An algorithm for generating vein images for realistic modeling of a leaf

Yodthong Rodkaew[1], Suchada Siripant[2], Chidchanok Lursinsap[2], Prabhas Chongstitvatana[1]

[1]Intelligent System Laboratory (ISL)
Department of Computer Engineering
Faculty of Engineering
Chulalongkorn University
Phayathai road, Bangkok 10330 Thailand
e-mail: 43718130@student.chula.ac.th, prabhas@chula.ac.th

[2]Advanced Virtual and Intelligent Computing (AVIC) Research Center
Department of Mathematics
Faculty of Science
Chulalongkorn University
Phayathai road, Bangkok 10330 Thailand.
e-mail: ssuchada@chula.ac.th, lchidcha@chula.ac.th

ABSTRACT- The shape of a natural leaf is very complicated. A computer can create the shape of a natural leaf using algorithms such as fractal systems or L-system. These systems require a human assistant to analyse and to synthesize equations of the shape. In this work, an algorithm is developed for generating vein images of a given leaf shape. The algorithm creates vein images from the trails of particles that scattered within a shape. The direction of motion of a particle is controlled by an equation $\hat{p}+\hat{q}$ where $\hat{p}$ is a unit vector denoting the direction from a particle to its nearest neighbor and $\hat{q}$ is a unit vector denoting the direction from a particle to a presumed target point. At each time step all particles move according to the equation of motion. When the particles come near each other they will be combined together. The process is repeated until only one particle remains. The result is the veins of that given leaf shape. Various parameters control the appearance of the output image. The figures generated from this algorithm look very natural. The algorithm can cover several leaf shapes.

KEY WORDS – Vein, Leaf, Leaf-modeling, Particle systems, Image synthesis.

1. Introduction

Realism is an important objective in computer graphics. Many natural shapes are complex. Modeling and rendering of natural shapes is a challenging problem. There are many work targeted on plant modeling such as those in [1-16]. The vegetation has a lot of complexity when one trys to model it. It is possible to collect the information from nature and uses that directly for rendering, but this could require a large database. This database is also inflexible and is not responsive to the change in the environment adhence makes the computer generated scene unrealistic.

The modeling synthesis can solve this problem. One of the most popular system is called L-system [1,2]. The L-system was developed by a biologist, Aristid Lindenmayer. Its model is based on a generative grammar that can be used to generate images. Many researches are interested in characterizing and modeling several kinds of plants. Modeling various parts of plants have been actively researched, for example the diameter of a branch, the angle between branches, the length, the form of a bush, the effect from the environment, the quality of soil, sunlight, wind, water, including topiary - pruning by human [5].

There are many works on the synthesis of a leaf [18-21]. It is difficult to characterize and make a good model. It requires a lot of computational power to fit parameters to natural data such as artery [17] or veins [18]. The work in [17] synthesizes the curve of the artery in the retina using L-system and tunes parameters by

Genetic Algorithms (GA) [22] and Evolutionary Computation (EC). In [19] a general form written as a grammar in L-system is used to synthesize several kinds of leaf shapes. The work in [18] uses a set of grammar in L-system to generate "skeleton" of a leaf. GA is used to find parameters that "shape" this skeleton to fit the outline of a target leaf. The work in [21] represents a leaf by complicated folding, similar to origami. In [23] veins are considered as the network and the network is produced probabilistically using the L-system.

L-system is one of the best algorithms for synthesizing the model of natural shapes. It can represent the growth easily by replacing the rule's symbol iteratively. Those symbols can be parts of a plant such as branches, buds, leaves (Figure 1). Writing the L-system rules by hand using human ingenuity produces the best result but it is a difficult task. The adaptation of L-system rules to generate leaves is possible [18]. It starts at the main vein then grows the secondary vein in every iterations (Figure 2). However in a leaf, only the main and the secondary vein can be produced well. The complexity of veins in the higher order prohibited any realistic generation because it is unlikely to find appropriate parameters for such complex shapes (Figure 4).

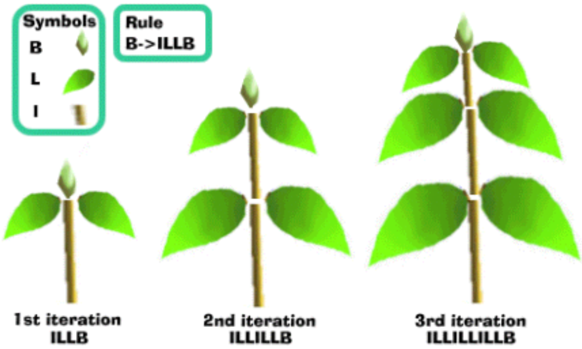


Figure 1. An illustration of plants synthesized by L-system.

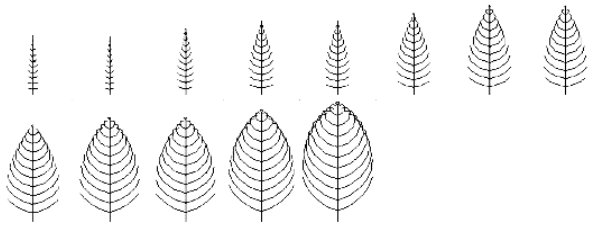


Figure 2. Simulation of the growth of leaf.

This paper proposes an algorithm based on a system of interacting particles to synthesize vein images from a given leaf shape. The paper is organized as follows: Section 2 describes the components of a leaf and the algorithm. Section 3 details the experiment. The discussion of the result is in Section 4. Section 5 describes a method to enhance the image quality and Section 6 concludes this paper.

2. The algorithm

The leaf (Figure 3) consists of several parts: the *vein* is an arborescence that immersed in the *limb* which is the leaf body. The leaf is connected to tree branches by a *petiole* (or *leafstalk*) through the *main vein*. The leaf's border and a *lobe* is part of a leaf.

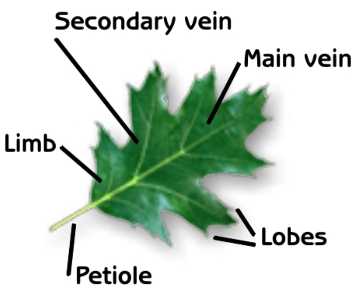


Figure 3. An illustration of a leaf.

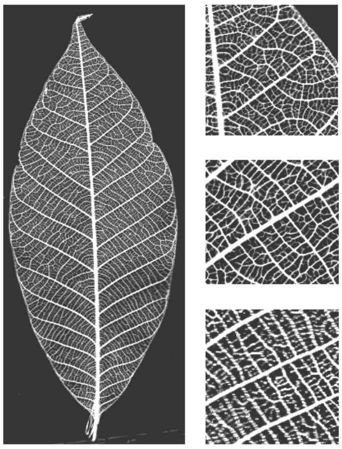


Figure 4. An illustration of veins.

An algorithm has been developed based on assumptions below:

- the vein is an arborescence that immersed in the limb.
- the limb has more area than veins.
- the vein transports substances between the limb and the tree.
- the diameter of a vein depends on its capability to transport a substance.
- the part of limb produces equivalent substance.
- the leaf shape depends on vein, in the opposite direction, a vein depends on the leaf shape.
- the leaf shape is controlled by its genetic.

In a photosynthesis process, we assume that each part of a leaf produces equal energy and sends it via veins. Each area can be represented as a particle. The particle produces some amount of energy and the energy is transported to the petiole (Fig. 5a). The transportation should be efficient. Two algorithms are presented that exhibit behaviours according to these assumptions. Algorithm 1 shows the main idea and Algorithm 2 incorporates the efficiency concern.

Algorithm 1

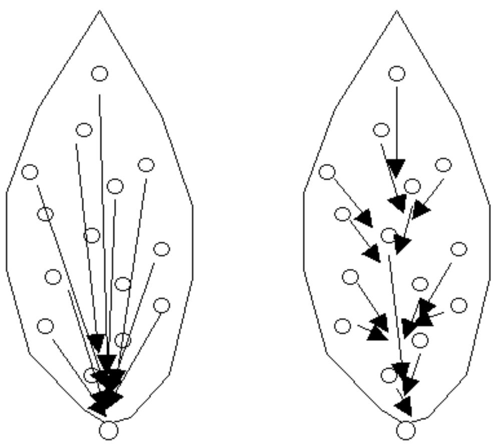
1. set the particles randomly in the boundary of a leaf shape
2. move the particles to the target (the petiole)
3. repeat (2) until all of particles reach the target

Each particle carries a fixed amount of energy and moves directly to the target. Figure 5a represents the path of motion of particles. Some leaf has this pattern. However, this is not an efficient way to transport energy, it is expensive to produce one path per particle. The next algorithm shares the paths.

Algorithm 2

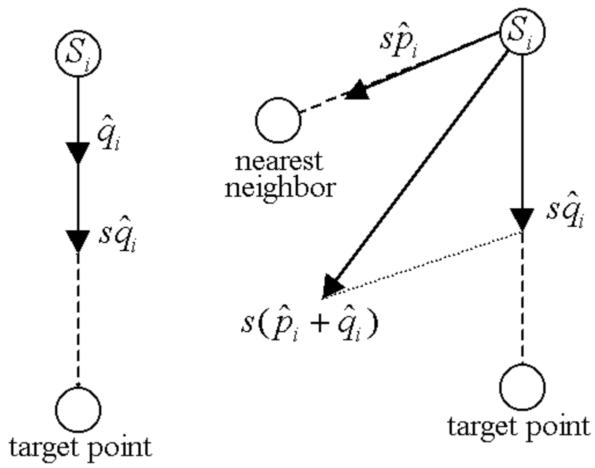
1. set the particles randomly in the boundary of a leaf shape
2. try to use the same path:
   2.1 move towards the nearest particle
   2.2 move to the target together
3. repeat (2) until all of the particles reach the target

The path sharing occurs on this algorithm. When two particles come close, they are merged into the same path. Figure 5b shows this behaviour. Several kinds of leaf have this pattern. The next section describes the algorithm in more details.



(5a) (5b)

Figure 5. The diagram of the movement of particles .



(6a) (6b)

Figure 6. The motion of a particle.

The particles transportation

The algorithm creates veins from the trail of particles that are scattered within a leaf shape. Let *S* be the set of particles. At first, each particle σ_i carries the energy $e_i = 1$. The direction of motions of particles are controlled by an equation : $\hat{p}_i + \hat{q}_i$ where $\hat{p}_i$ is a unit vector denoting the direction from a particle to its nearest neighbor and $\hat{q}_i$ is a unit vector denoting the direction from a particle to a presumed target point. Algorithm 1 is a special case of Algorithm 2. Algorithm 1 ignores $\hat{p}_i$ and the particle moves along $\hat{q}_i$ only.

At each time steps all particles move by a distance *s*, when the particles σ_i and σ_j come closer than r from each other they are combined together and the energy is conserved $e_{new} = e_i + e_j$. While moving, the particle leaves a trail with the width $w_i = f(e_i)$ adhence the thickness of veins are increased when it is closer to the petiole. The process is repeated until only one particle remains.

There are many parameters governing the generation of vein images. They are divided into two groups: particle distribution and their motion. The distribution pattern determines the initial positions of particles and their density. The parameters determining the motion include the neighbourhood radius, the initial energy of each particle and the speed (the step size).

3. Experiment

The leaf shape data can be obtained from the image of a real leaf. An image processing technique can be used to extract the boundary. Alternatively, the shape can be manually created using spline curves. The initial positions of particles are determined by the scatter pattern. Four patterns are used in the experiment (Figure 7):

- square grid – particles are on the corners of regular squares.
- hex – a variant of square grid, in every three rows the pattern is shifted to the right by half unit.
- jitter – perturbing square grid by randomly moving the particle with a square.
- random – random placement of particles

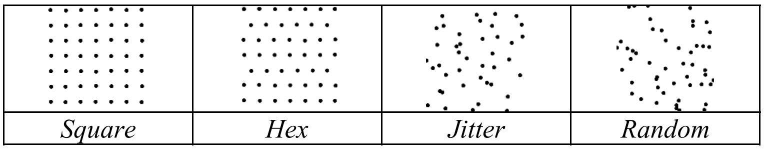


Figure 7. The scatter patterns.

The leaf is scanned and scaled to be approximately 200 × 200 pixels. The color of the trail reflects the amount of energy. The lower energy is presented by a light color. A darker color means higher energy. When particles move near to the petiole in the middle area of a leaf the trails are merged and create a dark color. Figure 8 shows the development of the veins from the motion of particles.

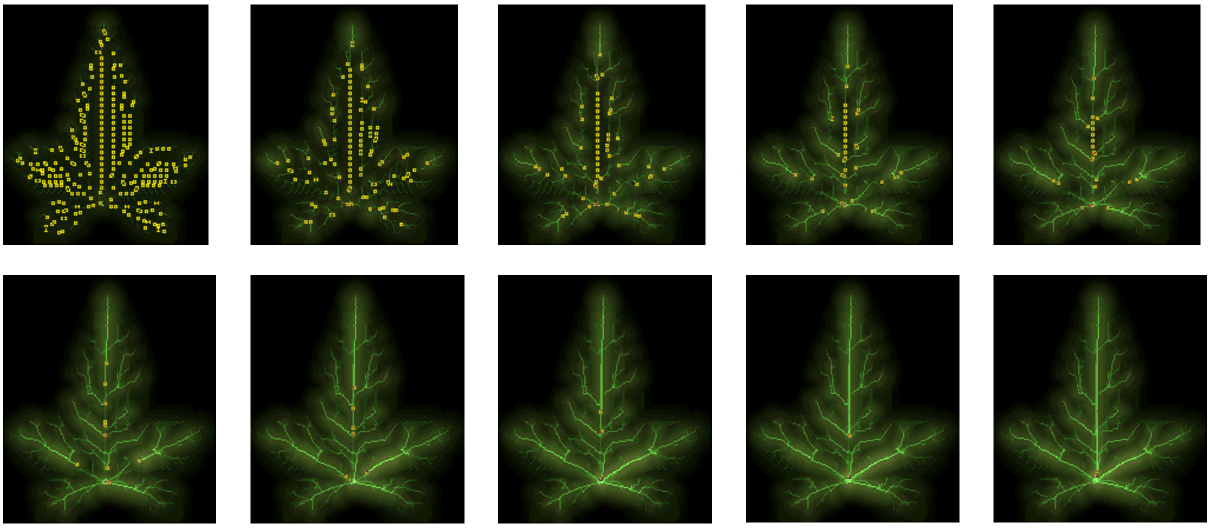


Figure 8. The development of veins.

The generation of vein images with various parameters are carried out to find the effect of these parameters on the quality of the outputs. The leaf shown is the ivy leaf. The density of particles is varied with the size of a grid from 3, 5, 7 and 10 pixels. For the random pattern, the number of particles is used instead of the grid size and they are 125, 254, 500 and 1378 particles.

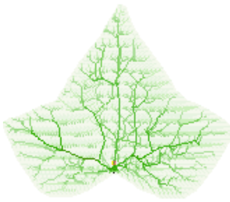
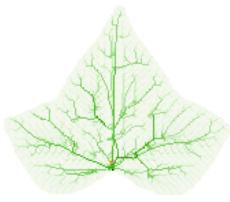
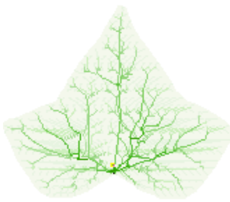
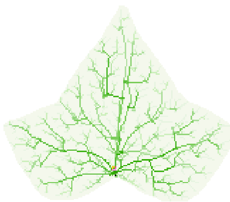
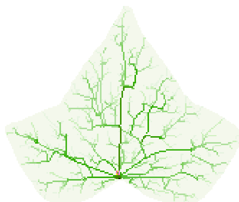
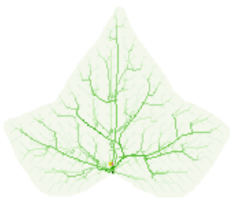
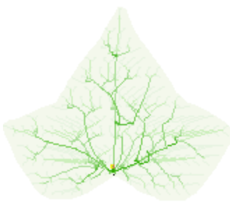
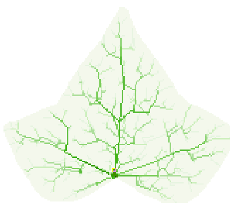
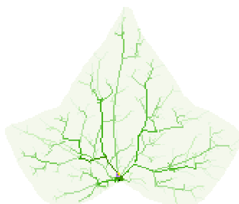
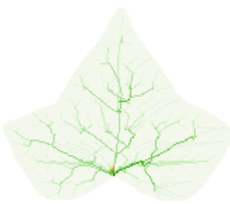
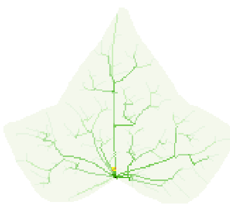
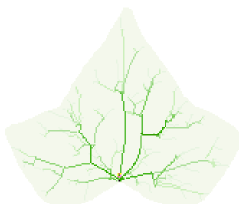
Real ivy	square	hex	jitter	random
space 3 pixels (1378 particles)				
space 5 pixels (500 particles)				
space 7 pixels (254 particles)				
space 10 pixels (125 particles)				

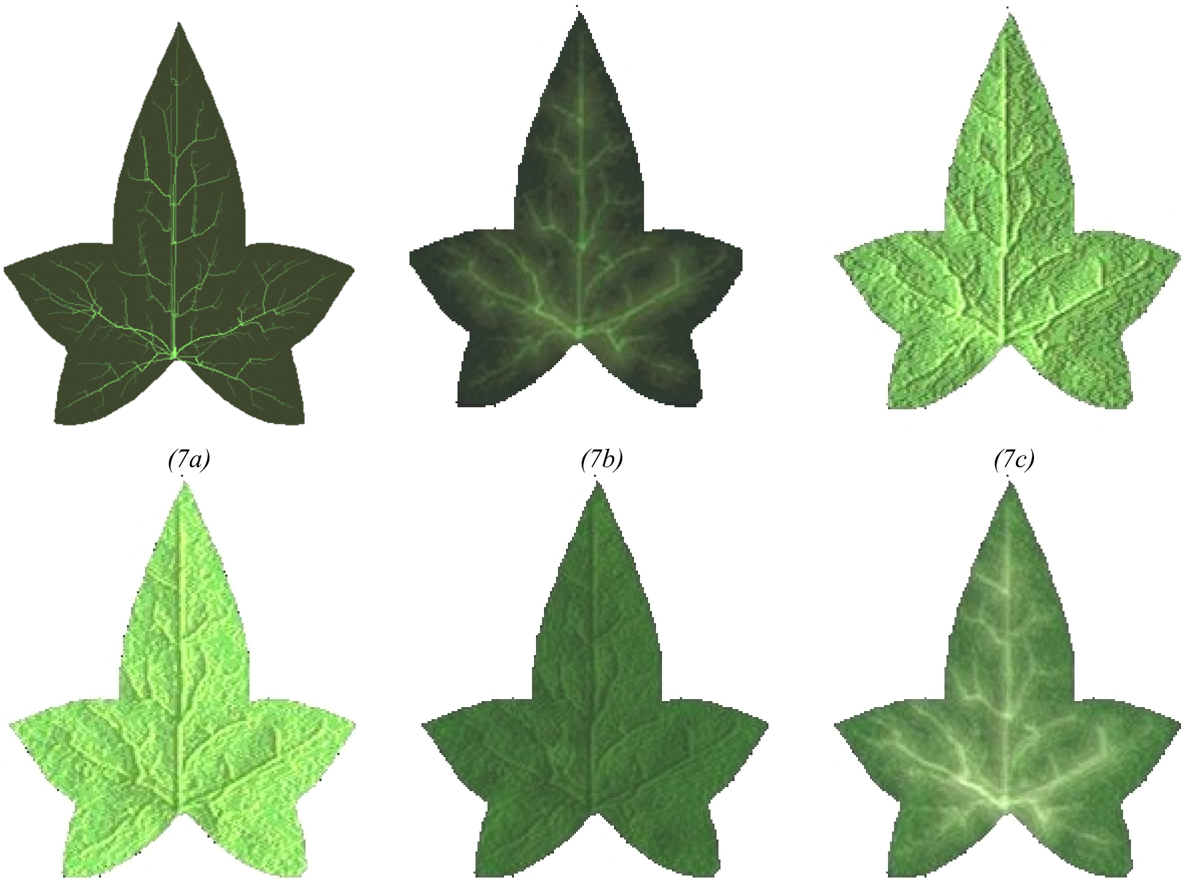
Figure 9. Generation of vein images with various parameters.

4. The Result

Different scatter patterns result in different look of vein images. The vein images from the jitter and random patterns have a good uniform distribution over the whole area. The form of lines from the square pattern is regular and symmetric compared with all other patterns. The image from the hex pattern veins has connected higher order horizontally. Comparing the jitter pattern and the random pattern, the jitter pattern gives rise to more regular lines than the random pattern. When the real leaf is compared with the synthesis image, the best result comes from the square and jitter patterns with the grid size 7 pixels. It is also interesting to note that beside three main veins, in the bottom of the leaf there are two more veins and they look like the real leaf. The result from many forms of leaf indicates that the square grid is better than other patterns. The best density is the grid size around 5 to 7 pixels. Figure 8 shows four different leaves. The outputs from our algorithm look quite realistic. This shows that the algorithm works well for many shapes.

5. Image enhancement of the output

To improve the realism of the output (Figure 7a) the depth (thickness) and the texture have been fabricated. Starting from the plain output the thickness of veins are increased and the noise is added to create some texture. The Gaussian operator (3 × 3) then applies to smooth the image (Figure 7b). To emboss the image, the intensity value is used to represent the thickness. The intensity along the 45 degree incident angle is calculated and a pseudo color is added (Figure 7c). The image now will look like the back side of a leaf with the petiole protruded. This image is then inverted (Figure 7d). However, the color of this image is not like a real leaf. Therefore color scanned from a real leaf is used (Figure 7e). Mixing the images from the appropriate steps creates the multi-tone color (Figure 7f).



(7a) *(7b)* *(7c)*

(7d) *(7e)* *(7f)*

Figure 7. Image enhancement process: (7a) The original output. (7b) Add noise and blur filter. (7c) Using 3D graphics techinique, see detail in the paper. (7d) Inversed from fig.7c. (7e) Color & intensity adjustment. (7f) Change Fig.7b to white and adding to Fig.7e.

The picture of the result contains more information than what can be described in words. The output image can be compared with the real leaf. Both the main and secondary veins look like the real leaf. The small details are also well constructed.

6. Conclusion

This paper proposes an algorithm to synthesize vein images from a given leaf shape. The algorithm is inspired by nature using the novel idea based on particle systems. The trails of the motion of interacting particles create very realistic veins. A few parameters control the quality of the output. The image enhancement has been introduced to improve the realism of the output. There are numerous forms of leaves, it is impossible for this algorithm to work well on all of them. However, for a large class of leaf, this algorithm produces the output that looks very natural.

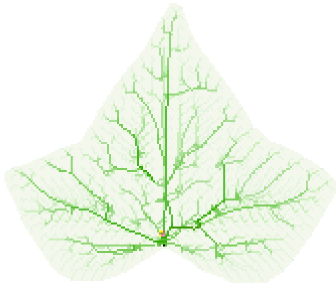
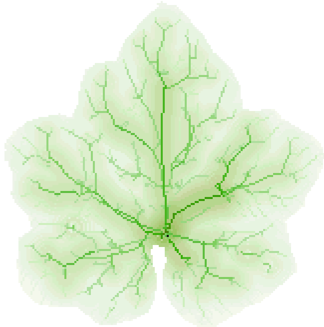
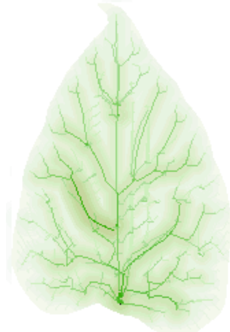
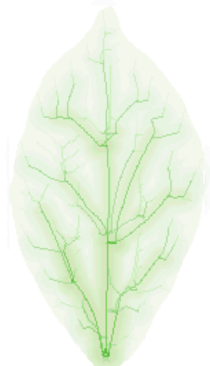
Real leaf	From Algorithm	From Algorithm
		
180x160 Pixels	Space 5 Pixels (500 Particles) Square	
		
160x167 Pixels	Space 5 Pixels (674 Particles) Square	
		
160x248 Pixels	Space 5 Pixels (537 Particles) Square	
		
141x267 Pixels	Space 12 Pixels (172 Particles) Square	

Figure 8. Image of leaves and outputs.

References

[1] P. Prusinkiewicz, and A. Lindenmayer , *The algorithmic beauty of plants*, Springer-Verlag, New York, 1990. With J. S. Hanan, F. D. Fracchia, D. R. Fowler, M. J. M. de Boer, and L. Mercer.
[2] P. Prusinkiewicz, M. Hammel, J. Hanan, and R. Mech. "L-systems: from the theory to visual models of plants". In M. T. Michalewicz, editor, *Proceedings of the 2nd CSIRO Symposium on Computational Challanges in Life Sciences*. CSIRO Publishing, 1996.
[3] P. Prusinkiewicz, M. Hammel, J. Hanan, and R. Mech. "Visual models of plant development". In G. Rozenberg and A. Salomaa, editors, *Handbook of formal languages*. Springer-Verlag, 1996.
[4] P. Prusinkiewicz, M. S. Hammel and E.Mjolsness, "Animation of plant development", Proceedings of SIGGRAPH 93 (Anaheim, California, August 1-6, 1993), In *Computer Graphics* Proceedings, Annual Conference Series, 1993, ACM SIGGRAPH, pp. 351-360
[5] P. Prusinkiewicz, M. James and R. Mech, "Synthetic topiary", Proceedings of SIGGRAPH 94 (Orlando, Florida, July 24-29, 1994). In *Computer Graphics* Proceedings, Annual Conference Series, 1994, ACM SIGGRAPH, pp. 351-358.
[6] R. Mech, and P. Prusinkiewicz, "Visual Model of Plants Interacting with Their Environment", Proceedings of SIGGRAPH 96 (New Orleans, Louisiana, August 4-9, 1996). In *Computer Graphics* Proceedings, Annual Conference Series, 1996, ACM SIGGRAPH, pp. 397-410.
[7] P. Prusinkiewicz, "Modeling and Visualization of Biological Structures", In *Proceeding of Graphics Interface '93*, pages 128-137, May 1993. Held in Toronto, Ontario, 19-21 May 1993.
[8] J. L. Power, A.J. B. Brush, P. Prusinkiewicz and D. H. Salesin, "Interactive Arrangement of Botanical L-System Models", Symposium on Interactive 3D Graphics Atlanta, 1999.
[9] N. Greene,"Voxel Space Automata: Modeling with Stochastic Growth Processes in Voxel Space", *Computer Graphics*, vol.23, no.3, July 1989, pp. 175-184.
[10] I. Shlyakhter, M. Rozenoer, J. Dorsey and S. Teller, "Reconstructing 3D Tree Models from Instrumented Photographs", *IEEE Computer Graphics and Applications*, May/June 2001.
[11] O. Deussen, P. Hanrahan, B. Lintermann, R. Mech, M. Pharr and P. Prusinkiewicz, "Realistic Modeling and rendering of plant ecosystems", SIGGRAPH'98, Orlando, Florida, July 19-24, 1998.
[12] N. Chiba, K. Ohshida, M. Muraoka and M. Miura, "Visual simulation of botanical trees based on virtual heliotropism and dormancy break", *The Journal of Visualization and Computer Animation*, 5,(1), 3-15 (1994).
[13] N. Chiba, K. Ohshida, M. Muraoka and N. Saito, "Visual simulation of Leaf Arrangement and Autumn Colours", *The Journal of Visualization and Computer Animation*, 7, 79-93 (1996).
[14] N. Chiba, K. Ohshida, M. Muraoka, M. Miura and N. Saito, "A growth model having the abilities of growth-regulations for simulating visual nature of botanical trees", *Computer & Graphics*, 18 (4), 496-479 (1994).
[15] G. Kokai, Z. Toth and R. Vanyi, "Evolving Artificial Trees Described by Parametric L-systems", *IEEE Canadian Conf. on Electrical and Computer Enginerring*, 1999, 3, pp. 1722-1727.
[16] S. Chuai-aree, S. Siripant and C. Lursinsap, "Animating Plant Growth in L-System by Parametric Functional Symbols", *Proceedings of The First International Conference on Intelligent Technologies* (InTech'2000), Dec 2000, pp. 135-143. Held in Bangkok, Thailand, 13-15 Dec 2000.
[17] G. Kokai, R. Vanyi, Z. Toth, "Parametric L-System Description of the Retina with Combined Evolutionary Operators", *Proceedings of the Genetic and Evolutionary Computation Conference* [*GECCO'99*], Vol. 2, 1999, pp. 1588-1595.
[18] Y. Rodkaew, S. Siripant, C. Lursinsap and P. Chongstitvatana, "Generate Leaf Shapes using L-system and Genetic Algorithms", *Proceedings of The First International Conference on Intelligent Technologies* (InTech'2000), Dec 2000, pp. 154-158. Held in Bangkok, Thailand, 13-15 Dec 2000.
[19] C. Lursinsap, P. Sophatsathit and S. Siripant, *Simulation of Leaf Growth Based On A Rewriting System: A Unified Leaf Model*, Advance Virtual and Intelligence Computing Research Center (AVIC), Chulalongkorn University, Techinical Report No. 01.01.2000.
[20] M. S. Hammel, P. Prusinkiewicz and B. Wyvill, "Modeling compound leaves using implicit contours", In Tosiyasu L. Kunii, editor, *Visual computing: integrating computer graphics with computer vision*, pages 119-212. Springer-Verlag, June 1992. Proceedings of Computer Graphics International '92 (Tokyo, Japan, 22-26 June 1992).
[21] K. Kaino, "Geometry of folded pattern of veins and origami model of digitate leaves", *Forma*, 9, 253-257 (1994).
[22] D.E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Redwoord City, 1989.
[23] N. Kato, T. Okuno, A. Okano, H. Kanoh and S. Nishihara, "An ALife Approach to Modeling Virtual Cities", *IEEE International Conf. on Systems, Man, and Cybernetics 1998*, 1998, 2, pp. 1168-1173.