Expressive Illumination of Foliage Based on Implicit Surfaces

Thomas Luft Michael Balzer
University of Konstanz, Germany

Abstract

This paper presents an approach for vivid representations of foliage based on implicit surfaces. It approximates the complex lighting interaction within the foliage and enables a clear illustration of the general shape and the local density of the foliage, thus supporting the three-dimensional depth cue of the viewer. Due to its straightforward implementation as a preprocessing step that only adjusts the normal vectors of the geometry, this method has no additional memory requirements during the rendering process, and is especially applicable to real-time visualizations. Furthermore, it may be applied to photorealistic as well as non-photorealistic visualizations, which both benefit from the modified normal vector information.

(a) standard local illumination

(b) expressive illumination based on implicit surfaces

Figure 1: The characteristics of foliage are not recognizable by using standard local illumination. In contrast, the approach presented in this paper accentuates the general structure and the local density of the foliage, which results in vivid and expressive representations that are applicable to the real-time rendering of large scenes.

1 Introduction

While for most real-time applications in computer graphics it is sufficient to use simple models of plants and trees based on billboards, the demand for highly realistic and detailed representations steadily increased over the last years. This is reasoned by the availability of sophisticated modeling techniques [6], and of graphics hardware that is able to render even large plant and tree populations in real-time [3,5,8]. Areas of application are visualization tasks in landscaping, architecture, and ecosystem simulations.

Realistic plant models are characterized by a high complexity with a large number of vertices that is far beyond the average of actual models in computer graphics. For example, a simple scrub in an existing plant model library [11] consists of approximately 30 000 vertices, and a tree often consists of more than 200 000 vertices. This high complexity is especially caused by the necessarily accurate modeling of the foliage, whereby a significant simplification of the fine branches and leaves would result in visual artifacts.

These delicate and homogeneous foliage structures necessitate a sophisticated rendering with regard to their lighting and shading. By simply using standard local illumination, information about the general shape and the local density of the foliage is lost and the foliage appears as a 'leaf cloud' without any three-dimensional cue. The qualitative optimal solution is to use global illumination models to obtain vivid representations. The disadvantages of global models are their computational expensiveness, which disqualifies them for real-time calculations, and their additional memory requirements and non-interactivity if they are applied as a preprocessing step. Especially the additional memory requirements are a pitfall for the representation of large scenes with a high number of different plant models that are essential for realistic landscaping and ecosystem visualizations.

This paper presents an approach for vivid representations of foliage based on implicit surfaces. It approximates the complex lighting interaction within the foliage and enables a clear illustration of the general shape and the local density of the foliage, thus supporting the three-dimensional depth cue of the viewer. Due to its straightforward implementation as a preprocessing step that only adjusts the normal vectors of the geometry, this method has no additional memory requirements during the rendering process, and is especially applicable to real-time visualizations. Furthermore, it may be applied to photorealistic as well as non-photorealistic visualizations, which both benefit from the modified normal vector information.

Section 2 discusses related work, explains the concept of implicit surfaces, and clarifies the used standard local illumination model. Section 3 introduces the approach of foliage illumination based on implicit surfaces. Section 4 presents results for the application to photorealistic and non-photorealistic rendering, and Section 5 concludes the paper with a discussion of the presented approach.

2 Background

2.1 Related Work

A method that resembles global illumination effects for real-time rendering is Precomputed Radiance Transfer [21]. It computes soft shadows and interreflections of objects in low-frequency lighting environments and represents them using low-order spherical harmonics. Another method that enhances the display of spatially complex scenes by

providing additional contrast is Ambient Occlusion [2, 18]. This method determines the percentage of the hemisphere above an object point that is not occluded by other parts of the object. The difference between the two methods is that [18] is implemented as a preprocessing step, and the results are stored in textures which are evaluated in the fragment shader, whereas [2] computes the ambient occlusion directly in the fragment shader.

A much more sophisticated parametric model for rendering plant leaves that achieves excellent visual results, is introduced in [22]. Here leaves are described in terms of spatially-variant bidirectional reflectance and transmittance functions that are extracted from real leaves. For the final illumination computation, this method extents Precomputed Radiance Transfer for additionally handling high-frequency sunlight. The disadvantage of this method is its computational expensiveness, and more importantly, the enormous additional memory requirements, which disqualifies it for the real-time rendering of large scenes.

A dedicated approach for the real-time rendering of trees and other plants has been presented in [12]. In this work an approximation of ambient occlusion is computed by using simple spherical or ellipsoidal occluders that are evaluated at runtime. However, the presented results are unconvincing, since especially the general shape of the foliage is not adequately reflected.

A very early work considering the illumination of trees is presented in [19]. This rendering approach is based on particle systems that are shaded by considering the particle position within the tree boundary. The authors use different schemes to approximate the shadowing of the particles: a light-independent scheme for the ambient, and a light-dependent scheme for the diffuse lighting term.

In addition to photorealistic rendering, illumination information is also important for non-photorealistic rendering. Especially the representation of homogeneous leaf structures benefits from an adapted illumination that reduces the visual complexity of the representations and allows more abstract rendering styles. [7, 14] are discussing the abstract rendering of vegetation and similar objects, thereby focussing on the simplification and abstraction of the complex geometry, while preserving details to emphasize structure and lighting. An approach that abstracts complex geometry by additionally using implicit surfaces is proposed in [16]. Here implicit surfaces are used for a rigid simplification as a modeling step thereby enabling a clear representation of the three-dimensional structure of the objects.

2.2 Illumination Model

For the real-time rendering we assume a standard local illumination model [9] that computes the local illumination I at a point p by

$$I = I_a k_a + \sum_i I_i S_i \left[k_d(\vec{N} \cdot \vec{L}) + k_s(\vec{V} \cdot \vec{R})^n \right]. \tag{1}$$

The ambient reflection is denoted by the term $I_a k_a$ with I_a as the intensity of the ambient light, assumed to be constant for all objects, and k_a as the ambient reflection coefficient. Furthermore, for each light source i the diffuse and specular reflection is added, which is multiplied with the intensity I_i of that particular light source and the corresponding shadowing term S_i. Thereby k_d and k_s specify the diffuse and specular reflection coefficient, $\vec{N}$ denotes the normal vector, $\vec{L}$ the vector to the light source i, $\vec{V}$ the vector to the view point, and $\vec{R}$ the reflection vector of the light source i.

2.3 Implicit Surfaces

Following the Metaball concept presented in [17,23], an *implicit surface s* is described by a set of generator points P, whereby each generator point $p_i \in P$ has a radius of influence r_i. The influence of a single generator point p_i at a point q is described by a density function $\mathscr{D}_i(q)$ defined as

$$\mathscr{D}_i(q) = \begin{cases} \left(1 - \left(\frac{\|q-p_i\|}{r_i}\right)^2\right)^2, & \text{if } \|q-p_i\| < r_i \\ 0, & \text{if } \|q-p_i\| \geq r_i \end{cases} . \tag{2}$$

The summation of the density function for all generator points forms the density field $\mathscr{F}$ as

$$\mathscr{F}(q) = \sum_i \mathscr{D}_i(q) - \tau \tag{3}$$

with $\tau \geq 0$. Thus, the implicit surface s with $\mathscr{F}(q) = 0$ is defined as those points q where the sum of the density values of all generators equals the threshold τ. Note that any vector norm can be used for the distance computation in Equation 2. For all examples in this paper, the Euclidian norm is used. Figure 2 shows an implicit surface defined in $\mathbb{R}^2$ by two generator points and a threshold of $\tau = 0.3$.

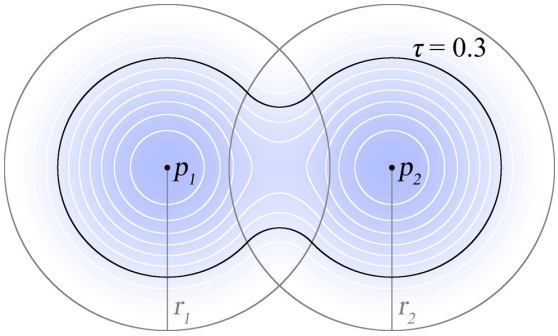


Figure 2: An implicit surface defined in $\mathbb{R}^2$ by two generator points p_1 and p_2 with radii of influence r_1 and r_2 and a threshold of $\tau = 0.3$.

The general shape of an implicit surface can be influenced by the global threshold τ in Equation 3. Greater values describe a closer modulation of the implicit surface to its generator set. Values of $\tau > 1$ result in surfaces that do not necessarily enclose the generator set.

The extraction of implicit surfaces is performed by a Marching Cube algorithm [15]. The result is a polyline in 2D or a triangle mesh in 3D, which approximates the implicit surface at a regular grid with a user specified resolution. Especially for 3D, an additional mesh optimization [13] may be applied to the resulting surface approximation to obtain a homogeneous mesh with a lower number of primitives and more consistent vertex normals. By choosing appropriate parameters for this re-meshing, the changes of the surface topology are not noticeable to the user.

3 Contribution

The main concern when visualizing complex plant models, either as photorealistic or non-photorealistic representations, is the rendering of the foliage. Sophisticated global illumination models are appropriate for off-line rendering, whereas real-time rendering demands fast approximations that does not have to evaluate object interactions at runtime. The issue of simply applying standard local illumination is the loss of the foliage characteristics given by its general shape and local density. This is caused by the fine and complex structure of branches and leaves that is perceived as a noisy 'leaf cloud' due to the lack of light interaction between the foliage components.

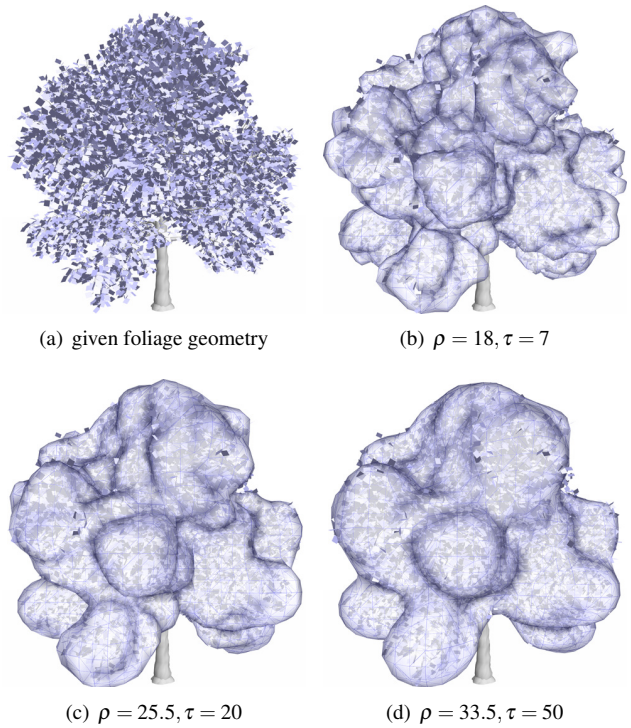
An appealing and vivid rendering of foliage requires a clear representation of its general shape and local density. Implicit surfaces offer a powerful abstraction for representing amorphous object sets. They extract the basic shape of a given object set based on the density distribution of the contained primitives. The here presented approach utilizes implicit surfaces in two different ways to adjust the illumination of the foliage. Firstly, it modifies the ambient term of the used local illumination model to approximate ambient occlusion effects by directly evaluating the density field formed by the generator set. The result are darkened parts within the foliage. Secondly, it realigns the normal vectors of the foliage vertices based on the implicit surface. Here the result is an illumination that emphasizes the general shape of the foliage and reduces the lighting-induced noise within the foliage renderings. Both methods are implemented as a preprocessing step, and the storage of the results within the normal vector information of the geometry enables an efficient real-time rendering without additional memory requirements.

Section 3.1 explains the generation of implicit surfaces from foliage geometry and its parameters. These implicit surfaces are utilized to extract density information for the ambient reflection, which is described in Section 3.2, and to realigns the normal vectors of the vertices for diffuse and specular reflection as described in Section 3.3.

3.1 Implicit Surface Generation

The set of generator points P of the implicit surface s is derived from the set of primitives of the given foliage geometry G that consists of points, triangles, quads or other shapes. Therefore, a generator point p_i is added to P for each geometric primitive $g_i \in G$. The spatial position of p_i is given by the center of mass of g_i. The radius of influence r_i of the generator point p_i is defined by a user specified global parameter ρ, so that each generator point has the same radius of influence. Figure 3 illustrates this implicit surface adaptation, whereby 3(a) shows the tree with its given foliage geometry, and Figures 3(b) to 3(d) show corresponding implicit surfaces with varying degrees of modulation.

The shape of the implicit surface can be adjusted by the global parameters ρ and τ. The parameter ρ defines the radius of influence r_i of each generator point $p_i \in P$ of the implicit surface s. This controls the degree of abstraction of the foliage, whereby small values generate more details within the implicit surface, and high values amplify more the overall shape of the foliage. The second parameter τ defines the threshold for the implicit surface s within the density field formed by the generator set P as described in Section 2.3. It influences the distance of the implicit surface to the generator points, causing a closer modulation. Experience proved that the interactive specification of these parameters is necessary due to the broad variety of plant models. Values that provide adequate results can be automatically specified with regard to the size of the



(a) given foliage geometry (b) $\rho = 18, \tau = 7$

(c) $\rho = 25.5, \tau = 20$ (d) $\rho = 33.5, \tau = 50$

Figure 3: Simplification and abstraction of a given foliage geometry by implicit surfaces with varying degrees of modulation specified by the parameters ρ and τ.

model, but only user-driven fine tuning offers the best results.

Although sophisticated methods exist for the evaluation of the density field and the generation of the implicit surface, these computations are very time-consuming. Due to the homogeneous distribution of the leaves, this issue can be resolved by randomly selecting just a fraction of the geometric primitives of the foliage as generator points.

3.2 Density-based Ambient Reflection

By examining foliage in nature, a gradual darkening from the outer to the inner parts of the foliage is observed. This ambient occlusion effect can be approximated for real-time rendering by utilizing an implicit function that is derived from the foliage model. Therefore, the density field of the corresponding set of generator points is directly evaluated. High values within this density field indicate dark areas of the foliage, whereas low density values indicate areas that are near the boundary of the foliage with more incident light. This method is also applicable to other parts of the geometry that is covered by foliage geometry, e.g. the trunk and the branches of the tree.

Having a set of generator points P, which is derived from the foliage geometry G and forms the density field $\mathscr{F}$, an ambient reflection coefficient k_{a_v} is generated for each vertex $v \in G$ by determining the density value $d_v = \mathscr{F}(v)$, and mapping d_v to

$k_{a_v} \in [0..1]$ by the following transfer function:

$$k_{a_v} = \begin{cases} 1, & \text{if } d_v \leq 0 \\ k_{a_{min}} + \left(1 - k_{a_{min}}\right)\left(1 - \frac{d_v}{d_{min}}\right)^n, & \text{if } 0 < d_v < d_{min} \\ k_{a_{min}}, & \text{if } d_v \geq d_{min} \end{cases} \tag{4}$$

Thereby $k_{a_{min}} \geq 0$ defines the minimum ambient reflection coefficient that is used for all density values above a chosen lower bound d_{min}. These areas characterize the core of the foliage where no variant ambient lighting is observed. For density values between zero and d_{min}, the exponent n influences the gradient from maximum to minimum ambient reflection. Values of $n < 1$ result in a slower darkening from the outer to the inner parts of the foliage, whereas values of $n > 1$ generate a more rapid darkening. The influence of these three parameters that allow for an accurate adjustment of the ambient reflection is illustrated in Figure 4. An example for a rendering using only the with this method computed ambient reflection coefficients is given in Figure 5(a).

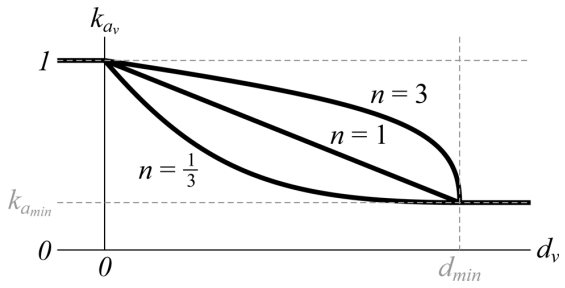


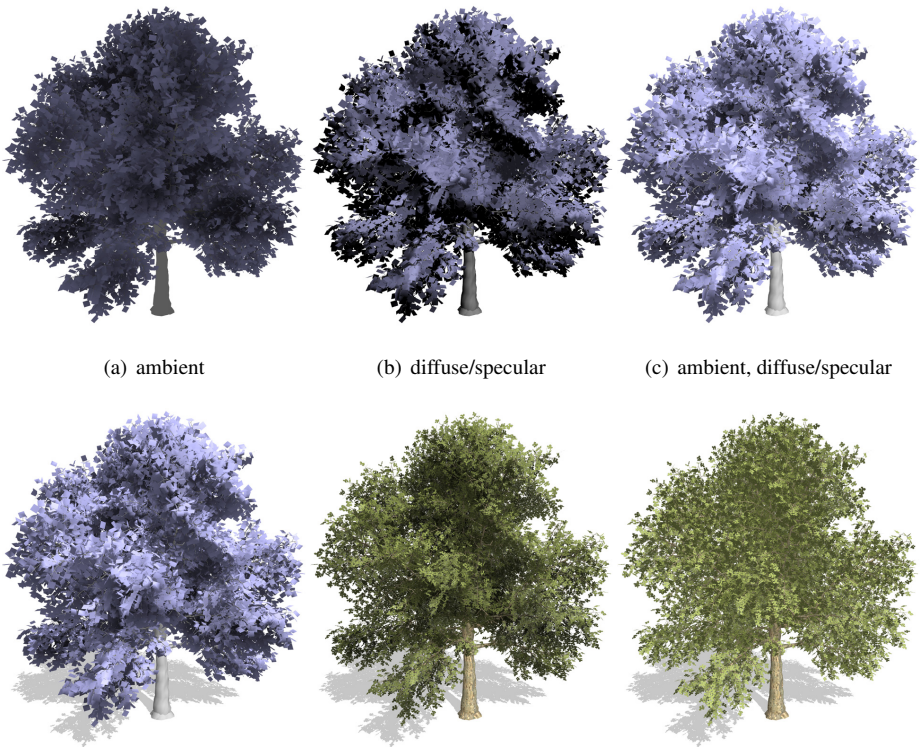
Figure 4: Dependency of the ambient reflection coefficient k_{a_v} on the density value d_v according to Equation 4 for the three different exponents $n = \frac{1}{3}$, $n = 1$, and $n = 3$.

At the end of the preprocessing step, the ambient reflection coefficients are encoded within the normal vector information of the vertices. Therefore, the normal vector of each vertex is scaled by the corresponding ambient reflection coefficient, which enables an effective interpolation and has no additional memory requirements. At runtime, a simple vertex shader program decodes the normal vector to extract the ambient reflection coefficient, and re-scales the normal vector for further reflection computations, e.g. the diffuse and specular reflection term in Equation 1.

3.3 Normal Vector Realignment for Diffuse/Specular Reflection

To accentuate the general shape of the foliage it is necessary to reduce the noisiness of the diffuse and specular reflection. According to the standard local illumination as given in Equation 1, the computation of the diffuse and specular reflection is based on the normal vector $\vec{N}$. The random orientation of the leaves in the foliage causes also a random alignment of the corresponding normal vectors that is responsible for the noisy reflection. Thus, it is necessary to realign the normal vectors of the foliage vertices to reduce this effect.

A naive method is to realign the normal vectors according to a bounding ellipsoid. The drawback is the disregarding of the general shape of the foliage that is determined by its main branches. A much better method to approximate the shape of the foliage



(a) ambient (b) diffuse/specular (c) ambient, diffuse/specular

(d) ambient, diffuse/specular, shadow (e) ambient, diffuse/specular, shadow, textures (f) standard local illumination model

Figure 5: Illumination components for the photorealistic rendering of foliage with the here presented method based on implicit surfaces. For comparison, the same model has been rendered with the standard local illumination model.

is to utilize implicit surfaces as already shown in Figure 3. Here the simplified shape is perfectly modulated to the general shape of the foliage, whereby the degree of this modulation can easily be adjusted.

The realignment of the normal vectors is applied for each vertex v of the foliage geometry. Therefore, the vertex v_s of the implicit surface is determined that has the minimal Euclidian distance to v. The new normal vector $\vec{N}'$ of v is then set to the normal vector $\vec{N}_s$ of v_s. Alternatively, $\vec{N}'$ is computed as a linear combination of the original normal vector $\vec{N}$ of v and $\vec{N}_s$. For all examples in this paper $\vec{N} = \vec{N}_s$ is used. The result of this normal vector realignment for the diffuse and specular reflection is presented in Figure 5(b).

4 Results

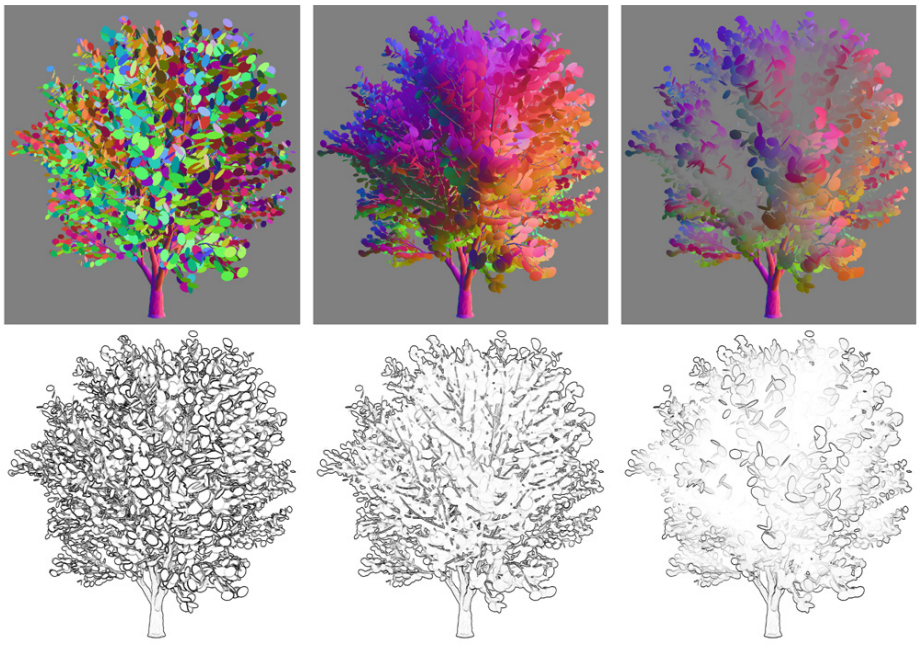
Photorealistic Rendering: Having approximated ambient occlusion information and realigned normal vectors for each vertex, these two effects are now combined (Figure 5(c)) and additionally enriched with shadow (Figure 5(d)) and texture (Figure 5(e)) information. The benefit of this implicit surface based method becomes evident when

comparing the final result with a standard local illumination (Figure 5(f)). The general shape and the local density of the foliage is clearly accentuated, and the rendering is more vivid and realistic. Another direct comparison between a photo of a real tree and a similar tree model is given in Figure 6. Again, the tree model is rendered using standard local illumination and the implicit surface based method. More examples for photorealistic renderings of single plants and also a complex scene are given in Figure 8.

(a) standard (b) photo (c) presented method

Figure 6: Comparison between the standard local illumination, a photo of a similar tree, and the in this paper presented method based on implicit surfaces.

Non-Photorealistic Rendering: The in this paper presented method is also adaptable as an abstraction mechanism for non-photorealistic rendering, and especially rendering techniques that rely on normal vector information benefit directly. For example, image space filters that detect discontinuities in the normal buffer in order to produce line drawings, e.g. [20], fail for complex botanical objects, if they are applied to the original normal vectors. Again, the reason is the spatially high signal frequency contained in the normal buffer resulting in cluttered artifacts as shown in Figure 7(a). In contrast, in Figure 7(b) these image space filters achieve simplified results by using the realigned normal vectors. Furthermore, the extracted ambient occlusion that is coded in the normal vector length suppresses interior details of the foliage, which in turn results in a reproduction of the characteristic features of the foliage topology with a reduced complexity as shown in Figure 7(c). In comparison to existing works that consider line drawing simplification [1, 10, 24], the advantage of the here presented method is its implementation as a preprocessing step, and the fact that it requires no additional memory and computation time during rendering. Other non-photorealistic rendering techniques that aim at producing an abstract shading, such as cartoon shading [4] or watercolor representations [16], also benefit from the method presented in this paper. A smooth and clear shading is achieved, which faithfully resembles the abstraction process that is usually performed by an artist. This allows the direct use of plant models that have been created for photorealistic rendering without an additional modeling effort. More examples for the application of this technique in the field of non-photorealistic rendering are presented in Figure 8 thereby using the watercolor approach presented in [16].



(a) standard (b) realigned (c) realigned & scaled

Figure 7: Application to non-photorealistic rendering. Top row: Normal buffer; normal vectors are mapped to RGB color space. Bottom row: Edge detection filter applied to the normal map.

5 Discussion

This paper presents an approach for the expressive illumination of foliage that is applicable to photorealistic as well as non-photorealistic representations. The usage of implicit surfaces for the abstraction and simplification of the foliage geometry enables the modification of the normal vector information of the vertices. Therefore, the darkening of the inner parts of the foliage is approximated by the density field of the implicit function, and the general shape of the foliage is accentuated by realigning the normal vectors to the implicit surface. The degree of abstraction of an implicit surface can be influenced directly by two parameters that describe its modulation to the given foliage geometry. Due to the variety of the individual plant models, it is difficult to find general settings. Therefore, these two parameters have to be interactively adjusted by the user.

The implementation as a preprocessing step and the encoding of the results within the normal vector information of the vertices allow for an efficient storage, interpolation, and rendering without additional memory requirements. The corresponding, very simple, decoding method for the normal vectors can be implemented in a vertex shader thereby producing only a minor computational overhead during rendering. These attributes are advantageous for the application of the presented method to real-time visualizations of large-scale ecosystems.

The approximation of ambient occlusion effects is not restricted to the vertices of the foliage, rather it also allows to adequately modify branches or other objects that are covered by the foliage. Furthermore, it is possible to compute the implicit surface for a

plant population, which in turn allows to incorporate static object-to-object interactions between the individuals. An exemplary result is a small tree that is naturally darkened by a nearby large tree due to the ambient occlusion effects. Nevertheless, such a scenario cannot be combined with instancing—a common technique to present large tree populations—since the normal vector modification of the foliage is thereby computed individually for each tree.

Admittedly, the implementation as a preprocessing step does not permit dynamic object to object interactions. Although dynamic scenes are possible using Precomputed Radiance Transfer [21], this is not adequate for complex scenes due to the high computational effort and memory requirements. Likewise, a correct global illumination simulation is not applicable to scenes with thousands or even millions of plant models because of the tremendous computational effort. In comparison to ambient occlusion [18], the here presented method is an effective and easy to implement approximation that is especially appropriate to amorphous object sets.

References

[1] P. Barla, J. Thollot, and F. Sillion. Geometric clustering for line drawing simplification. In *Proceedings of the Eurographics Symposium on Rendering*, pages 183–192, 2005.

[2] M. Bunnell. Dynamic ambient occlusion and indirect lighting. In *GPU Gems 2*, pages 223–233. Addison-Wesley, 2005.

[3] C. Colditz, L. Coconu, O. Deussen, and H.-C. Hege. Real-time rendering of complex photorealistic landscapes using hybrid level-of-detail approaches. In *Proceedings of the 6th International Conference for Information Technologies in Landscape Architecture*, pages 97–106. Herbert-Wichmann Verlag, 2005.

[4] P. Decaudin. Cartoon looking rendering of 3D scenes. Research Report 2919, INRIA, jun 1996.

[5] O. Deussen, P. Hanrahan, B. Lintermann, R. Měch, M. Pharr, and P. Prusinkiewicz. Realistic modeling and rendering of plant ecosystems. In *Proceedings of ACM SIGGRAPH*, pages 275–286. ACM Press, 1998.

[6] O. Deussen and B. Lintermann. *Digital Design of Nature: Computer Generated Plants and Organics*. Springer-Verlag, 2005.

[7] O. Deussen and T. Strothotte. Computer-generated pen-and-ink illustration of trees. In *Proceedings of ACM SIGGRAPH*, pages 13–18. ACM Press, 2000.

[8] A. Dietrich, C. Colditz, O. Deussen, and P. Slusallek. Realistic and interactive visualization of high-density plant ecosystems. In *Proceedings of the Eurographics Workshop on Natural Phenomena*, pages 73–81, 2005.

[9] J. D. Foley, A. van Dam, S. K. Feiner, and J. F. Hughes. *Computer Graphics: Principles and Practice in C*. Addison Wesley, 1997.

[10] S. Grabli, F. Durand, and F. Sillion. Density measure for line-drawing simplification. In *Proceedings of Pacific Graphics*, pages 309–318, 2004.

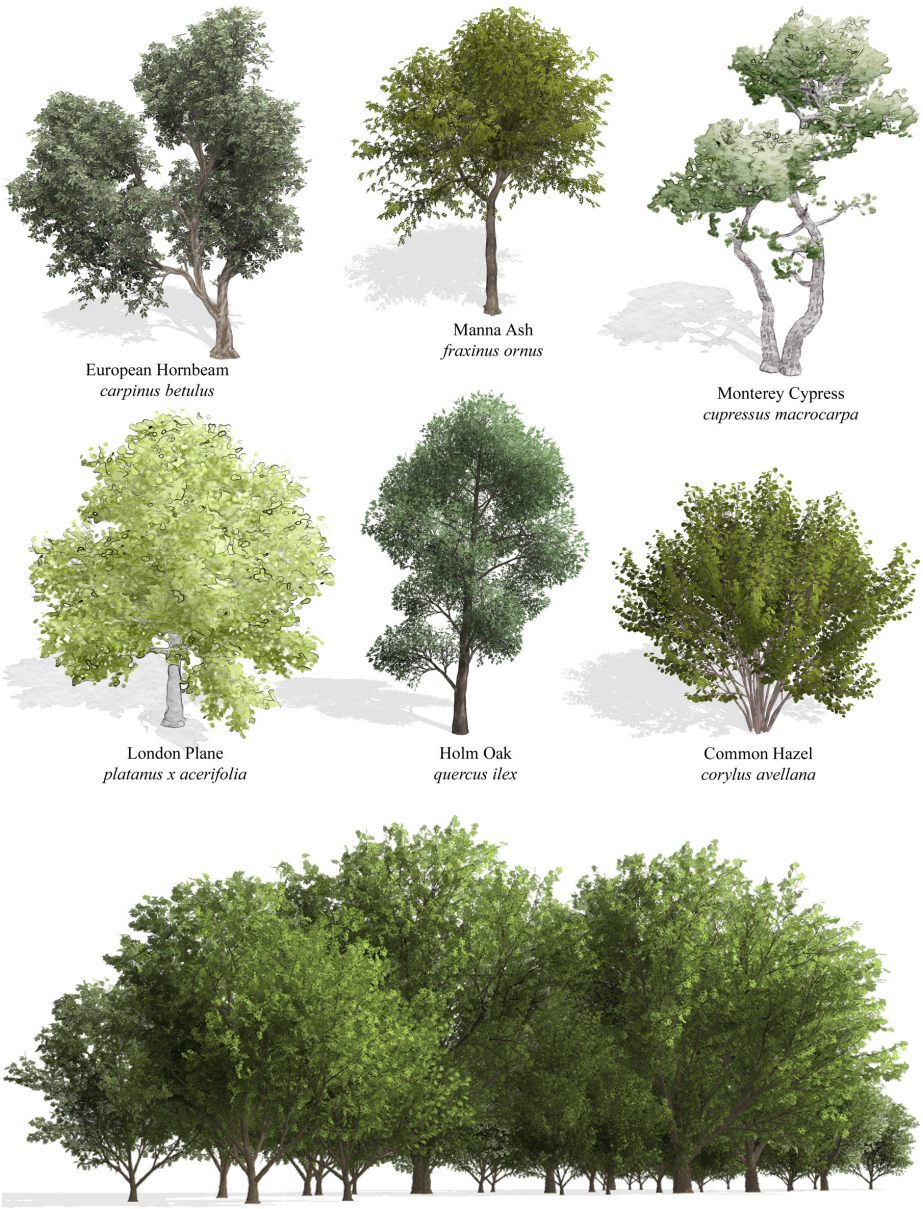


Figure 8: Photorealistic and non-photorealistic representations of single plant models and of a complex scene.

[11] Greenworks Organic Software. Homepage of the Xfrog modeling software. http://www.greenworks.de, 2006.

[12] K. Hegeman, S. Premože, M. Ashikhmin, and G. Drettakis. Approximate ambient occlusion for trees. In *Proceedings of the Symposium on Interactive 3D Graphics and Games*, pages 87–92. ACM Press, 2006.

[13] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Mesh optimization. In *Proceedings of ACM SIGGRAPH*, pages 19–26. ACM Press, 1993.

[14] M. A. Kowalski, L. Markosian, J. D. Northrup, L. Bourdev, R. Barzel, L. S. Holden, and J. Hughes. Art-based rendering of fur, grass, and trees. In *Proceedings of ACM SIGGRAPH*, pages 433–438. ACM Press, 1999.

[15] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3D surface construction algorithm. In *Proceedings of ACM SIGGRAPH*, pages 163–169. ACM Press, 1987.

[16] T. Luft and O. Deussen. Real-time watercolor illustrations of plants using a blurred depth test. In *Proceedings of NPAR*, 2006. to appear.

[17] S. Murakami and H. Ichihara. On a 3D display method by metaball technique. *Transactions of the Institute of Electronics, Information and Communication Engineers*, J70-D(8):1607–1615, 1987. In Japanese.

[18] M. Pharr and S. Green. Ambient occlusion. In *GPU Gems*, pages 279–292. Addison-Wesley, 2004.

[19] W. Reeves and R. Blau. Approximate and probabilistic algorithms for shading and rendering structured particle systems. In *Proceedings of the Conference on Computer Graphics and Interactive Techniques*, pages 312–322, 1985.

[20] T. Saito and T. Takahashi. Comprehensible rendering of 3-D shapes. In *Proceedings of ACM SIGGRAPH*, pages 197–206, 1990.

[21] P.-P. Sloan, J. Kautz, and J. Snyder. Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. In *Proceedings of ACM SIGGRAPH*, pages 527–536. ACM Press, 2002.

[22] L. Wang, W. Wang, J. Dorsey, X. Yang, B. Guo, and H.-Y. Shum. Real-time rendering of plant leaves. *ACM Transactions on Graphics*, 24(3):712–719, 2005.

[23] A. Watt. *3D Computer Graphics*. Addison-Wesley, 2000.

[24] B. Wilson and K.-L. Ma. Rendering complexity in computer-generated pen-and-ink illustrations. In *Proceedings of NPAR*, pages 129–137, 2004.

Figure 9: Direct comparison of the standard local illumination (left) and the illumination based on implicit surfaces (right) of the European Hornbeam tree model applied to a photorealistic representation.

Figure 10: Direct comparison of the standard local illumination (left) and the illumination based on implicit surfaces (right) of the Manna Ash tree model applied to a photorealistic representation.

Figure 11: Direct comparison of the standard local illumination (left) and the illumination based on implicit surfaces (right) of the Monterey Cypress tree model applied to a non-photorealistic representation.

Figure 12: Direct comparison of the standard local illumination (left) and the illumination based on implicit surfaces (right) of the London Plane tree model applied to a non-photorealistic representation.

Figure 13: Direct comparison of the standard local illumination (left) and the illumination based on implicit surfaces (right) of the Holm Oak tree model applied to a photorealistic representation.

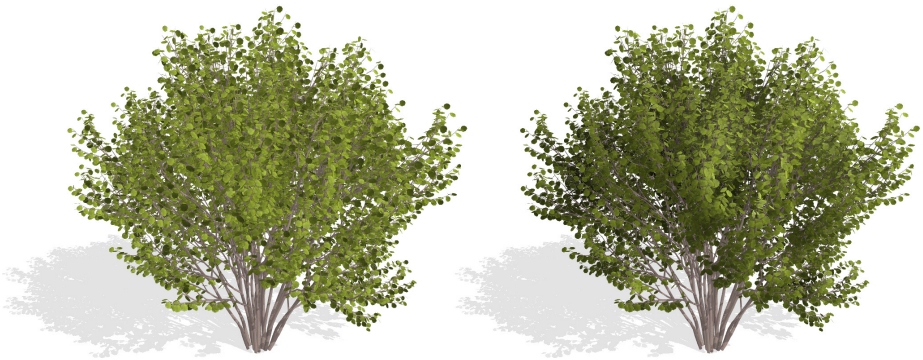


Figure 14: Direct comparison of the standard local illumination (left) and the illumination based on implicit surfaces (right) of the Common Hazel plant model applied to a photorealistic representation.

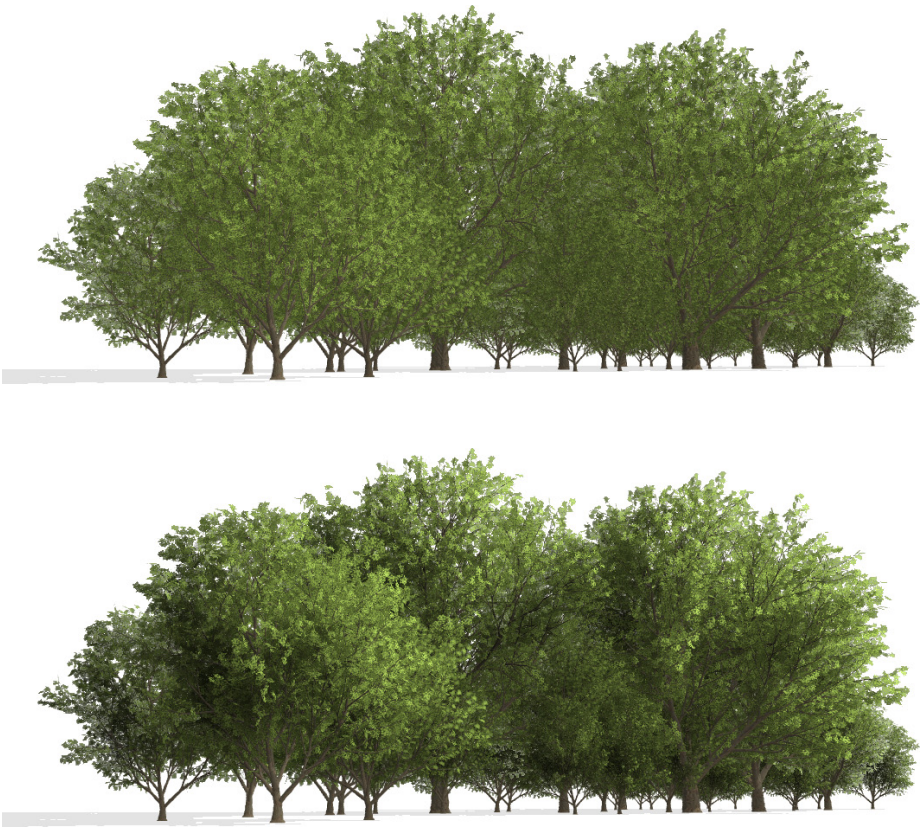


Figure 15: Direct comparison of the standard local illumination (top) and the illumination based on implicit surfaces (bottom) of a complex scene of tree models applied to a photorealistic representation.