Modeling Autumn Sceneries

Brett Desbenoit†, Eric Galin†, Samir Akkouche† and Jérome Grosjean‡

†LIRIS, CNRS, Université Claude Bernard Lyon 1, France
‡LSIIT, CNRS, Université Louis Pasteur, France

Abstract

This paper presents a system for modeling autumn leaves covering vegetation and monuments. We simulate the coloring and aging process by a stochastic model that represents the probability of evolution of a leaf according to the characteristics of the environment. We distribute leaves over the ground by approximating their complex movement by trajectory templates such as fluttering, rolling and tumbling. Leaves stack onto the ground in successive layers so as to improve the collision detection step.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Three-Dimensional Graphics and Realism

Keywords: *ecosystem simulation, autumn coloring, aging and weathering.*

1. Introduction

Modeling complex and realistic synthetic landscapes covered with vegetation such as forests, meadows or gardens is a challenging and important problem in computer graphics. The challenge stems not only from the complexity and diversity of biological species interacting together and with their environment, but also from the many details that need to be modeled and rendered to create realistic images. Details not only include small geometric and color defects produced by aging and weathering phenomena, but also small plant such as leaves or lichens which are conspicuous in a natural scene.

In this paper, we present a system for modeling autumn leaves covering vegetation and monuments. Modeling aging leaves and disseminating them in a virtual scene to simulate autumn sceneries is a challenging problem. Autumn leaves show a vast variety of texture patterns, a large palette of colors, and complex deformed shapes. Existing techniques for synthesizing the variations of colors in autumn [COMS96, MCK*01, BDE04] do not generate realistic texture patterns. In our approach, leaves are organized into an atlas of template geometric models created from scanned images so as to capture the diversity and the complexity of shapes and texture patterns.

The characteristics of the environment such as the temperature, or the amount of sunlight and wetness have a correlated influence over the aging process. Several techniques such as Open L-Systems [MP96] and Open Diffuse Limited Aggregation [DGA04] exist for modeling the interactions between plants and their environment. In this paper, we present a method for simulating the coloring and aging process of leaves in autumn by an Open Markov Chain model that represents the probability of evolution of a leaf according to the characteristics of the environment.

The distribution of leaves in a scene results from very complex dynamics, combining leaves falling and flying in the wind, tumbling and rolling and eventually colliding and stacking to the ground. Physically based techniques which have been proposed for animating leaves in wind fields [WH91, WZF*03] are computationally expensive and therefore ill suited for simulating the fall of thousands of leaves. In this paper, we are more concerned by the final distribution of thousands of leaves onto the ground rather than by the accurate simulation of the movement of a single leaf. Thus, we approximate the complex trajectories of leaves by template movement models such as fluttering, tumbling or spiral fall.

2. The leaf aging process

Leaf models are created from scanned images which enables us to capture the complex texture patterns of real leaves.

© The Eurographics Association 2006.

delivered by EUROGRAPHICS DIGITAL LIBRARY www.eg.org diglib.eg.org

The triangle representation is created by using a Delaunay triangulation of the polygonal silhouette of a leaf from the scanned images as presented in [MMPP03]. The triangulation may be generated at an arbitrary resolution, which enables us to adapt the number of generated triangles to the required level of detail (Figure 1). Deformations such as large wrinkles or folds are obtained by assigning a mass-spring system to the triangulated polygon to constrain the surface area to remain constant and by moving some of its vertices.

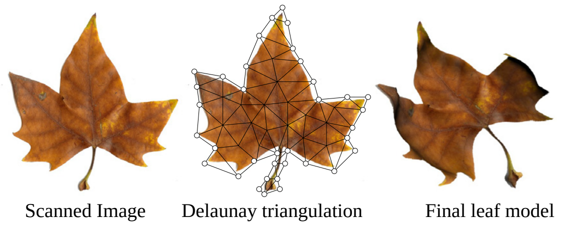

Figure 1: *Overview of the leaf modeling process*

The leaf aging process is represented by an Open Markov Chain model which is implemented as graph. The nodes of the graph, denoted as $\mathcal{S}_i$ $0 \leq i < n$ where n denotes the number of nodes, represent the possible leaf states and store a reference to template leaf models of the leaf atlas.

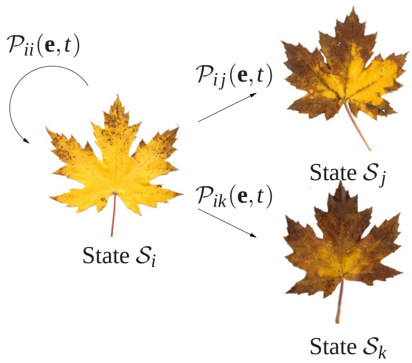

Figure 2: *Notations for the Open Markov Chain model*

The arcs $\mathcal{A}(i,j)$ of the graph represent possible ways for one leaf to progressively age into another one and store the probability that a leaf at a given state $\mathcal{S}_i$ will evolve to another state $\mathcal{S}_j$. In our Open Markov Chain model, the transition probabilities are functions of the characteristics of the environment $\mathbf{e}$ and the elapsed time t, thus they will be denoted as $\mathcal{P}_{ij}(\mathbf{e},t)$ (Figure 2).

Characteristics of the environment We take into account the temperature, denoted as θ, which is mostly responsible for triggering autumn coloring, as well as the wetness, denoted as ω, which plays an important part in the decay and putrescence of a leaf. The characteristics of the environments will be denoted as $\mathbf{e} = (\theta, \omega)$.

Temperature is correlated to the direct and indirect lighting as the sun warms exposed and accessible surfaces. Therefore, we create the temperature map, denoted as $\mathcal{T}$ by combining a direct lighting map and an indirect lighting map as described in [DGA04]. The local wetness of the surface is computed by constraining a particle system to the surface of the objects in the scene [DPH96] to track the paths of the droplets of rain and find wet and dry regions. As for the temperature, the wetness is encoded in a map denoted as $\mathcal{W}$.

Overview of the aging process The overall aging process is performed as follows. Given an elapsed time t and a leaf at an initial state $\mathcal{S}_i$, we evaluate the transition probabilities $\mathcal{P}_{ij}(t,\mathbf{e})$ according to the local temperature and wetness. The characteristics of the environment are obtained by evaluating the temperature and wetness in the environment maps $\mathcal{T}(\mathbf{p})$ and $\mathcal{W}(\mathbf{p})$ which are computed at the location of the leaf $\mathbf{p}$. Then, we check the evolution of the leaf according to the computed probabilities. If the state of the leaf changes, we update the reference to the template leaf model in the atlas, otherwise the leaf remains unchanged.

The half life aging model The probability that a leaf will change its state is inspired by the way radioactive atoms decay in a given period of time. Recall that radioactive decay proceeds according to the half life principle. The half life is the amount of time necessary for one half of the radioactive element to decay.

We define the half life τ_i of a leaf state $\mathcal{S}_i$ as the amount of time necessary for half the number of those leaves in that state to decay to another state. The corresponding decay constant, denoted as λ_i, is defined as:

$$\lambda_i = \frac{\ln 2}{\tau_i}$$

In our system, the half life is a function of the temperature and wetness of the leaf and will be denoted as a function $\tau_i(\mathbf{e})$. The corresponding decay function will be denoted as $\lambda_i(\mathbf{e})$. The probability that a leaf will not change its state $\mathcal{S}_i$ after an elapsed time t is defined as:

$$\mathcal{P}_{ii}(\mathbf{e},t) = e^{-\lambda_i(\mathbf{e})\,t} \qquad 0 \leq i < n$$

Every leaf state may have a different decay function so that leaves at different states may age at a different speed.

For every arc $\mathcal{A}(i,j)$ in the graph of the Open Markov Chain model, the probabilities $\mathcal{P}_{ij}(\mathbf{e},t)$ are defined by a transition function denoted as $X_{ij}(\mathbf{e})$. Those functions define the relative distribution of a leaf at state $\mathcal{S}_i$ aging to other states $\mathcal{S}_j$. Since the probability that a leaf ages in a time t is $1-\mathcal{P}_{ii}(\mathbf{e},t)$, we have:

$$\mathcal{P}_{ij}(\mathbf{e},t) = (1-\mathcal{P}_{ii}(\mathbf{e},t))\,X_{ij}(\mathbf{e}) \qquad 0 \leq i < n \qquad i \neq j$$

Influence of the environment The half life functions $\tau_i(\mathbf{e})$ are defined as a bilinear interpolation of four coefficients denoted as $\tau_i(0,0)$, $\tau_i(1,0)$, $\tau_i(0,1)$ and $\tau_i(1,1)$ which repre-

© The Eurographics Association 2006.

sent the half life of state $\mathcal{S}_i$ in the four following extreme cases: wet and cold, wet and warm, dry and cold and dry and warm. The transition functions $X_{ij}(\mathbf{e})$ are defined in the same way as a bilinear interpolation of four constants. Thus, the designer has to define four constants for every arc $\mathcal{A}(i,j)$ and every state $\mathcal{S}_i$ in the Open Markov Chain model.

3. Modeling leaf stacks

Our method proceeds in three steps as follows. First we evaluate which leaves will detach from the branches of the trees and will be falling according to the speed of the wind and the state of the leaf. Then, we distribute the leaves in the scene by applying template procedural trajectory models. Finally, we create leaf stacks by performing an accurate collision detection and rigid body simulation.

Leaf detachment step Leaves have a different resistance to the wind in their life cycle, and become less and less resistant as they age. In our system, every leaf state $\mathcal{S}_i$ in our Open Markov Chain model implements a function $\mathcal{D}_i(s,t)$ which represents the probability that a leaf detaches and falls as the strength of the wind s increases. If the leaf detaches, it is inserted into the set of flying leaves and will be processed by the leaf falling and stacking algorithms. Otherwise, it remains attached in the tree.

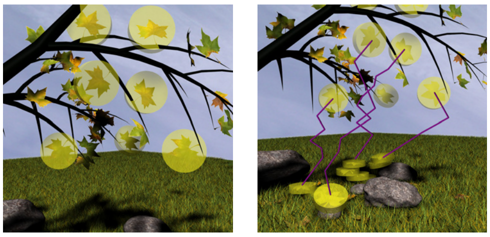

Figure 4: *Leaf detachment and falling step*

The detachment probability function $\mathcal{D}_i(s,t)$ is defined as follows:

$$\mathcal{D}_i(s,t) = e^{-\delta_i t}\, e^{-s^2/\sigma_i^2}$$

The constant δ_i is defined as $\delta_i = \ln 2/\tau_i$ where τ_i represents the amount of time necessary for half the number of leaves to detach and fall without wind. The constant σ_i characterizes the sensitivity of a leaf to the strength of the wind.

Simulating the trajectories of leaves The movement of falling leaves is defined by template trajectory models which are applied according to the geometry and weight of the leaves, and to the speed of the wind (Figure 5).

Trajectories are implemented as a set of relative displacement vectors that represent the speed of the leaf at every time step. The final movement of a leaf is obtained by summing the trajectory displacement vectors with the wind speed vector field at every time step.

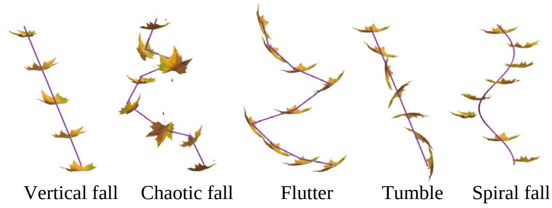

Figure 5: *Fundamental leaf falling patterns*

The movement of leaves blown by the wind and rolling and tumbling on the ground is very complex because of the many collisions and the highly turbulent flows near the ground. As for falling leaves, we invoke template different trajectory models according to the speed of the wind. Under mild gusts of wind, leaves show sliding, rolling and tumbling patterns depending on the geometry of the ground. As the strength of the wind increases however, leaves may be captured in vortices and turbulences which propels them in the air, so that the leaves fly for a few seconds before falling again to the ground.

Leaves have very complex geometry which makes collision detection between falling leaves as well as with other static elements in the scene computationally demanding. Therefore, we rely on bounding cylinders to speed up the leaf collision detection process during the distribution process.

We speed up the collision detection process by sampling the geometry of the leaf models by sample points and by computing the intersection between the trajectory of those points and the other objects in the scene.

Creating leaf stacks At the end of the leaf falling process, we obtain a set of leaves floating slightly above each other. Leaves are not forming compact stacks however as their bounding cylinder has been used for collision detection. Therefore, we invoke a final rigid body simulation and collision detection using the real geometry of the leaves.

Figure 6: *Leaf stacking step*

This step which creates dense realistic and complex leaf stacks is the most computationally demanding part of the process. Still, this last step is only necessary for the creation of the final image of a scene. In practice, the leaves packed

© The Eurographics Association 2006.

Figure 3: *Autumn leaves covering the grass and stacking in a wheelbarrow*

according to their bounding cylinder provide a sufficient approximation of the final scene so that the designer may skip the accurate leaf stacking step so as to obtain for a faster feedback when editing the scene.

4. Conclusion and future work

We applied our leaf aging and falling algorithms to model several autumn sceneries. The corresponding images are shown throughout the paper.

The total number of leaves in every scene ranges from 14302 to 25202. The simulation of aging leaves was performed in less than one second. The falling step required less than 3 minutes for the most complex case which involved 7439 falling leaves, whereas the leaf packing step required between half and hour and two hours for the creation of the largest stacks.

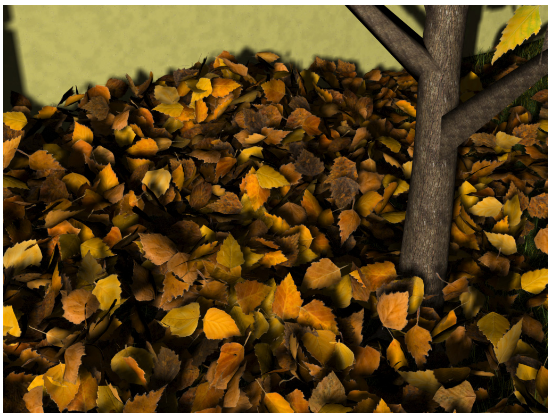

Figure 7: *Leaves forming dense stacks near a wall*

In a near future, we plan to further investigate more complex decay models for leaves, including dry leaves being torn and broken into pieces or wet leaves putrefying. Wet leaves should be handled as deformable models so that they should stick together and deform according to the geometry of nearby objects. We are also working on the influence of fallen leaves over the environment in an information feedback loop. Fallen leaves should retain moisture and therefore affect the environment, change its parameters locally and favor or limit the growth of other kinds of plants, such as mosses or lichens.

References

[BDE04] BRAITMAIER M., DIEPSTRATEN J., ERTL T.: Real-time rendering of seasonal influenced trees. In *Theory and Practice of Computer Graphics* (2004).

[COMS96] CHIBA N., OHSHIDA K., MURAOKA K., SAITO N.: Visual simulation of leaf arrangement and autumn colors. *The Journal of Visualization and Computer Animation 7* (1996), 79–93.

[DGA04] DESBENOIT B., GALIN E., AKKOUCHE S.: Simulating and modeling lichen growth. *Computer Graphics Forum (Proceedings of Eurographics) 23*, 3 (2004), 341–350.

[DPH96] DORSEY J., PEDERSEN H. K., HANRAHAN P. M.: Flow and changes in appearance. In *Proceedings of SIGGRAPH* (1996), pp. 411–420.

[MCK*01] MOCHIZUKI S., CAI D., KOMORI T., KIMURA H., HORI R.: Virtual autumn coloring system based on biological and fractal model. In *Proceedings of Pacific Graphics* (2001), pp. 348–354.

[MMPP03] MUNDERMANN L., MACMURCHY P., PIVOVAROV J., PRUSINKIEWICZ P.: Modeling lobed leaves. In *Computer Graphics International* (2003), pp. 60–65.

[MP96] MĚCH R., PRUSINKIEWICZ P.: Visual models of plants interacting with their environment. In *Proceedings of SIGGRAPH* (1996), pp. 397–410.

[WH91] WEJCHERT J., HAUMANN D.: Animation aerodynamics. In *Proceedings of SIGGRAPH* (1991), pp. 19–22.

[WZF*03] WEI X., ZHAO Y., FAN Z., LI W., YOAKUM-STOVER S., KAUFMAN. A.: Blowing in the wind. In *Symposium on Computer Animation* (2003), pp. 75–85.

© The Eurographics Association 2006.