where *rand* is a random phase. To solve $N(t)$, we discretize the frequency f from 0 to $N-1$ with an integer interval 1, then generate discrete spectrum S_f and use IFFT (Inverse Fast Fourier Transform) to obtain a series of noise number $N(t)$ from $t=0$ to $N-1$.

3.2 Practical use

The practical use of fBm noise function is flexible. Firstly, we make use of the self-similarity features by changing β value. Small β value corresponds to sharp change of noise, and large β value generates smooth fBm noise. If we use the same initial random noises, the generated fBm noises will have the similar macro-shape even when β values are significantly different (see Fig. 3).

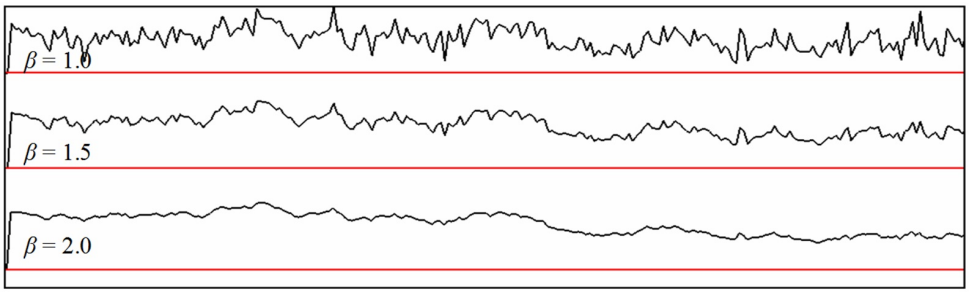


Fig. 3: Noise sequences correspond to various β with spectrum in Fig. 2

The second flexible use of fBm noise is the extension of $N(t)$ to $N'(t)$ by linear transformation:

$$N'(t) = N(rt + c) \tag{3}$$

where r is the sampling interval and c is the phase. Using the equation, we can generate different noise functions only from a single noise $N(t)$ with a specific β value.

4 Improved Cantilever Beam Model

In biomechanics filed, researcher found that branches, petioles and some leaf laminae also function like cantilever beams. To model a curved beam bending in three-dimensions subjected to an arbitrary load, we use a method proposed by Hu et al. [17].

We assume that there is an inextensible curved beam (such as a petiole or a branch) in three dimensions that consists of n discrete segments with length $ds_i (i = 1, 2, ..., n)$ loaded by an external force $\mathbf{F}_i$; $\mathbf{P}_i^0$ and $\mathbf{P}_i$ represent the initial position and the rest join position of segment i respectively. The initial angle between the direction of segment i and the load F_i is θ_i^0 and the equilibrium angle is θ_i; φ_i is the bending angle of segment i with its joint $\mathbf{P}_i$; $\mathbf{N}_i$ is the normal of the plane determined by $\mathbf{F}_i$ and $\mathbf{P}_i^0$; EI_i is the flexural rigidity; and M_i is the resultant bending moment at segment i.

To find the equilibrium angle θ_i and replace the joint position $\mathbf{P}_i^0$ with $\mathbf{P}_i$, we can use the Bernouilli-Euler hypothesis to obtain the following equilibrium function:

$$ds_i^2 sin(\theta_i^0 - \varphi_i)|\sum_{j=i}^{n} \mathbf{F}_j| = EI_i\varphi_i \tag{4}$$

The bending angle φ_i can be solved by Newtonfs method. Finally, we rotate segment i about the normal N_i with φ_i and obtain the equilibrium position $\mathbf{P}_i$.

5 Rippled Fractal Patterns of Leaf Edges

5.1 Skeleton extraction and triangulation

To model wavy leaves, we first need to generate a triangulated leaf model with skeletons. We use image processing method to detect edges from a scanned leaf as shown in Fig. 4(a), and extract the primary vein and the secondary veins of the leaf by using an interactive method as shown in Fig. 4(b). Before triangulation, we generate sample vertices inside the leaf membrane using a user defined square template as shown in Fig. 4(c).

The general triangulation method is the Delaunay triangulation. However, we cannot use this method to triangulate leaf surfaces since it fails to handle concavities and holes as seen in Fig. 4(c). Ruppert [18] proposed a Refined Delaunay Triangulation (RDT) that can handle holes and concavities of planar objects. Fig. 4(d) shows the triangulation result of Fig. 4(c) using the RDT method.

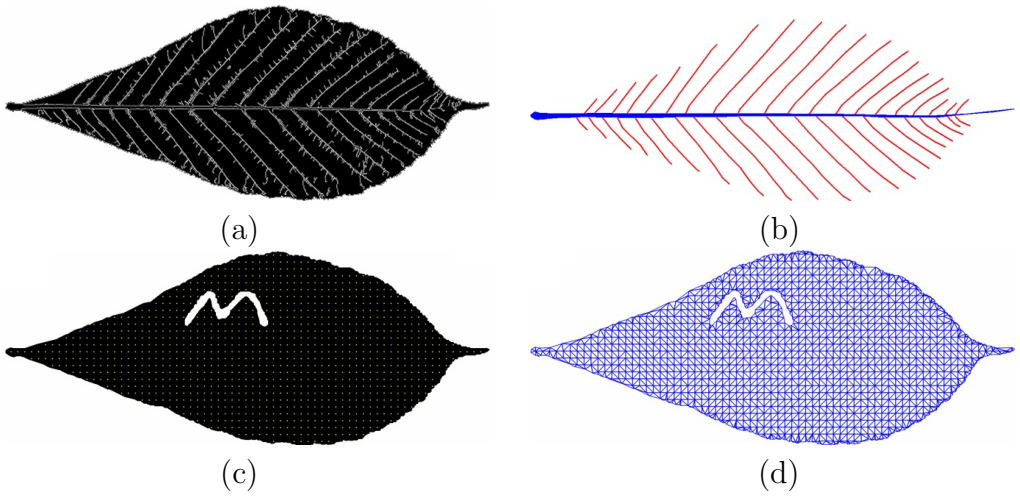


Fig. 4: Skeleton extraction and triangulation of a leaf

5.2 Skeleton extraction and triangulation of a leaf

5.2.1 Leaf model

Our leaf model includes a petiole jointing at a twig, and a lamina jointing at the petiole. Fig. 5 shows the position relationship between twig and petiole, petiole and leaf. We assume that $OT(M_1, M_2, \ldots, M_i, \ldots, M_n)$ represents the primary vein part, and L_iM_i and R_iM_i are the secondary veins. If we treat both the petiole and veins as a curved beam, then we can deform the whole leaf surface. After using the improved cantilever beam model introduced in Section 4, we can flexibly generate non-planar surfaces of leaves just by controlling the petiole, the primary vein and the secondary veins.

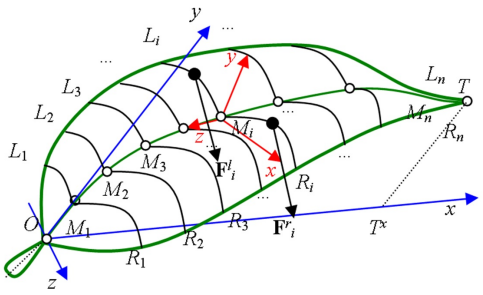
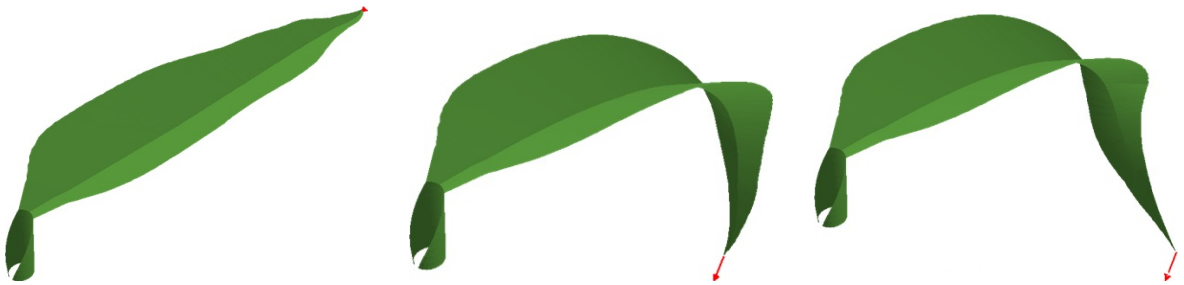


Fig. 5: A leaf model consists of a petiole, a primary vein and secondary veins

5.2.2 Modeling nonplanar leaves

To be specific, the petiole part is deformed when subjected to a load $\mathbf{F}_p$, with a 4×4 transformation matrix $\mathbf{M}_p$. Then, we update the primary vein coordinates by $M_i = \mathbf{M}_p \times M_i$. Next, we bend the vein $OT(M_1, M_2, \ldots, M_i, \ldots, M_n)$ by a load $\mathbf{F}_m$, and obtain new main vein coordinates $M_i = M_a \times M_i$ where M_a is the transformation matrix. In some cases, we observe obvious rotation phenomena of leaves (see Fig. 1). However, this kind of rotation cannot be generated by the beam bending method. Thus, we apply a rotation matrix M_r to each joint position. The rotation matrix is generated by rotating point M_i about axis M_iM_{i+1} with a user-defined angle. As a result, $M_i = M_r \times M_i$. Here, the transformations of these three steps will generate a concatenation matrix $M_s = M_r \times M_a \times M_p$. By concatenating a sequence of matrices into a single one, we can simplify the transformation process and gain efficiency. Then, the secondary vein coordinates P_j on L_iM_i and R_iM_i can be updated by $P_j = M_s \times P_j$. Finally, let the external forces on L_iM_i and R_iM_i be F_i^l and F_i^r, respectively. After using the beam bending method for each secondary vein, we will generate the final position on the leaf skeletons $P_j = M_f \times P_j$ where M_f is the transformation matrix.

Because the rest curved shape is determined by the external loads on the petiole (F_p) and the veins (F_m, F_i^l *and* F_i^r), it is easy to generate various curved leaves only by controlling few parameters. Fig. 6(a) shows the basic maize leaf model in our simulation. Besides the pure bending controlled by beam bending method, we also apply a user defined rotation angle to the primary vein to increase the flexibility of the deformation. Fig. 6(b) and Fig. 6(c) show the deformed maize leaf after bending the primary vein and rotating the primary vein by a user defined angle, respectively.



(a) (b) (c)

Fig. 6: (a) Before and (b) after bending and (c) rotating the primary vein of a leaf

5.2.3 Modeling rippled edges

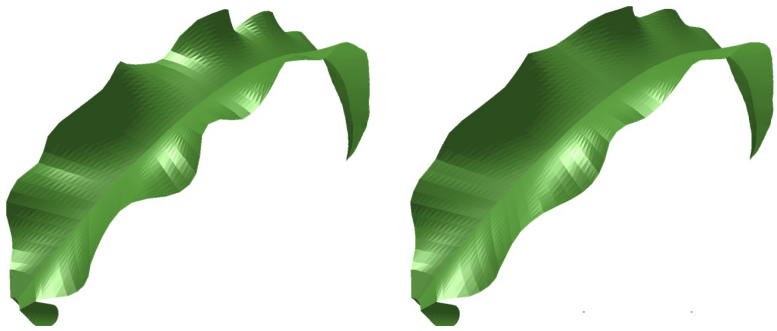
Based on the fBm noise model presented in the Section 3, we can visualize the fBm noise to generate the rippled edges of leaves by carrying out secondary veins deformation. Here, the external loads F_i^l and F_i^r on the secondary veins L_iM_i and R_iM_i are converted to turbulent loads as follows:

$$\begin{aligned} |\mathbf{F}_i^l| &= P_i^l[\lambda + (1-\lambda)N(k^l i + \theta^l)] \\ |\mathbf{F}_i^r| &= P_i^r[\lambda + (1-\lambda)N(k^r i + \theta^r)] \end{aligned} \tag{5}$$

Where P_i^l and P_i^r are amplitudes of loads; $N(k^l i+\theta^l)$ and $N(k^r i+\theta^r)$ are fBm noise functions; λ represents the fluctuation degree of noise; k^l and k^r are used to control the sampling intervals of noises; θ^l and θ^r are phases to control the offsets of noise sequences.

6 Experimental Results

We took a maize leaf as an example and generated various rippled effects. Compared to Fig. 6, we change the deformation from the primary vein to secondary veins using Eq. (5). The control parameters for the secondary veins consist of the strength of load, sampling intervals of fBm noises and the offsets of the noises for the left and the right veins.

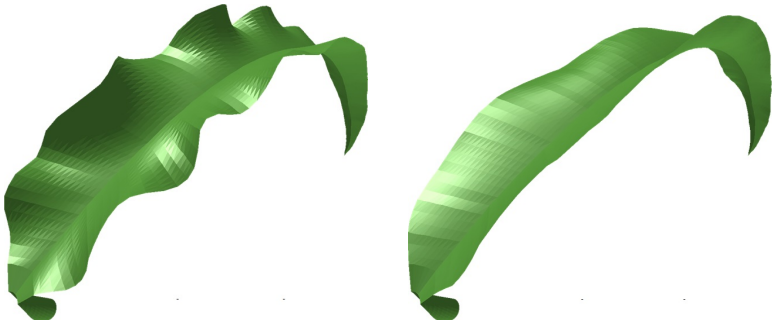


(a) $\beta = 1.0$ (b) $\beta = 2.0$

Fig. 7: Results of wavy maize edges with different β

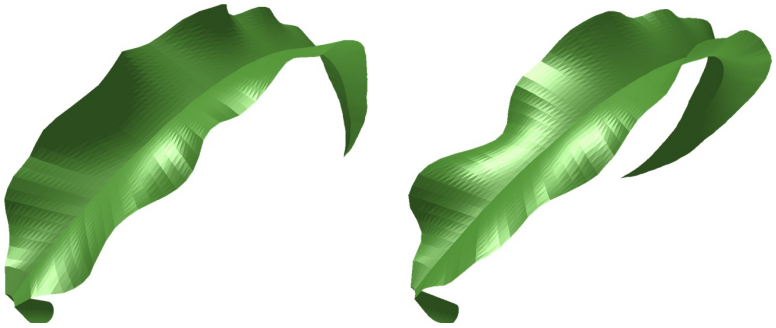
Fig. 6(a) shows the initial positions of the tested maize leaf. Since different β value of fBm noise function will generate different kinds of turbulence. We tested two types of fBm noise functions with $\beta = 1.0$ and $\beta = 2.0$ and they have the same parameters $k^l = k^r = 0.3$ and $\theta^l = 8, \theta^r = 10$, respectively. Fig. 7(a) and (b) show the result of the same maize leaf with different β. We found that the rippled edges in Fig. 7(b) with large β are relatively fairer than that in Fig. 7(a). To test the influence of sampling intervals k^l and k^r, we set different sampling intervals with the same $\beta = 1.0$ and $\theta^l = 8, \theta^r = 10$. Compared to the case with $k^l = k^r = 0.3$ in Fig. 8(a), the simulation wavy edges change more slowly with $k^l = k^r = 0.1$ (see Fig. 8(b)). One of the advantages of this modeling method is the generation of various wavy styles only by changing the phase θ^l and θ_r. Fig. 9(a) and (b) are two visualized maize leaves with the same $\beta = 2.0$, and sampling intervals $k^l = k^r = 0.3$. Yet, by choosing different θ^l and θ^r, the generated leaf shapes are obviously different. After choosing suitable parameters, we can obtain a maize leaf as shown in Fig. 10. In some situation, we need to create lots of leaves with different shapes. Traditional ways may not

solve this problem. Our modeling method, which is controlled only by few parameters (β, k^l and k^r, θ^l and θ^r), is possible to generate thousands of leaves with diversity in a nutshell.



(a) $k^l = k^r = 0.3$ (b) $k^l = k^r = 0.1$

Fig. 8: Results of wavy maize edges with different sampling intervals



(a) $\theta^l = 8, \theta^r = 10$ (b) $\theta^l = 110, \theta^r = 94$

Fig. 9: Results of wavy maize edges with different phases

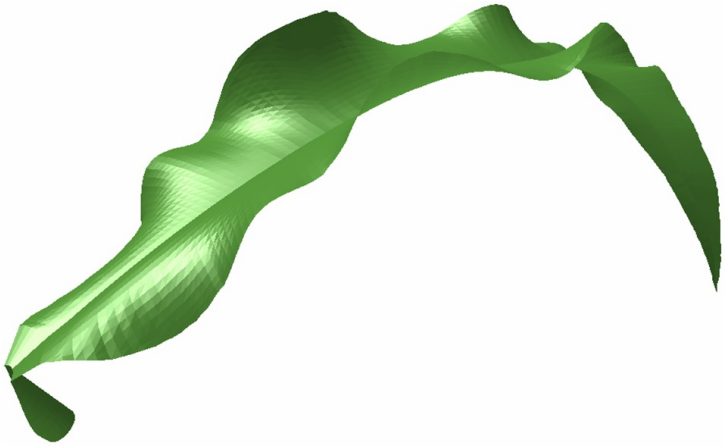


Fig. 10: Visualization of a maize leaf after adjusting several parameters

7 Conclusion

In this paper, we proposed a hybrid method that includes an fBm noise model and an improved cantilever beam model to create wavy leaves. We employed the beam model to interactively control the curved shape of a leaf. Furthermore, we introduced an fBm noise function to simulate the

rippled edges of a leaf. By adjusting the β and the phase of the generated noise, we implemented various fractal patterns along leaf edges. Compared to previous modeling methods, our method is not manually intensive and is easy to model numerous leaves with different shapes in a short time. In future, we intend to extend our method to model flowers since flowers show similar wavy phenomena on the edges of their petals.

References

[1] O. Deussen and B. Linterman, Digital Design of Nature: Computer Generated Plants and Organics, Springer Press, New York, 2005.

[2] P. Prusinkiewicz, M. Hammel and E. Mjolsness, Animation of Plant Development, In Proceedings of ACM SIGGRAPH '93, 1993, pp. 351-360.

[3] A. Runions, M. Fuhrer, B. Lane, P. Federl, A. Rolland-Lagan and P. Prusinkiewicz, Modeling and visualization of leaf venation patterns, ACM Transactions on Graphics 24(3) (2005) 702-711.

[4] P. De Reffye, C. Edelin, J. Francon, M. Jaeger and C. Puech, Plant models faithful to botanical structure and development, ACM SIGGRAPH Computer Graphics 22(4) (1998) 151-158.

[5] P. Prusinkiewicz and A. Lindenmayer, The Algorithmic Beauty of Plants, Springer-Verlag Press, New York, 1990.

[6] O. Deussen, P. Hanrahan, B. Lintermann, M. Mech, M. Pharr, P. Prusinkiewicz, Realistic modeling and rendering of plant ecosystems, In Proceedings of ACM SIGGRAPH '98, 1998, pp. 275-286.

[7] P. Prusinkiewicz, L. Mundermann, R. Karwowski and B Lane, The use of positional information in the modeling of plants, In Proceedings of ACM SIGGRAPH '01, 2001, pp. 289-300.

[8] T. Ijiri, M. Okabe, S. Owada, T. Igarashi, Floral diagrams and inflorescences: Interactive flower modeling using botanical structural constraints, ACM Transactions on Graphics 24(3) (2005) 720-726.

[9] L. Mundermann, P. MacMurchy, J. Pivovarov and P. Prusinkiewicz, Modeling lobed leaves, In Proceedings of Computer Graphics International '03, 2003, pp. 60-65.

[10] S. M. Hong, R. B. Simpson and G. V. G. Baranoski, Interactive venation-based leaf shape modeling, Computer Animation and Virtual Worlds 16(3-4) (2005) 415-427.

[11] L. Quan, P. Tan, G. Zeng, L. Yuan, J. Wang and S. Kang, Image-based plant modeling, ACM Transactions on Graphics 25(3) (2006) 772-778.

[12] E. Sharon, B. Roman, M. Marder, G. Shin and H. L. Swinney, Buckling cascades in free sheets, Nature, 2002, pp. 419-579.

[13] E. Sharon, M. Marder and H. L. Swinney, Leaves, flowers and garbage bags: making waves, American Scientist 92 (2004) 254-261.

[14] A. Fournier, D. Fussell and L. Carpenter, Computer rendering of stochastic models, Communication of the ACM 25 (1982) 371-384.

[15] F. K. Musgrave, C. E. Kolb and R. S. Mace, The synthesis and rendering of eroded fractal terrains, Computer Graphics 23(3) (1989) 41-50.

[16] H. O. Peitgen, D. Saupe, The Science of Fractal Images, Springer-Verlag Press, New York, 1988.

[17] S. Hu, T. Fujimoto and N. Chiba, Pseudo-dynamics model of a cantilever beam for animating flexible leaves and branches in wind field, The Journal of Computer Animation and Virtual Worlds 20(2-3) (2009) 279-287.

[18] J. Ruppert, A Delaunay refinement algorithm for quality 2-dimensional mesh generation, Journal of Algorithms 18(3) (1995) 548-585.