EUROGRAPHICS 2004 / M.-P. Cani and M. Slater
(Guest Editors)

Volume 23 (2004), Number 3

Simulating and modeling lichen growth

Brett Desbenoit, Eric Galin and Samir Akkouche

LIRIS, CNRS, Université Claude Bernard Lyon 1, France

Abstract

This paper presents a system for modeling lichens and simulating their propagation and growth in a virtual scene. Lichens colonize almost every substrate in nature and play an important role in the visual appearance of a natural object. The propagation of lichens over the substrate is performed by an Open Diffusion Limited Aggregation model constrained by the characteristics of the environment. The designer can control the development of lichens with simple parameters. Rendering the complex geometry and texture of lichens is achieved by instantiating three dimensional lichen models stored in an atlas of shapes created after real world images. The lichens obtained by our approach considerably increase the realism of complex natural scenes.

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

Keywords: *plant modeling, ecosystem simulation, lichen growth, environment sensitive morphogenesis.*

1. Introduction

Synthesizing realistic images of landscapes covered with vegetation is a challenging and important problem in computer graphics. Complex scenes include natural living ecosystems such as forests [DHL*98], gardens [PJM94], stone statues or monuments, and even fallen leaves in autumn. The challenge stems from the complexity and diversity of biological species interacting together and with their environment. Beautiful and realistic images of complex natural environments have been produced by many computer graphics researchers and artists in the film industry. Unfortunately, the rendered models are often too perfect, which betrays the synthetic nature of the scene. Several techniques simulating the physical and chemical phenomena that age stone [DEL*99, CDM*02] and metallic structures [DH96] in a natural environment have been proposed. Although lichens are present almost everywhere in nature, the impact of this biological species has not received much attention in the computer graphics community.

In this paper, we present a system for modeling lichens and simulating their propagation and growth in a virtual scene. Lichens are very resistant and may live on many different structures and materials such as stone, wood and even corroded metal. They play an important role in the realism of a natural scene. The presence of lichens provides the viewer with a hint about the age and orientation of trees or stones in an ecosystem, as well as indirect indications about the weather and the characteristics of the environment.

Figure 1: *Lichens growing over a fallen branch.*

1.1. Related work

The growth of lichens is a biological morphogenesis process that changes the appearance of the object which is colonized.

Our work relates to aging and weathering techniques, and to environment sensitive plant growth methods.

Aging and weathering Several techniques for modelling the imperfections and the degradations of an otherwise original object have been proposed. Physically based methods simulate both physical and chemical phenomena such as the erosion of stone by water [DEL*99], corrosion in marine environments[CS03], rust formation [MDG01, CS03], patina covering buried objects [CS00], metallic patinas produced by rain [DH96], cracks [HTK98] and fractures [OH99]. Microscopic phenomena such as the oxidation of metal generally involve complex simulations that generate a texture that stores the imperfections and changes in appearance, whereas larger scale phenomena such as fractures, erosion or cracks produce local or global changes in the geometry of the model. Several procedural approaches for modelling surface imperfections such has dust accumulation [WNH97] or impacts [EPD01] have been proposed as well. Contrary to physically based techniques, those methods provide greater control and avoid time consuming simulations.

Ecosystem simulations Simulating the influence of the environment over the development of plants living in a complex ecosystem has received a lot of attention in the computer graphics community. Měch and Prusinkiewicz [MP96] proposed an Open L-System to model the interactions between plants and the environment. Deussen et *al.* [DHL*98] proposed a method that simulates the growth of plants competing for territory. Lane and Prusinkiewicz [LP02] extend this principle to a whole ecosystem including different types of plants. The parameters of the environment affect both the growth of the plants and the spatial distribution of the different species. Recent methods evaluate direct and indirect lighting accurately and efficiently by hierarchical radiosity techniques that simulate the radiant energy transfer between plant models and other objects in the scene [SSBD03].

Modelling lichens Sumner [Sum] developed a mathematical model based on a Diffusion Limited Agregation (DLA) model [WS81]. The principle consists in moving a particle randomly in an area containing an aggregate of particles and solidifying its position if it gets into contact with the cluster. This mathematical model makes it possible to create shapes with a multitude of dendrites organized into a fractal pattern. The proposed method provides some control over the shape of the one lichen cluster by parameterizing the probability of aggregation by a function of the local curvature. Unfortunately, this approach does not take into account the characteristics of the environment and creates almost perfect regular patterns with equally sized lobes that cannot be found in nature. Moreover, hundreds of thousands of particles are needed to generate a single lichen cluster, which is both time and memory demanding.

Recently, Cuttler [CDM*02] presented a weathered stone statue model where lichens were represented by colored particles. Those methods that model lichens as textures fail at creating convincing realistic images as they do not capture the complex three dimensional fractal patterns of lichens.

1.2. Contributions

The complexity and diversity of the lichen patterns, shapes and textures [BSS01] makes modeling every single lichen by hand impossible. Our approach consists in distributing spores over the objects in the scene, simulating the lichen propagation to capture the complex distribution patterns of real world lichens, and computing the final complex geometry using an atlas of textured mesh models for every kind of lichen as first suggested in [FPB92]. Our contributions are as follows:

Lichen atlases We propose a system for modeling three dimensional lichens after real world images and storing template models into an atlas. Our complex lichen models capture the complexity and diversity of the lichen patterns, shapes and textures. Lichens colonizing surfaces are instantiated from the template models of the atlas so as to save memory.

Lichen seeding and propagation We simulate seeding by dispersing lichen spores over the objects in the scene and transporting or removing spores that are blown away by the wind or washed by rain water flows. The propagation of lichens is performed by an original Open Diffusion Limited Aggregation (Open DLA) model that captures the development of several lichens influenced by the environment, limited by self collisions and competing for space.

Control The designer may control the patterns formed by lichen growth by adjusting the biological parameters of the lichen and its sensitivity to direct and indirect lighting and moisture. The characteristics of the environment are coded as texture maps that may be generated either automatically by physically based simulations, or created by hand by the designer.

The remainder of this paper is organized as follows. Section 2 recalls the fundamentals of lichen biology and presents an overview of the important parameters that influence its development. In Section 3, we describe our lichen modeling system and address lichen seeding, propagation and atlas instantiation. Section 4 presents some implementation details of our Open DLA algorithm, as well as timings. Eventually, we show some images illustrating realistic lichen growth over synthetic objects in complex natural environments in Section 5.

2. Fundamental concepts in biology

Lichens are composite, symbiotic organisms made up from members of as many as three kingdoms. The dominant partner is a fungus. Fungi are incapable of making their own

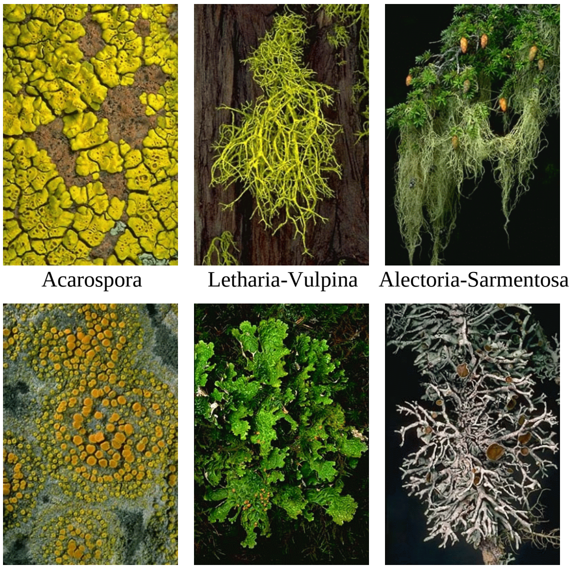

Figure 2: *Different types of lichens (courtesy of Sylvia Sharnoff and Stephen Sharnoff).*

food and usually provide for themselves as parasites or decomposers. The lichen fungi (kingdom Fungi) cultivate partners that manufacture food by photosynthesis. Sometimes the partners are algae (kingdom Protista), other times cyanobacteria (kingdom Monera), formerly called blue-green algae.

Classification Lichens may be arbitrarily classified into three growth forms that do not reflect how the lichens are related to each other. Different species within a genus may have different growth forms (Figure 2).

Crustose lichens form crusts that are so tightly attached to the rocks, trees, sidewalks, or soils they grow on that they cannot be removed without damaging the substrate. Foliose lichens are somewhat leaf-like, composed of lobes. They are relatively loosely attached to their substrates, usually by means of rhizomes. Their lobes have upper and lower sides and usually grow parallel to the substrate. Fructicose lichens are the most three-dimensional. They are usually round in cross section, and most are branched. They can be like little shrubs growing upward, or they can hang down in long strands.

Lichen growth Lichens grow in the leftover spots of the natural world that are too harsh or limited for most other organisms. They are pioneers on bare rock, desert sand, cleared soil, dead wood, animal bones, rusty metal, and living bark. Able to shut down metabolically during periods of unfavourable conditions, they can survive extremes of heat, cold, and drought.

Given appropriate amounts of light and moisture, clean air, and freedom from competition, lichens can colonize almost any undisturbed surface. Most lichens, especially crustose lichens, grow very slowly, often less than one millimetre per year. Fructicose lichens are an exception as they can grow up to 20 millimetres per year. Lichens grow from spores that are transported not only by the flow of droplets of rain and the wind, but also by small animals and insects. Therefore, spores may be almost transported anywhere, but the local characteristics of the environment will either favour or prevent them from springing a new lichen.

Lichens grow very slowly because the photosynthesis is slowed by the strands of the fungi covering the cells of the algae which filters incoming light. Therefore, lichens need a lot of light, although for most species, only indirect lighting favours growth, whereas strong direct sun lighting burns the cells and is a limiting factor. Lichens also need water and moisture to develop, and develop on porous substrates such as wood or porous stone that retain waters.

3. Modeling and simulating lichen growth

Our goal is to provide the designer with simple tools to control the location, the distribution pattern and the geometry of lichens in a natural scene. The creation of lichens is performed in three steps (Figure 3). First, we spread lichen spores over the objects that will be colonized (Section 3.1). Then, lichens propagate over the surface, competing for space and favourable conditions (Section 3.2). Spores are converted into a particle system constrained to propagate over the triangle mesh of the colonized objects. The lichen patterns are obtained by invoking an Open Diffuse Limited Aggregation (Open DLA) model that takes into account the characteristics of the environment. Eventually, we create the complex geometry and texture of lichens (Section 3.3). The particles spread over the surface of colonized objects are transformed into textured mesh models by instantiating template lichen models stored in a lichen atlas.

3.1. Seeding lichens

In this section, we address the dispersion of spores that will develop and propagate colonies over the substrate if the proper lighting and moisture conditions are met. All those physical and biological parameters are very hard to simulate. The seeding process is a key step of the algorithm as it characterizes the distribution of lichens in the scene. Locating spores seeding lichens by hand would be cumbersome though. We have developed two complementary techniques for controlling the dispersion of lichen seeds in a virtual ecosystem.

Wind and water flow simulations In nature, we observe that most lichens develop on parts of surfaces that are easily accessible but also protected from gusts of wind and rain

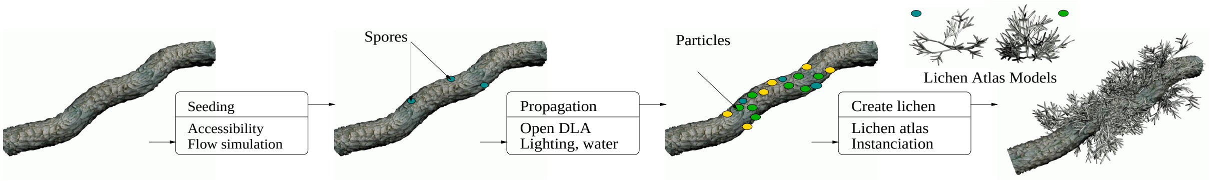

Figure 3: *Overview of our lichen growth simulation*

flows. We propose a global procedural approach that consists in randomly dispersing spores in some regions of the scene. In our system, the designer may control regions where spores are spawned with bounding spheres.

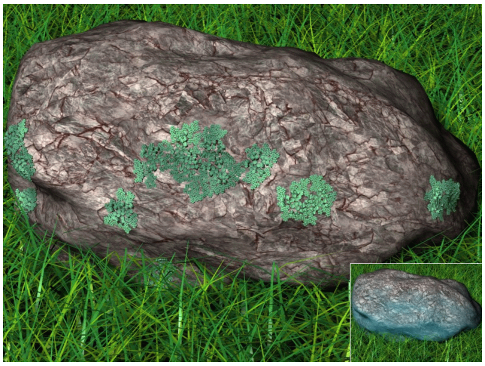

Figure 4: *Seed regions and lichen growth using accessibility and water flow simulations.*

Spores stick to the surface depending on the characteristics of the substrate, its local geometry, the type of the lichen and the local accessibility [Mil94] of the surface. A candidate spore that lies in a highly accessible region is removed to take into account the fact that wind may blown the spore away. Spores that lie in a region that is hardly accessible are also removed. Flow simulations may be performed as proposed in [DPH96] to simulate rain washing and transporting spores. Figure 4 shows a stone colonized by lichens. Spores may stick to the stone on its side, near the grass, in regions that are not exposed to the rain and the wind. Spores were randomly spread in light-blue regions with medium accessibility.

Painting Since the previous approach is computationally demanding, we propose a second technique that consists in spreading spores directly over the surface of the objects in the scene by painting.

The color of the virtual paint defines the local density of spores. This approach provides the designer with a tight control over the seeding process. Figure 5 illustrates how a lichen symbol was created by constraining spores to areas painted by the designer.

Figure 5: *Seeding regions characterized by painting.*

3.2. Lichen propagation

Our propagation algorithm proceeds in two steps. The first step evaluates the moisture and lighting parameters over the surface of colonized objects. Simulations take into account the trajectory of the sun in the sky hemi-sphere, the dispersion of rain water and moisture sources such as rivers or lakes. Those parameters are stored in the description of the objects in the scene graph to speed up future queries.

The second step invokes an Open Diffusion Limited Aggregation (Open DLA) model that distributes particles over the surface forming a multitude of dendrites organized into a fractal pattern. The Open-DLA model enables us to capture the development of several lichens influenced by the environment and competing for space in favourable areas. Lichens stop growing at the border of unfavourable regions or when they tend to fold onto themselves.

3.2.1. Characteristics of the environment

Simulating the characteristics of the environment is essential to obtain a realistic distribution of the lichens in the scene as well as realistic lichen patterns.

Lighting Lighting plays a major role in the growth of lichens. Indirect lighting is favourable to most lichens, whereas direct lighting limits their development. Therefore, we create two different lighting maps. Direct and indirect lighting may be evaluated accurately and efficiently by hierarchical radiosity techniques that simulate the radiant energy transfer between plant models and other objects in the scene as proposed in [SSBD03]. Those methods are complex however, and must be applied to all the objects in the scene.

In our system, we only need to evaluate lighting for objects that will be colonized. Those objects may be easily identified in the scene graph as the nodes that hold the lichen spores spread during the seeding step. We rely on stochastic Monte Carlo ray tracing techniques to approximate direct and indirect lighting. The triangles of the meshes are reorganized into a BSP data structure to speed up computations. We compute the amount of direct light as proposed in [MP96] by evaluating the integral of the amount of light coming from the sun in the day. The trajectory of the sun is discretized into a set of light sources. We avoid aliasing by using a different stochastic sampling of the trajectory of the sun after the method presented by Keller et Heidrich in [KH01].

Moisture Moisture also plays an important part in the growth of lichens. Lichens are very resistant to drought however, as they are able to shut down metabolically during periods of unfavourable water conditions. Simulating the flow of rain water in a whole scene as well as the influence of water sources such as rivers or ponds would be a computationally demanding process though. Therefore, in our system, the moisture map is defined by the user with a three dimensional painting interface that allows him to define which regions in the scene are wet or dry.

3.2.2. The Open DLA algorithm

In this section, we present our Open DLA algorithm that simulates the propagation of lichens over the substrate. Implementation details are provided in Section 4.

Spores are first transformed into seed particles that will progressively grow into clusters competing for space and favourable development conditions with the others. The standard DLA model [WS81] aggregates particles into clusters that form shapes with multiple dendrites organized in a fractal pattern. In this approach, the probability of aggregation of a particle, denoted as $\mathcal{P}(\mathbf{p})$, does not take into account the characteristics of the environment. In contrast, our Open DLA model defines $\mathcal{P}(\mathbf{p})$ as a combination of a theoretical probability of aggregation function, denoted as $\mathcal{A}(\mathbf{p})$, that characterizes the shape of the cluster, and a probability function $\mathcal{E}(\mathbf{p})$ that defines whether the environment favours or limits lichen growth.

Starting from a seed particle, the Open DLA algorithm incrementally creates a cluster of particles by randomly moving new particles over the mesh and aggregating them to a cluster whenever they come into contact. Therefore, lichen growth is inhibited by neighbouring lichens that have already colonized a surface. Figures 5-6 illustrate several lichens competing for space: clusters stop growing at their interface.

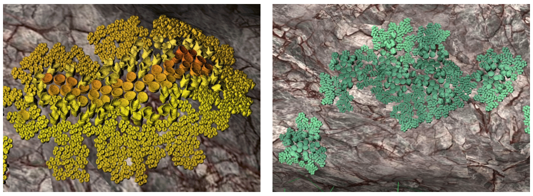

Figure 6: *Several lichen clusters competing for space.*

We control lichen propagation by evaluating the probability of aggregation of a particle during its random walk along the surface. Let $\mathbf{p}$ denote a particle, the probability of aggregation of a particle is characterized by a distribution function, denoted as $\mathcal{P}(\mathbf{p})$, defined as follows:

$$\mathcal{P}(\mathbf{p}) = \mathcal{E}(\mathbf{p}) \times \mathcal{A}(\mathbf{p})$$

Theoretical aggregation The probability of aggregation of a particle, denoted as $\mathcal{A}(\mathbf{p})$, is a function of the local density of neighbouring particles that characterizes the lichen pattern without the influence of the environment. Let $n(\mathbf{p})$ denote the number of neighbouring particles, we rely on the following distribution function that follows the *Pareto* law (rich get richer):

$$\mathcal{A}(\mathbf{p}) = \alpha + (1-\alpha)\, e^{-\sigma\,(n(\mathbf{p})-\tau)^2}$$

The number of neighbouring particles $n(\mathbf{p})$ is evaluated by searching the number particles within a distance ρ of $\mathbf{p}$. Let r denote the radius of the particle. The parameter ρ has an influence over the resulting lichen pattern: small values ($\rho \approx 2r$) produce compact patterns, whereas large values ($\rho \geq 3r$) create lacy patterns. Figure 7 shows some lichen patterns that were produced with our system by varying the control parameters σ, τ and α in the probability of aggregation $\mathcal{A}(\mathbf{p})$ and the radius ρ involved in the function $n(\mathbf{p})$.

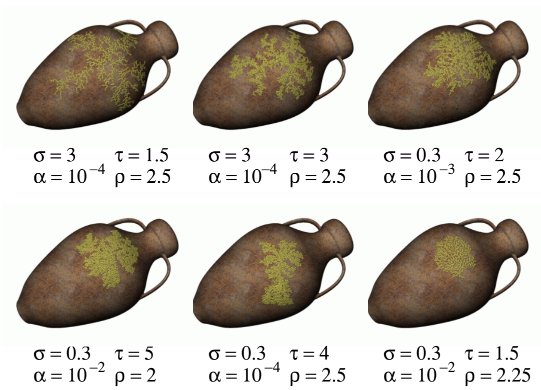

Figure 7: *Patterns obtained with different parameters*

The parameter σ controls the overall aspect of the pattern:

high values result in non isotropic patterns whereas low values tend create rounded shapes. The parameter τ characterizes the shape of dendrites: the lobes or dendrites are the more dense as the parameter τ gets higher. Eventually, α is a bias parameter in the probability of aggregation that forces particles to stick even if they should have not: higher values produce thicker dendrites, whereas lower values create thin strands.

Influence of the environment The environment function, referred to as $\mathcal{E}(\mathbf{p})$, combines the sensitivity of the lichen to indirect and direct lighting and moisture parameters that are evaluated at the center of the particle $\mathbf{p}$:

$$\mathcal{E}(\mathbf{p}) = \min(\mathcal{I}(\mathbf{p}), \mathcal{L}(\mathbf{p}), \mathcal{W}(\mathbf{p}))$$

where $\mathcal{I}(\mathbf{p})$, $\mathcal{L}(\mathbf{p})$, and $\mathcal{W}(\mathbf{p})$ are the functions for indirect lighting, direct lighting, and moisture conditions, respectively. Those functions return values within interval $[0,1]$, high values mean that conditions are favourable, low values mean that conditions are unfavourable. Therefore, $\mathcal{E}(\mathbf{p})$ is equal to 1 if the characteristics of the environment are favourable at the location of a particle $\mathbf{p}$, and equal to 0 otherwise. Every kind of lichen is characterized by optimal direct lighting, indirect lighting and moisture conditions. Figure 8 shows some sensitivity curves for crustose, foliose and fructicose lichens that were derived from biological observations. The curves reflect that the crustose lichen is more resistant to the direct light than the other kinds of lichens, whereas the foliose lichen will tend to develop on porous stone or wood substrate that retain moisture.

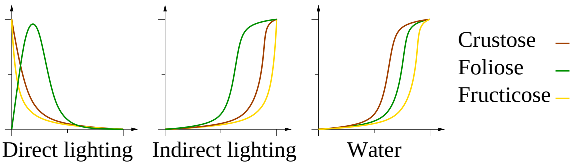

Figure 8: *Lighting and moisture sensitivity for different kinds of lichens*

3.3. Generating the complex shape of lichens

All lichens have very complex color and texture patterns, which makes procedural modeling very difficult. Our approach consists in modeling a variety of realistic lichens shapes after images and storing them in a lichen atlas for every kind of lichen. The particles generated by the lichen propagation algorithm are instantiated into mesh models taken from the lichen atlas, which enables us to save memory and create large areas covered by a variety of lichens efficiently. The instantiation scheme selects models in the database after the type of the lichen, the age and the relative orientation of the particle in the cluster.

Lichen atlases A lichen atlas is organized as a table storing different textured models of the same lichen type that are classified according to their age (Figure 9). Classification by age is an important feature as lichens may have completely different geometries in their life.

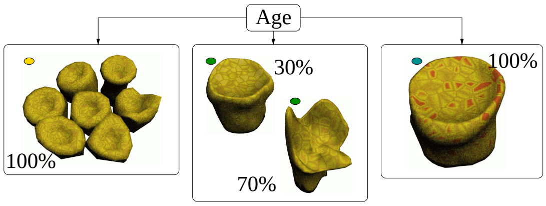

Figure 9: *A lichen atlas with different geometric patterns, figures stand for the relative occurrence probabilities.*

Within each age interval, several models of the same lichen with different characteristics are represented and stored in sub-tables according to their orientation (growing upward or downward) and size. Every model is also characterized by its relative probability of appearance, which can be provided after biological observations.

Lichen instantiation scheme Our lichen instantiation scheme iteratively transforms the particles of an input cluster into a set of three dimensional textured models that are instances selected among the collection of models in the corresponding lichen atlas (Figure 10).

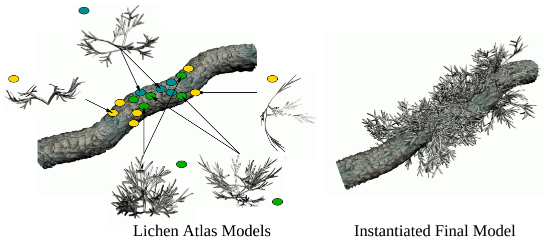

Figure 10: *The shared reference instantiation process*

The selection process proceeds as follows. Given a particle $\mathbf{p}_i$, we first select the table in the lichen atlas corresponding to the age of the particle. Then, we select possible models in the table whose size meet the orientation growth and accessibility requirements. Eventually, we select a model randomly among the different remaining possibilities according to their relative occurrence probability. The selected model is instantiated by a shared reference.

Modeling complex lichens We have developed a interactive method for modeling crustose lichens. Crustose lichens that form almost two dimensional bumpy crusts and shells are modeled using cellular textures [FLCB95] implemented as textured height field models.

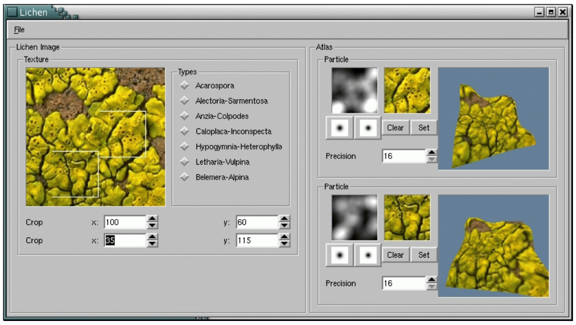

Figure 11: *The cellular texture editor for crustose lichens*

The editing process begins with the selection of a crustose lichen image. The designer selects interesting feature regions in the image to extract smaller images, and interactively defines a corresponding height field for every small image. The final cellular texture is obtained by mapping the sub-image on the height field. Figure 11 shows a screen shot of the interface of our crustose lichen modeling tool.

Crustose lichens form spots that exhibit a radial distribution. Therefore, young particles close to the border of the cluster will have a different shape and texture than older particles at the center of the cluster, which enables us to characterize the relative orientation and position of the cellular textures easily.

Foliose and fructicose lichens that produce three dimensional leaf-like lobes and little shrubs growing upward are created using specific L-Systems [Lin68]. The corresponding texture map is obtained from real world images.

4. Implementation details

In this section, we present the implementation details of our modified Open DLA algorithm. The propagation of lichen over the surface of the colonized objects is performed by constraining a particle system to the triangle meshes. Starting from a seed particle, our method incrementally creates a cluster of particles by randomly moving new particles over the mesh and possibly aggregating them to the cluster when collisions occur.

As a lichen cluster may involve hundred of particles, we need efficient algorithms for moving a particle over the triangle mesh and for detecting the collision between particles.

Triangle mesh representation Triangles meshes are implemented using a half-edge data-structure [CKS98]. In our implementation, a half-edge contains only a reference to a vertex and a reference to its opposite half-edge, which are coded as integers. Triangles are coded as half-edge triples, and the next and previous half edges are procedurally defined as suggested in [CKS98] to save memory. This compact half-edge representation provides us with sufficient topological information for moving a particle over the surface efficiently.

Particle representation Particles store geometrical data characterizing their location over a triangle mesh. A particle is characterized by an index to the lichen cluster and its age in the creation in the lichen cluster. A particle is constrained to the triangle mesh and is geometrically defined by its center, radius, and an index to its supporting triangle. A particle also stores a rotation angle that is used for orienting the instance of the model in the lichen atlas. The normal of a particle is simply defined as the normal of its supporting triangle.

Diffusion Limited Aggregation Let us recall the fundamentals of the DLA algorithm [WS81]. Given a starting particle denoted as $\mathbf{p}_0$, the algorithm incrementally creates a cluster of particles by aggregating particles moving randomly over the surface and that come into contact with the cluster. In general, new particles are created at a fixed distance R of $\mathbf{p}_0$. If the particles move to far away from $\mathbf{p}_0$, they are removed from the simulation.

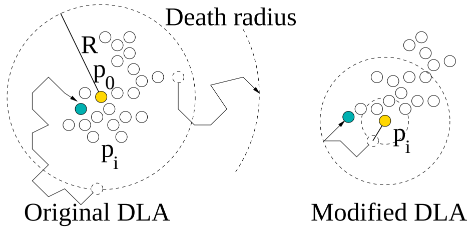

Figure 12: *The original and modified DLA algorithms.*

This approach is computationally demanding as the particles may have very long and complex movement before a collision and aggregation occurs with the cluster. Our modified model improves performance by creating new particles at a small distance from a particle of the cluster. Let $\mathcal{C}$ denote a cluster of particles, and let r denote the radius of the particles (Figure 12). Adding a new particle $\mathbf{p}$ to the cluster is performed as follows:

1. Select a particle $\mathbf{p}_i$ of the cluster $\mathcal{C}$ randomly and create a candidate particle $\mathbf{p}$ at a distance $2r+\varepsilon$ of particle $\mathbf{p}_i$.
2. Move the candidate particle $\mathbf{p}$ in a random direction over the surface with a small distance d.

   2.1 If $\mathbf{p}$ comes into contact with the cluster $\mathcal{C}$, then check if aggregation occurs using the aggregation probability $\mathcal{P}(\mathbf{p}) = \mathcal{E}(\mathbf{p}) \times \mathcal{A}(\mathbf{p})$. If conditions for aggregation are met, stick the candidate particle and add it to the cluster, otherwise remove it from the simulation.

   2.2 If $\mathbf{p}$ collides with another cluster, or moves too far away from $\mathbf{p}_i$, remove it from the simulation.

Step 1 of the algorithm improves speed by adding new particles not too far away from the existing cluster. ε is a parameter that guarantees that new particles are created far away enough from the randomly selected particle $\mathbf{p}_i$ so that collision with the cluster should be avoided at the very first step of the algorithm. In our simulations, ε was set to $r/2$. Step 2.1 of the algorithm ensures that the cluster will not develop in regions where the characteristics of the environment prevent propagation. Step 2.2 of the algorithm performs a collision detection between growing clusters and simulates lichens competing for space.

Moving particles efficiently The position of a particle $\mathbf{p}$ is defined by the index i of its supporting triangle T, and by its center c. Let $\vec{d}$ denote a normalized random direction vector in the plane of the triangle T, and l denote the walk distance. The new position of the particle is computed as follows:

1. Compute the intersection p' between the edges of triangle T with the ray starting at c and with direction $\vec{d}$.
2. If $\|pp'\| > l$, then set the final position of the particle to $p + l\,\vec{d}$ and terminate.
3. Otherwise, the particle moves outside T, so move the particle to p' and set i to the index of the triangle neighbouring T. Let n denote normal of the triangle neighbouring T, the new direction is computed by evaluating the tangent vector $\vec{t} = \vec{d} \wedge \vec{n}$, which defines the new direction $\vec{d} = \vec{n} \wedge \vec{t}$ in the new triangle. Update the distance $l = l - \|pp'\|$ and return to Step 1 of the algorithm until movement ends.

Collision detection Collision detection between a candidate particle and the particles of the growing cluster is the most computationally demanding step of the algorithm. A straight forward approach that checks all collisions between particles has $O(n)$ complexity, which makes the cluster growth a $O(n^2)$ process.

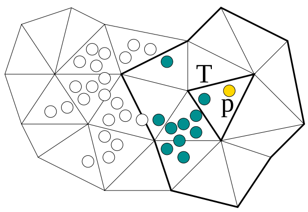

Figure 13: *Collision detection between a particle (yellow) and a subset of the particles of the existing cluster (green).*

We reduce the number of collision detection calls as follows. Given a candidate particle $\mathbf{p}$ in a triangle T, we only compute the collision between $\mathbf{p}$ and the subset of particles that lie in the triangles in the 1-ring neighbourhood of T (Figure 13). We pre-process the triangle meshes to create a table storing the indexes of the triangles in the 1-ring neighbourhood of a given triangle, which enables us to avoid on the fly topological queries that are computationally expensive.

Timings Table 1 reports timings for propagating lichen particles over a variety of mesh models. The two stones, tree, mascot and branch models refer to the images shown in Figures 5, 15, 16 and 1 respectively. The number of triangles of the mesh models, as well as the number of lichen clusters and the overall number of particles involved in the Open DLA algorithm are also reported.

	Triangles	Clusters	Particles	Time
Stone	450	5	505	50
Eurographics	2400	71	2400	483
Tree	450	5	54	1
Mascot	8388	26	2000	150
Branch	8212	24	522	26

Table 1: *Timings (in seconds) for propagating lichen particles using the Open DLA algorithm*

In most cases, the lichen propagation step may be performed in less than one minute. The only exception to this is the Eurographics logo (Figure 5) covered with crustose lichens, which involves many clusters competing for space and therefore requires many collision detection calls.

5. Results

We applied our method to model several lichens propagating over trees, rocks and stone monuments. The corresponding images are shown throughout this document. For reference, some of the real lichens we attempted to model are shown in Figure 2. Spreading lichens and controlling their patterns and geometry in each scene was performed in approximately half an hour.

Figure 14 shows a stone statue in a garden colonized by green algae. The parameters of the environment have been computed automatically. The green algae develop in the regions that are not exposed to the direct light of the sun, and favour shadow regions.

6. Conclusion and future work

We have presented a model that synthesizes different kinds of lichens over surfaces. The growth of the lichen is obtained by using an Open Diffusion Limited Aggregation simulation that models the propagation of lichens interacting with the environment and competing for favourable conditions. We model the complexity of shapes and texture and the diversity of the lichen patterns by instantiating three dimensional lichens shapes organized in a lichen atlas. We have also proposed some tools to model realistic lichen models after images.

Figure 14: *Green algae colonizing a stone statue*

Figure 15: *A complex scene with different types of lichens colonizing the bark of a living tree, dead fallen branches and rocks.*

A key feature of our system is its flexibility and control. Given an input ecosystem description, the designer can either rely on lighting and water flow simulations to automatically compute the regions where lichen will seed and develop, or directly control lichen growth areas by specific tools. A vast variety of lichens patterns and shapes can be synthesized by controlling a few parameters. New kinds of lichens may be incrementally added in the lichen atlas to generate complex scenes.

In a near future, we plan to further investigate the creation of some specific lichens such as Alectoria-Sarmentosa (Figure 2) that hang down in long strands and that cannot be modelled with our approach. We are also working on lichens interacting with the environment in an information feedback loop. In our implementation, only the light and water affect the lichens: in reality the lichens should reciprocally affect the environment and change its parameters locally.

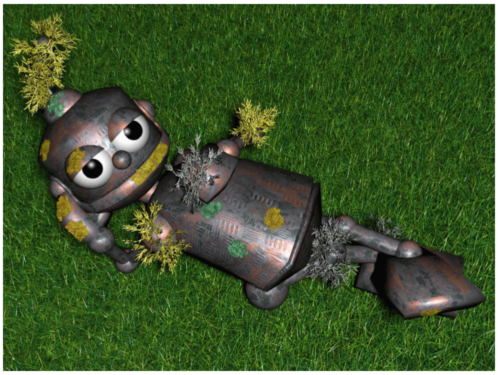

Figure 16: *The Eurographics'2004 mascot abandoned in grass for years and colonized by lichens.*

Acknowledgements

The authors would like to thank Sylvia Sharnoff and Stephen Sharnoff for graciously providing the images of real lichens in this paper. The authors also thank Laurence Boissieux (INRIA) for providing the 3D model of Hugo.

References

[BSS01] BRODO I., SHARNOFF S., SHARNOFF S.: *Lichens of North America*. 2001. ISBN 0-300-08249-5.

[CDM*02] CUTLER B., DORSEY J., MCMILLAN L., MŬLLER M., JAGNOW R.: A procedural approach to authoring solid models. In *Proceedings of SIGGRAPH* (2002), pp. 302–311.

[CKS98] CAMPAGNA S., KOBBELT L., SEIDEL H.-P.: Directed Edges - A Scalable Representation for Triangle Meshes. *Journal of Graphics Tools 3*, 4 (1998), 1–12.

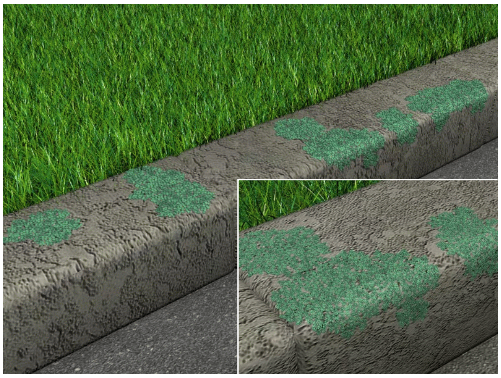

Figure 17: *Lichens developing over a concrete sidewalk border.*

[CS00] CHANG Y.-X., SHIH Z.-C.: Physically-based patination for underground objects. *Computer Graphics Forum 19*, 3 (2000), 109–117.

[CS03] CHANG Y.-X., SHIH Z.-C.: The synthesis of rust in seawater. *The Visual Computer 19*, 1 (2003), 50–66.

[DEL*99] DORSEY J., EDELMAN A., LEGAKIS J., JENSEN H. W., PEDERSEN H. K.: Modeling and rendering of weathered stone. In *Proceedings of SIGGRAPH* (1999), pp. 225–234.

[DH96] DORSEY J., HANRAHAN P. M.: Modeling and rendering of metallic patinas. In *Proceedings of SIGGRAPH* (1996), pp. 387–396.

[DHL*98] DEUSSEN O., HANRAHAN P. M., LINTERMANN B., MĚCH R., PHARR M., PRUSINKIEWICZ P.: Realistic modeling and rendering of plant ecosystems. In *Proceedings of SIGGRAPH* (1998), pp. 275–286.

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

[EPD01] E. PAQUETTE P. P., DRETTAKIS G.: Surface aging by impacts. In *Graphics Interface* (2001).

[FLCB95] FLEISCHER K., LAIDLAW D., CURRIN B., BARR A. H.: Cellular texture generation. In *Proceedings of SIGGRAPH* (1995), pp. 239–248.

[FPB92] FOWLER D. R., PRUSINKIEWICZ P., BATTJES J.: A collision-based model of spiral phyllotaxis. In *Proceedings of the 19th annual conference on Computer graphics and interactive techniques* (1992), ACM Press, pp. 361–368.

[HTK98] HIROTA K., TANOUE Y., KANEKO T.: Generation of crack patterns with a physical model. *The Visual Computer 14*, 3 (1998), 126–137.

[KH01] KELLER A., HEIDRICH W.: Interleaved sampling. In *Rendering Techniques : 12th Eurographics Workshop on Rendering* (2001), pp. 269–276.

[Lin68] LINDENMAYER A.: Mathematical models for cellular interaction in development, parts i and ii. *Journal of Theoretical Biology 18* (1968), 280–315.

[LP02] LANE B., PRUSINKIEWICZ P.: Generating spatial distributions for multilevel models of plant communities. In *Proceedings of Graphics Interface* (2002), pp. 69–80.

[MDG01] MERILLOU S., DISCHLER J.-M., GHAZANFARPOUR D.: Corrosion: Simulating and rendering. In *Graphics Interface* (2001), pp. 167–174.

[Mil94] MILLER G.: Efficient algorithms for local and global accessibility shading. In *Proceedings of SIGGRAPH* (1994), pp. 319–326.

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

[OH99] O'BRIEN J., HODGINS J.: Graphical modeling and animation of brittle fracture. In *Proceedings of SIGGRAPH* (1999), pp. 137–146.

[PJM94] PRUSINKIEWICZ P., JAMES M., MĚCH R.: Synthetic topiary. In *Proceedings of SIGGRAPH* (1994), pp. 351–358.

[SSBD03] SOLER C., SILLION F., BLAISE F., DEREFFYE P.: An efficient instantiation algorithm for simulating radiant energy transfer in plant models. *ACM Transactions On Graphics 22*, 2 (April 2003).

[Sum] SUMNER R. W.: Pattern formation in lichen. Ph.D. thesis, *Département of Electrical Engineering and Computer Science*, Massachusetts institute of technology, 2001.

[WNH97] WONG T.-T., NG W.-Y., HENG P.-A.: A geometry dependent texture generation framework for simulating surface imperfections. In *Eurographics Rendering Workshop* (1997), pp. 139–150.

[WS81] WITTEN T., SANDER L.: Diffusion-limited aggregation, a kinetic critical phenomenon. *Physical Review Letters 47* (1981), 1400–1403.