The Journal of the Society for Art and Science Vol. 10, No. 2, pp. 48 – 57

Abstract Shape Extraction For Watercolor Tree Rendering Using Anisotropic Density Functions

Michio Shiraishi **Yu Kamimura** **Mikio Shinya**

Toho University Graduate School of Science

Abstract

Trees are important in synthesizing outdoor scenes. One of the features of trees is their complicated shape composed of many leaves, branches and trunks. In many non-photorealistic rendering methods, abstract shapes are required to convey the information of shape to the viewer.

Luft and Deussen proposed a method that extracts abstract shapes from tree polygon models, which adopted isotropic density functions for leaves. The leaves are generally flat, however, so that their method tends to extract the abstract shape with dilation. This paper proposes an abstract shape extraction method that utilizes anisotropic density functions to capture anisotropic structures of leaves.

The proposed method was implemented and the several types of tree models were tested. This paper also introduces metrics to evaluate dilation and erosion of extracted shapes. The result shows that the dilation and erosion can be controlled by choosing the parameters, so that users can extract abstract shapes in several styles according to their preference.

1 Introduction

Plants, especially trees, are very familiar to us and they have been playing important roles in many computer-synthesized scenes. One of the features of trees is their complicated shape composed of many leaves, branches and trunks.

In general, abstraction is a key component in non-photorealistic rendering (NPR) [1], and thus, it is very important to allow users to control abstract shapes to reflect their preference. Abstraction can be performed in image-space [2, 3] as well as object-space, but shape abstraction in object-space is more advantageous for animation applications to preserve temporal coherence. Shape abstraction methods are often specialized to a ceratin domain of object types and rendering styles. Kawalski proposed a stroke-based method that procedurally reproduces complexities on simple object surfaces using graftals for illustrative tree rendering [4]. Noble proposed a NURBS-based method cartoon hair animation [5], and Mehra proposed a method that abstracts artificial objects.

Luft and Deussen [7, 8] proposed a method that extracts abstract shapes from tree polygon models. They adopted isotropic functions that only depend on the distance from the leaf vertices to the point. An abstract shape is extracted as implicit surfaces on which the sum of the density functions equals to a given threshold. This method works well in those regions where leaves are located sparsely. However, the extracted implicit surfaces tend to be too bloated in case that many leaves are localized. This happens because the method fails to capture the flat shape of leaves due to their use of the isotropic density function.

When users try to reduce bloat in dense regions, satisfactory shapes in sparse regions may shrink. Therefore, better user controls are required in shape abstraction for watercolor tree rendering.

This paper proposes an abstract shape extraction method that utilizes anisotropic density functions to capture anisotropic structures of leaves. The proposed method modulates function parameters to make the density function more isotropic in sparse regions because it is known that the isotropic functions give successful result in those regions. On the other hand, in regions where the leaves are placed densely, the method employs functions with more anisotropic features in order to retain the shapes of leaves. In this way, the proposed method provides better controls in shape abstraction for tree watercolor rendering. The extracted shapes can be also used for photorealistic rendering, as mentiond in [9].

The anisotropic density function is defined in a local coordinate system centered at the vertex on a leaf. The vertex normal direction is used to define the first axis, and, the other two axes are defined to span the tangent plane at the vertex. The function is controlled by three parameters along each axis, so that the distribution of the density function becomes flatter like a shape of leaf. Our method controls the parameters according to the leaf density in such a way that the extracted implicit surface could preserve the shapes of leaves where the leaves are located densely. Users can also manually adjust the parameters to control the abstract shape according to their preference, ranging from the sparse look to the close-packed one.

The proposed method was implemented and several types of tree models were tested. It was found that the extracted abstract shapes were favorable in most cases: the resulting abstract shape did not have too bloated region or too thin region. It was also demonstrated that users can extract an abstract shape of a tree from the various styles including a skinny and dilated look with the proposed method.

The remainder of this paper is organized as follows. Section 2 describes the previous work by Luft and Deussen. Subsequently, we describe our method in Section 3, and Section 4 describes how to control the abstract shape by the parameters. Some NPR results of the abstract shape are shown in Section 5. Finally, Section 6 concludes the paper.

2 Previous Method

Luft and Deussen [8] utilized the isotropic density functions D_j, which only depend on the distance from the vertex c_j on a leaf for extracting an abstract shape. The density value $D_j(p)$ at the point p is defined as the following equation.

$$D_j(p) = \begin{cases} \left(1 - \left(\frac{\|p - c_j\|}{R_1}\right)^2\right)^2 & (\|p - c_j\| \leq R_1) \\ 0 & (\|p - c_j\| > R_1) \end{cases} \tag{1}$$

This equation shows that the density value becomes zero when the distance from the vertex to the point is larger than its influencing bound R_1.

The implicit function $N^{(iso)}(p)$ is defined as the sum of these density functions:

$$N^{(iso)}(p) = \sum_j D_j(p) - T, \tag{2}$$

where T is a threshold. The abstract shape of a tree is determined as the implicit surface where the value of the implicit function becomes zero. Figure 1 shows an example of the implicit surface when two density functions are placed side by side.

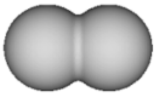
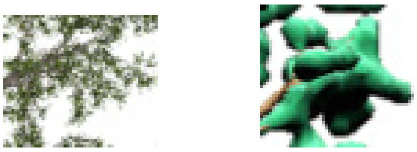


Figure 1: Example of the implicit surface generated by two density functions.

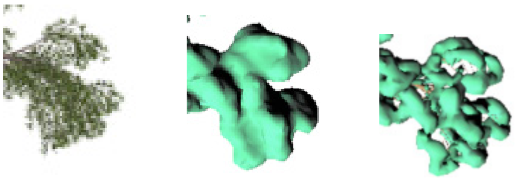


(a) Geometry (b) By Luft and Deussen

Figure 2: The abstract shape in the low leaf density region.

We have implemented the method and applied to several models of trees. Figure 2(a) shows a part of the models, and the extracted abstract shape is shown in Figure 2(b). This result suggests that the method can well extract the abstract shape in the low leaf density region. However, when the method is applied to another part of the model shown in Figure 3(a), the abstract shape is bloated as shown in Figure 3(b). As shown in the example, the method tends to generate bloated shapes in the low leaf density region. The major reason is that the isotropic density functions fail in capturing *flat* shapes of leaves.

In addition to this *leaf level* anisotropicity, there is the *branch level* anisotropicity, which is the anisotropicity in a sense that the leaves are distributed along with branches. We consider that the previous method reflects the branch level anisotropicity because it is automatically captured by density functions are summed over all leaves. Therefore, we focus on the leaf level anisotropicity. If we ignore the fact that a leaf is not spheral, dilation is found as shown in Figure 3. In this paper, the term *anisotropicity* refers to the leaf level characteristic of leaves.



(a) Geometry (b) Isotropic (c) Anisotropic

Figure 3: The abstract shape where the high leaf density region. (a) Original geometry. (b) Extracted by isotropic density functions. (c) Extracted by anisotropic density functions

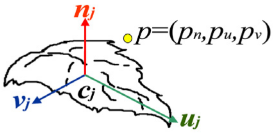
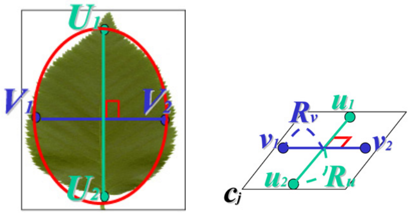


Figure 4: Local coordinate system on a leaf.



(a) Texture (b) Leaf mesh

Figure 5: The influencing bounds of R_u and R_v.

3 Proposed Method

3.1 Anisotropic Density Function

We propose a method that deals with the directivity of leaf shapes by introducing anisotropic density functions. Our anisotropic density function is defined in a local coordinate system whose origin is the leaf vertex c_j as shown in Figure 4. Normal direction n_j at the vertex c_j is used as one of the axes, and the other two axes, u_j and v_j, are defined on the leaf surface. The density function $D_j^{(a)}(p)$ is defined in the local coordinate (p_n, p_u, p_v) as follows:

$$E(p) = \left(\frac{p_n}{R_n}\right)^2 + \left(\frac{p_u}{R_u}\right)^2 + \left(\frac{p_v}{R_v}\right)^2, \quad (3)$$

$$D_j^{(a)}(p) = \begin{cases} (1 - E(p))^2 & (E(p) < 1) \\ 0 & (E(p) \geq 1) \end{cases}. \quad (4)$$

In these equations, the parameters R_n, R_u and R_v represent the influencing bounds in n_j, u_j and v_j axes, respectively. R_u, R_v, u_j axis and v_j have to be determined based on the leaf shape. The isosurface of a single anisotropic density function derives an ellipse on a leaf plane. So we approximate a leaf shape by an ellipse as shown in Figure 5(a). The major axis of the ellipse is used as u_j axis, the minor axis as v_j axis, the major diameter as R_u and the minor diameter as R_v.

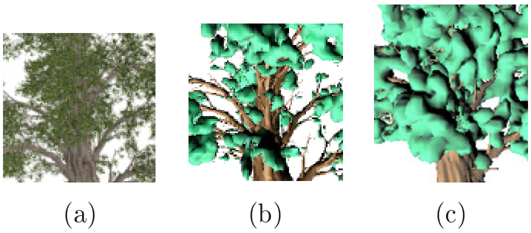


Figure 6: Result of anisotropicity control. (a) Original geometry. (b) Extracted by constant anisotropicity. (c) Extracted by controlled anisotropicity.

In our implementation, approximated ellipses are set in textures that are mapped to the leaf meshes. The major and minor diameters are defined on the texture plane. The method calculates the points U_1 and U_2 that correspond to the major diameter and V_1 and V_2 that correspond to the minor diameter. By mapping the texture, the point u_1, u_2, v_1 and v_2 are calculated as shown in Figure 5(b). The u_j axis is given by $u_1 - u_2$ and the v_j axis by $v_1 - v_2$. The influencing bounds are defined as the following equations:

$$\begin{aligned} R_u &= \|u_1 - u_2\|/2, \quad (5) \\ R_v &= \|v_1 - v_2\|/2. \quad (6) \end{aligned}$$

If u_j axis and v_j axis are not orthogonal, they have to be orthogonalized, for example, by the Gram-Schmidt orthonormalization.

Since the leaf shape is flat, it is not natural that the influencing bound R_n of normal direction is set larger than R_u or R_v. Therefore, R_n is defined by parameter λ_R as:

$$R_n = \lambda_R \times R_v, \quad (7)$$

for$0 < \lambda_R \leq 1$.

3.2 Anisotropicity Control Based On Leaf Density

The anisotropic density functions are applied to the geometry shown in Figure 3(a). As a result, the bloated shapes are eased in the high leaf density region as shown in Figure 3(c). However, when the anisotropic density functions are applied to the model shown in Figure 6(a), the extracted shape gives poor result as shown in Figure 6(b), where the method fails to capture the shape in the central region. This suggests that naive introduction of the anisotropic density functions causes artifacts that the extracted shape is not captured well in the low leaf density region.

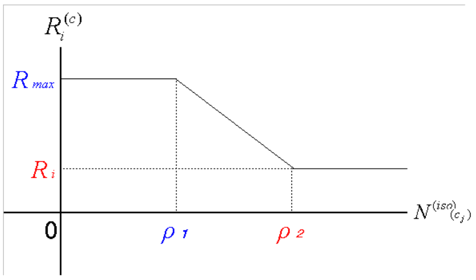


Figure 7: Controlling $R_i^{(c)}$ by the leaf density.

In order to solve this problem, we introduce the method that controls the anisotropicity in such a way that it becomes more anisotropic in the high leaf density region and it becomes more isotropic in the low leaf density region. The anisotropicity is represented by the influencing bounds

$R_u^{(c)}$, $R_v^{(c)}$ and $R_n^{(c)}$ which correspond to the bounds for u, v and n axes, respectively. We use $R_i^{(c)}$ to denote one of the three influencing bounds in the following explanation.

$R_i^{(c)}$ is calculated for $i \in \{u, v, n\}$ as follows:

$$R_i^{(c)} = \begin{cases} R_{max} & (N^{(iso)}(c_j) \leq \rho_1) \\ R_{max} + \dfrac{(R_i - R_{max})(N^{(iso)}(c_j) - \rho_1)}{\rho_2 - \rho_1} & \\ & (\rho_1 < (N^{(iso)}(c_j) < \rho_2) \\ R_i & (N^{(iso)}(c_j) \geq \rho_2). \end{cases} \quad (8)$$

It means that $R_i^{(c)}$ is controlled by the value of implicit function at the origin of the local coordinate system c_j, $N^{(iso)}(c_j)$, is computed by Equations (2) as illustrated in Figure 7. In order to make the density function isotropic in the low leaf density region, the influencing bound is set to R_{max} when $N^{(iso)}(c_j)$ is less than a threshold ρ_1. On the other hand, in order to make the density function anisotropic in the high leaf density region, the influencing bound is calculated by Equations (5), (6) and (7). The influencing bound between ρ_1 and ρ_2 is calculated by the linear interpolation. ρ_1 and ρ_2 are calculated by the mean m and the variance σ^2 of the implicit function values at leaf vertices, v_i $(1 \leq i \leq N_v)$:

$$\begin{aligned} m &= \sum_i N^{(iso)}(v_i)/N_v \\ \sigma^2 &= \sum_i (N^{(iso)}(v_i) - m)^2/N_v \\ \rho_1 &= m + \alpha_1 \sigma^2 \quad (9) \\ \rho_2 &= m + \alpha_2 \sigma^2, \quad (10) \end{aligned}$$

In our current implementation, α_1 is set to 0 and α_2 is set to 2. Figure 6(c) shows the results from the controlled anisotropic density functions.

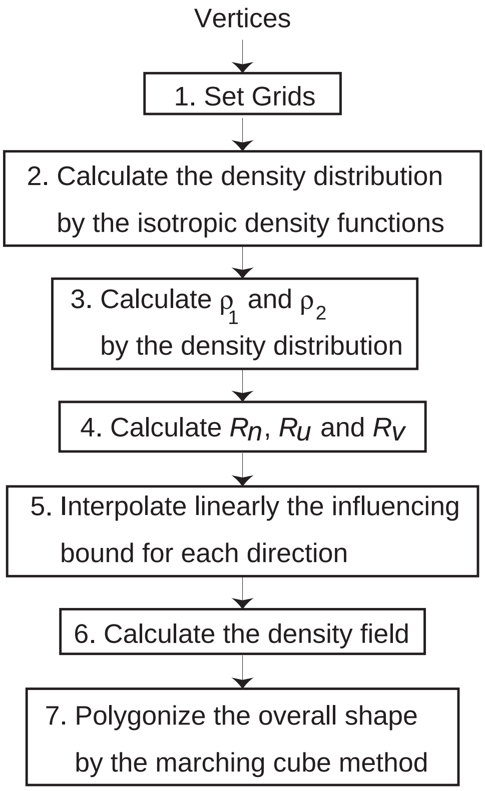

Vertices
1. Set Grids
2. Calculate the density distribution by the isotropic density functions
3. Calculate ρ1 and ρ2 by the density distribution
4. Calculate Rn, Ru and Rv
5. Interpolate linearly the influencing bound for each direction
6. Calculate the density field
7. Polygonize the overall shape by the marching cube method

Figure 8: Algorithm.

Compared with Figure 6(b), the neutralization of erosion at the central region is observed.

3.3 Algorithm

Figure 8 shows the procedure of abstract shape extraction by the proposed method. In step 1, the method calculates the bounding box of the tree model and generates three-dimensional grids. In step 2, the method calculates $N^{(iso)}(p_i)$ at the cell vertex p_i. Then, by interpolating $N^{(iso)}(p_i)$, the method calculates $N^{(iso)}(c_j)$ that is the value of the implicit function at the leaf vertex c_j. In step 3, the method calculates the mean and variance of the implicit function at the leaf vertices, and then, determines ρ_1 and ρ_2 by Equations (9) and (10). In step 4, R_n, R_u and R_v at each vertex c_j are calculated by Equations (5), (6) and (7). In step 5, the influencing bounds $R_i^{(c)}$ of each vertex c_j are computed using Equation (8). In step 6, the method evaluates the value of the implicit function at the cell vertex p_j using the anisotropic density function $D_j^{(a)}$ in the influencing bounds $R_i^{(c)}$. Finally, the implicit function is polygonized by the marching cube method [10].

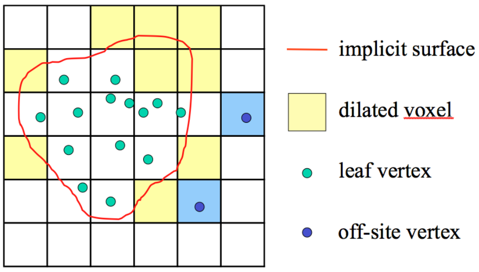


Figure 9: Dilated voxels and off-site vertices.

4 Experiments

In this section, we evaluate how well the proposed method can control dilation and erosion of extracted shapes. We introduce two metrics, the number of *dilated voxels* and *off-site vertices*, to quantitatively measure the effects. Figure 9 illustrates these metrics.

Dilated voxel A voxel can be classified into three categories, these are inside/on/outside the implicit surfaces. An extracted shape is organized by the voxels of the former two categories, but some voxels do not contain any leaf vertices. These voxels do not contain the original polygon model, but appears in the resulting extracted shape. We define these voxels as *dilated voxels*. The number of dilated voxels can be considered as a metric to measure the extent of dilation.

Off-site vertex On the contrary, some leaf vertices in the original model are not included in the extracted shape. It is undesirable that locations indicated by the original vertices do not remain in the resulting shape, so they can be regarded as a sign of erosion. We define these vertices as *off-site vertices*. The number of off-site vertices gives a measure to examine how much the shape are shrinked.

4.1 Anisotropicity Control

We extract abstract from the tree model as shown in Figure 10 (52648 vertices) by the three method:

- The method by Luft and Deussen that uses isotropic density functions. (L&D)
- The proposed method that uses anisotropic density functions without the anisotropicity control. (UNCTRLED)
- The proposed that uses anisotropic density functions with the anisotropicity control based on leaf density described in Section 3.2. (CTRLED)

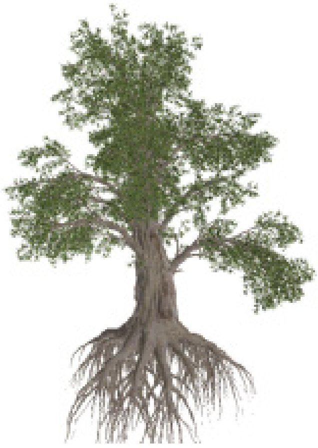


Figure 10: Tree model used in evaluation.

The voxel resolution used in this evaluation is 64 × 64 × 64.

We count the number of dilated voxel and off-site vertex for three extracted shapes. Table 1 shows the result. The number of dilated voxels is the smallest when we use UNCTRLED, but the number of off-site vertices is largest. The number of off-site vertices is smallest when we use L&D, but the number of dilated voxels is largest.

The abstract shape by CTRLED succeeds in damping the number of dilated voxels compared with L&D and the number of off-site vertices compared with UNCTRLED. This demonstrates that the proposed method is capable to control abstraction and to extract intermediate shapes. In the next section, we will discuss more details of the control issues.

4.2 Parameter Effects

The method described in the previous section includes three parameters that users can choose:

- T: a threshold to extract the implicit surface from the density field
- R_{max}: the influencing region of a density function where the leaf density is low
- λ_R: the flatness of the density function

In this section, we discuss the behavior of extracted shape according to the change of the parameters.

Figure 11 shows the variation of the number of dilated voxels and the number of off-site vertices according to the change of the threshold T. As T increases, the number of dilated voxels becomes smaller, but the number of off-site vertices becomes larger. This means that the dilation and erosion of the abstract shape cannot be controlled independently by T. Figure 12 show the effects of changing the threshold T.

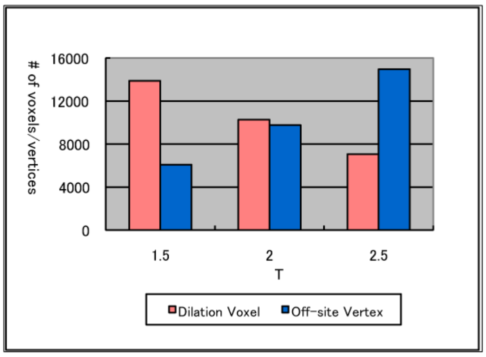


Figure 11: Number of dilated voxels and off-site vertices according to change of Threshold T.

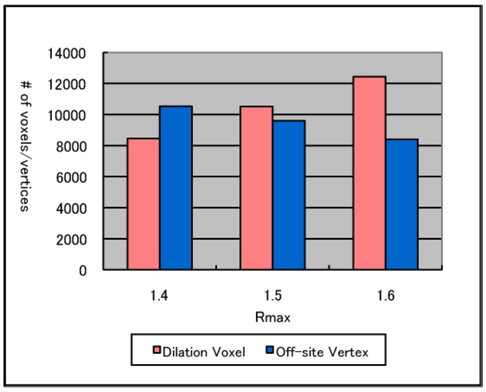


Figure 13: Number of dilated voxels and off-site vertices according to change of Parameter R_{max}.

The parameter R_{max} also influences the extracted shape. Figure 13 shows the same metrics as Figure 11 when R_{max} varies. As R_{max} increases, the change of the number of off-site vertices is repressed compared to Figure 11, but dilated voxels increases. Namely, we can extract the preferable dilated shape by controlling R_{max} while the erosion are eased. If users prefer a fat shape, they can increase R_{max} to get the desired shape as shown in Figure 14.

Figure 15 indicates the variation of the metrics as λ_R changes. As shown in this graph, the number of dilation voxels does not dramatically change by the change of λ_R. However, the number of off-site vertices increases by decreasing λ_R. Therefore, the decrease of λ_R controls the magnitude of erosion. For example, users can make holes where the leaves

Table 1: The number of dilated voxels and the number of off-site vertices.

Density Function	isotropic (L&D)	anisotropic (without anisotropicity control) (UNCTRLED)	anisotropic (with anisotropicity control) (CTRLED)
The number of dilated voxels	13368	2649	10275
The number of off-site vertices	172	24766	9593

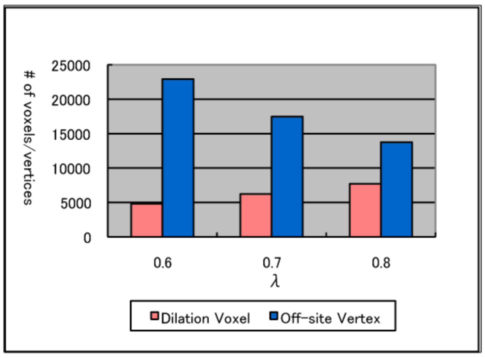


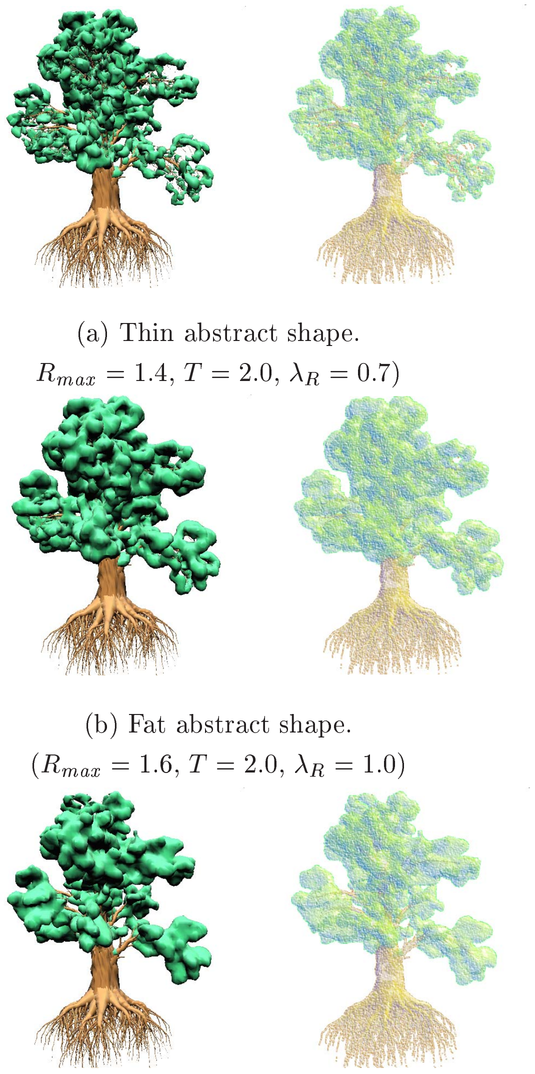
Figure 15: Number of dilated voxels and off-site vertices according to change of Parameter λ_R.

are not so dense if such a shape is desirable. In such cases, they can decrease λ_R, as shown in Figure 16.

5 NPR Results

Some NPR examples are shown in Figure 17. The images on the left are rendered by Phong shading and those on the right are rendered by the water color shading [8]. The abstract shape in Figure 17(a) becomes thin by setting λ_R small. On the other hand, the shape in Figure 17(b) becomes fat by setting R_{max} large. Figure 17(c) is the abstract shape by the previous method for comparison. As shown in these examples, the method can extract the shapes that give different impressions by choosing the parameters.

Table 2 shows the processing time of the previous method and the proposed method. The processing time of the proposed method is measured as the sum of execution time from step 1 to step 7 in Figure 8. The previous method is composed of the step 1, 6 and 7. So the proposed method requires



(a) Thin abstract shape.
$R_{max} = 1.4$, $T = 2.0$, $\lambda_R = 0.7$)

(b) Fat abstract shape.
($R_{max} = 1.6$, $T = 2.0$, $\lambda_R = 1.0$)

(c) The previous method.
($R_i = 1.5$, $T = 1.5$)

Figure 17: Rendering results.

Table 2: Processing time. (CPU: AMD Athlon 64 Processor 3200+ 2.01GHz, Memory: 1.00GB)

Resolution (voxels)	32^3	64^3	128^3
Luft and Deussen (sec.)	1.297	5.164	35.382
The proposed method (sec.)	2.029	7.676	41.208

additional execution time to the previous method, corresponding to the steps 2 through 5.

6 Conclusion

In this paper, we proposed a method that utilizes the anisotropic density functions instead of the isotropic ones used in the previous method. The anisotropicity of the density functions are controlled based on leaf density. This anisotropicity control succeeded in reducing artifacts such as unnatural dilation and erosion observed in the abstract shapes extracted by the previous method. In addition, users' preferences can be reflected in abstract shape extraction by controlling the parameters.

The method cannot extract the abstract shape in real time because the abstract shape is polygonized by the marching cube method. It is considered that the direct rendering of the implicit surface[11] accelerates the algorithm by obviating the polygonization process.

References

[1] Hertzmann, A. 2010. Non-Photorealistic Rendering and the Science of Art. Proceedings of the Third International Symposium on Non-Photorealistic Animation and Rendering (NPAR).

[2] Hertzmann, A. 1998. Painterly rendering with curved brush strokes of multiple sizes. In Proceedings of SIGGRAPH 98. pp. 453-460.

[3] Shiraishi, M., Yamaguchi, Y. 2000. An algorithm for automatic paiterly rendering based on local source image approximation, In NPAR 2000, pp. 53-58.

[4] Kowalski, M. A., Markosian, L., Northurp, J. D., Bourdev, L., Loring, R., B., Hughes, J., F. 1999. In Proceedings of SIGGRAPH 99, pp. 433-438.

[5] Noble, P., Tang, W. 2004. Modelling and Animating Cartoon Hair with NURBS Surface, Proceedings of the Computer Graphics International 2004, pp. 60-67.

[6] Mehra, R., Zhou, Q., Long, J., Sheffer, A., Gooch A., J. Mitra, N., J. 2009. Abstraction of Man-Made Shapes. ACM Transactions on Graphics, Vol. 28, No. 5.

[7] Luft, T., Deussen, O. 2005. Interactive watercolor animations. In PG '05: Poster Proceedings of the 13th Pacific Conference on Computer Graphics and Applications, 7-9.

[8] Luft, T., Deussen, O. 2006. Real-time watercolor illustrations of plants using a blurred depth test. In NPAR '06: Proceedings of the 4th international symposium on Non-photorealistic animation and rendering, 11-20.

[9] Luft, T., Balzer, M., Deussen, O. 2007. Expressive illumination of foliage based on implicit surfaces. In Proceedings of Eurographics Workshop on Natural Phenomena (EGWNP). pp. 71-78.

[10] Lorensen, W. E., AND Cline, H. E. 1987. Marching cubes: A high resolution 3d surface construction algorithm. SIGGRAPH Computer Graphics 21, 4, 163-169.

[11] Kanamori, Y., "GPU-based Fast Rendering of Metaballs," The 5th Korea-Japan Joint Workshop on Computer Graphics, Oct, 2007.

[12] Shiraishi, M., Kamimura, Y. AND Shinya, M., 2010, Abstract Shape Extraction For Non-photorealistic Tree Rendering Using Anisotropic Density Functions, In NICOGRAPH International 2010.

Michio Shiraishi

Michio Shiraishi received the PhD degree in arts and sciences from the University of Tokyo in 2003. He is a lecturer in the Department of Information Sciences, Toho University. His current research interest includes non-photorealistic rendering and perception of illustrations. He is a member of

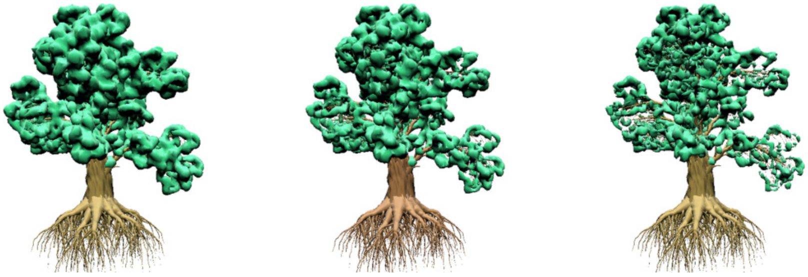
ACM, IEEE Computer Society, Information Processing Society of Japan, Society for Art and Science, etc.

Yu Kamimura

Yu Kamimura received the BS degree in science from Toho University in 2009. He is a graduate student in Toho University Graduate School of Science. His research interest is computer graphics.

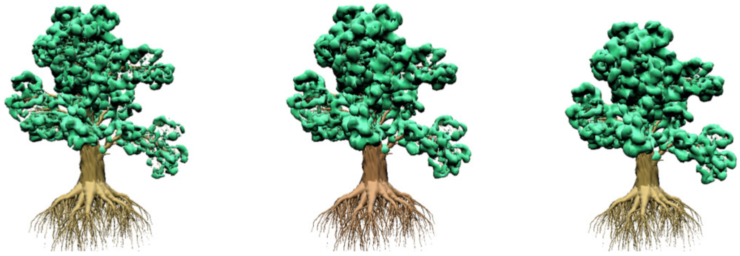
Mikio Shinya

Mikio Shinya is currently a Professor at Department of Information Science, Toho University. He received a BSc in 1979, an MS in 1981, and a PhD in 1990 from Waseda University. He joined NTT Laboratories in 1981, and moved to Toho University in 2001. He was a visiting scientist at the University of Toronto in 1988-1989. His research interests include computer graphics and visual science.



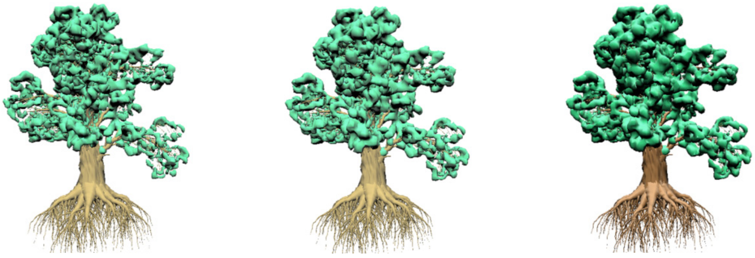
(a) $T = 1.5$ (b) $T = 2.0$ (c) $T = 2.5$

Figure 12: The effects of changing the threshold T.



(a) $R_{max} = 1.4$ (b) $R_{max} = 1.5$ (c) $R_{max} = 1.6$

Figure 14: The effects of changing the parameter R_{max}.



(a) $\lambda_R = 0.6$ (b) $\lambda_R = 0.7$ (c) $\lambda_R = 0.8$

Figure 16: The effects of changing the parameter λ_R.