www.cadanda.com
Computer-Aided Design
AND APPLICATIONS

Design for Manufacturing of IFS Fractals from the Perspective of Barnsley's Fern-leaf

A. M. M. Sharif Ullah[1], Yoshimi Sato[2], Akihiko Kubo[3] and Jun'ichi Tamaki[4]

[1]Kitami Institute of Technology, ullah@mail.kitami-it.ac.jp
[2]Kitami Institute of Technology, m1452100156@std.kitami-it.ac.jp
[3]Kitami Institute of Technology, kuboak@mail.kitami-it.ac.jp
[4]Kitami Institute of Technology, tamaju@mail.kitami-it.ac.jp

ABSTRACT

This study deals with the design for manufacturing (DFM) of fractals created by a random walk called iterated function system (IFS). In particular, the DFM of an IFS-created fractal called Barnsley's fern-leaf is considered. The IFS dedicated for creating virtual models of a fern-leaf uses a set of four strictly-contracting affine mappings in the onto manner. The interactions among these mappings are studied in detail in order to identify some data structures. Based on the identified data structures, a DFM procedure is proposed. In the proposed DFM procedure, three out of the four mappings are employed in both the onto and one-to-one manner. The proposed DFM procedure is applied to the redesign of the shape (fern-leaf). Physical models of the redesigned fern-leaf are manufactured using both additive and subtractive manufacturing technologies (3-D printing and milling). The factors affecting accuracy of the physical models are also described. Although this study is limited to the shape of the fern-leaf, other IFS-created shapes can be redesigned using the proposed DFM procedure. Nevertheless, this study sheds some light on our understanding of how to develop more accurate physical models of IFS-created fractals.

Keywords: iterated function system, fractals, design for manufacturing, CAD/CAM, additive manufacturing.

1. INTRODUCTION

Mandelbrot brought a mathematical entity called fractal into our attention in the 1960s as a means to modeling complex shapes in a realistic manner [16–18]. Formally, a set whose Hausdorff-Besicovitch dimension exceeds its topological dimension is called fractal, i.e., a set having a non-integer Hausdorff-Besicovitch dimension is called fractal [11]. Hausdorff-Besicovitch dimension is popularly known as fractal or power-law dimension. See [8, 18, 32] for more details on the Hausdorff-Besicovitch dimension and its derivatives. Generally, a self-similar (strictly speaking, self-affine) shape under all scales of magnification or dilution is considered a fractal [16, 18], and a simple set of rules is good enough for creating the shape, even though it looks very irregular and fragmented. Therefore, creating a fractal simply means creating a set of points (point-cloud) that exhibits a non-integer Hausdorff-Besicovitch dimension and, at the same time, models an irregular and fragmented shape that can be subdivided in parts wherein each part is a reduced copy of the whole. In order to create such a point-cloud a mathematical entity called Iterated Function System (hereafter referred to as IFS) has been introduced [1, 2, 6]. An IFS executes a recursive mapping process and creates a point-cloud. In each iteration, it randomly selects one out of a predefined set of strictly-contracting affine mappings, and the occurrences of the mappings are controlled by a set of predefined probabilities or weights. See Section 3 for more details on how an IFS operates and creates a point-cloud.

However, one of the remarkable features of IFS is that it helps model natural and aesthetically valuable objects (e.g., leaf, tree, landscape, jewelry, and architectural artifact) in a realistic manner [2]. For this reason, it has earned a great deal of attention from the computer-aided design and manufacturing community. Even though an IFS can create an accurate virtual model of an shape, it does not necessarily

© 2014 CAD Solutions, LLC, http://www.cadanda.com
Taylor & Francis
Taylor & Francis Group

mean that a manufacturing technology can accurately manufacture it. In order to manufacture an accurate physical model of an IFS-created fractal, an appropriate procedure for design for manufacturing (DFM) is needed because it considers the needs of the underlying manufacturing technology while designing a shape and makes necessary design changes [13]. The objective of this article is to get some insights into the DFM of IFS-created fractals. In particular, the DFM of a special IFS-created fractal used for modeling fern-leaf introduced by Barnsley [2] will be described. The proposed DFM procedure emphasizes such aspects as the nature of applying the mappings (onto or one-to-one mapping), selection of the important mappings, and sequencing of the mappings. The ultimate goal is to controlling the distribution of the points in the point-cloud, i.e., in the virtual model of the underlying shape.

The remainder of this article is organized as follows. Section 2 briefly describes some relevant research works that deal with the design and manufacturing of fractal objects and with the utilization of the concept of fractal. Section 3 describes the formal settings of an IFS and its application in modeling such natural objects as tree and fern-leaf. Section 4 describes the proposed DFM procedure. Section 5 describes some useful results obtained while building physical models of a fern-leaf using the proposed DFM procedure with the aid of both subtractive manufacturing (i.e., milling) and additive manufacturing (i.e., 3-D printing). Section 6 provides the concluding remarks of this study.

2. RELATED WORKS

Like mathematicians, engineers have been researching fractals since its reincarnation in the 1960s by Mandelbrot. Among others, such issues as CAD modeling of complex and aesthetically valuable objects using fractals, visual presentation of fractals, manufacturability of the shapes modeled by fractals, systems engineering based on the concept of fractal (i.e., self-similarity), quantification of the complexity using self-similarity (or fractal) dimension, and creation of the random walks using fractals have been investigated by numerous authors. Some of the selected works are briefly described below.

Numerous authors have studied the issue of complex shape modeling using fractals. For example, Pang and Hui [24] have developed a methodology to generate numerous variants of visually appealing shapes by integrating IFS with genetic algorithms. Rochman and Vázquez [26] have developed a methodology based on the fractal geometry to model complex shapes for rapid prototyping. Wannarumon [40] has developed an expert system to automate the process of generating art forms of jewelry wherein the basic art forms are represented by IFS fractals. Wannarumon et al. [38] and Wannarumon and Bohez [39] have developed an approach that heuristically simulate the parameters of IFS fractals with the aid of genetic algorithms for jewelry shape modeling taking the user-preferences into account. Chan and Hui [4] have developed a method to automate the surface pattern generation process using Lindenmayer-system (an approach that also creates IFS fractals) and genetic algorithms. Sun and Starly [33] have developed a method to generate shapes having scaffolds (internally networked channels) using Lindenmayer-system for bio-CAD and rapid prototyping.

In terms of computer graphics (virtual representation) of fractals, convex-hull creation has gained a great deal of attention. For example, Martyn [20] has developed an algorithm for approximating the rectangular convex hulls of 2D affine IFS fractals, ensuring a given accuracy. Martyn [19] has developed an approach to realistic real-time rendering of 3-D IFS fractals that requires both approximations of convex hulls of IFS fractal subsets and estimations of normals on a fractal surface. Mishkinis et al. [21] have developed an algorithm for approximating the curvilinear convex hulls of 2D IFS fractals preserving the desired accuracy.

In terms of manufacturing of fractals, the following works can be noted. Soo and Yu [31] have developed a data structure called radial-annular tree in order to create tool-paths for CNC machining of some well-known fractals. Soo and Yu [30] have developed a data structure called radial-blossoming tree in order to create the physical models of fractal objects using rapid prototyping (additive manufacturing). Soo et al. [29] have developed the traversal algorithms to extract necessary information from the radial-blossoming tree structure of a fractal object and have shown its usefulness in fabricating the aesthetically valuable objects by layered manufacturing. In order to generate tool-path for layer manufacturing of fractal objects, Chiu et al. [5] have developed a data structure that creates the data of slab-grid pixels for each layer of layered manufacturing. Li et al. [15] have developed a voxel-based data structure to capture the geometric information of a fractal object in order to fabricate its physical model using layered manufacturing. Ullah et al. [36] have reported the manufacturing defects of fractal objects while producing them with the aid of subtractive manufacturing (e.g., milling) and additive manufacturing (e.g., stereolithography). They have suggested the ways for creating defect-free physical models of fractals. Omori et al. [23] have described a general procedure to redesign an IFS fractal for the sake of manufacturing.

The concept of self-similarity (or fractalness) has also gained a great deal of attention. Particularly, such issues as complexity quantification, random walk creation, and systems engineering have used the concept of fractalness. For example, Brown et al. [3] have shown that the complexity of an engineered surface can be quantified using the quantity called fractal dimension [18, 32, 8]. They have also shown that the

© 2014 CAD Solutions, LLC, http://www.cadanda.com

variability in the fractal dimension identifies the periodic components of the surface profiles, and the characters of the surface roughness can be expressed as a function of length-scale that is used to measure the fractal dimension. Ullah et al. [34, 37] have shown that the fractal dimension and roughness of a machined surface are difficult to correlate, i.e., a low fractal dimension may not necessarily mean a low surface roughness. Purintrapiban and Kachitvichyanukul [25] have shown that the fractal dimension can detect patterns in process data of a manufacturing system. In order to analyze the impact of operational sequences on the product quality, Ruschin-Rimini et al. [27] have developed an IFS fractal based methodology for the visual analysis and classification of operational sequences. Mizugochi et al. [22] have developed a methodology for creating the fractal tool-path for polishing the metal molds and shown its effectiveness in improving the surface finish. Griffiths [9] has developed a tool-path creation algorithm based on the space-filling fractal called Hilbert curve [36], which outperforms the conventional tool-path generation process. Kumar et al. [14] have used fractal geometry based raster tool-paths for layered manufacturing of porous objects. Shin et al. [28] have developed an agile and flexible manufacturing system wherein the control architecture is fractal (self-similar) in nature. Fractals are used to design physical devices, too. For example, Hohlfeld and Cohen [10] have shown that the self-similar and origin symmetry fractals help achieve frequency independent performance of antennas that are used in telecommunication.

3. IFS-CREATED FRACTALS

This section describes the formal settings of the IFS-created fractals (i.e., point-clouds on an x-y plane). This section also describes two examples (tree and fern-leaf) of virtual model creation using IFS. The objective is twofold: 1) to define the necessary mathematical operations and 2) to elucidate the general features of the IFS-created point-clouds while representing a shape.

As mentioned earlier, an IFS is a useful means for creating virtual models of a fragmented and irregular shape in terms of point-clouds so that the point-clouds exhibit a non-integer Hausdorff-Besicovitch dimension, i.e., the virtual models or point-clouds become a fractal [1, 6]. Sometimes, the natural and aesthetically meaningful objects (e.g., tree, leaf, landscape, mountain, cloud, jewelry, and architectural artifacts, and alike) can realistically be modeled by IFS-created fractals [2, 7, 18, 29]. However, two main constituents of an IFS are as follows.

- A set of strictly-contracting affine mappings of the form, $x_i = a_jx_{i-1} + b_jy_{i-1} + e_j$, $y_i = c_jx_{i-1} + d_jy_{i-1} + f_j$, $j = 1, \ldots, n$ (n is the number of mapping), $i = 1, \ldots, N$.
- A set of probabilities $\{p_j \mid j = 1, \ldots, n\}$, $p_1 + \ldots + p_j + \ldots + p_n = 1$, to control the occurrences of each mapping. Since $p_1 + \ldots + p_j + \ldots + p_n = 1$, p_j acts as the relative weight of j-th mapping in N iterations.

An IFS starts its operation from a given point $(x_{i=0}, y_{i=0})$ called seed, and ultimately results a point-cloud consisting of $N + 1$ points $\{(x_i, y_i) \mid i = 0, \ldots, N\}$. The points are created recursively by the random application of the affine mappings $x_i = a_jx_{i-1} + b_jy_{i-1} + e_j$, $y_i = c_jx_{i-1} + d_jy_{i-1} + f_j$, $j = 1, \ldots, n$, and preserving, at the same time, the relative weights of the mappings given by their probabilities $\{p_j \mid j = 1, \ldots, n\}$. Thus, an IFS can be defined by the settings in equation (1). As seen from equation (1), to run an IFS one needs to set the seed, number of iterations, the vectors of the parameters and probabilities (i.e., weights) of n affine mappings, i.e., (x_0, y_0), N, $(a_j, b_j, c_j, d_j, e_j, f_j)$, $j = 1, \ldots, n$, and $(p_j \mid j = 1, \ldots, n)$, respectively. The probabilities $(p_j \mid j = 1, \ldots, n)$ are used to calculate the cumulative probabilities, as follows: $cp_j = p_1 + \ldots + p_j$, $j = 1, \ldots, n$. Needless to say, $cp_n = 1$. From the calculated cumulative probabilities, mutually exclusive intervals are calculated, as follows: $w_1 = [0, cp_1), \ldots, w_j = [cp_{j-1}, cp_j), \ldots, w_n = [cp_{n-1}, cp_n]$. This means that the intervals w_j, $j = 1, \ldots, n$, divide the interval $[0,1]$ into n mutually exclusive intervals so that $w_1 + \ldots + w_n = [0,1]$. Therefore, if a random number denoted as r_i, $i = 1, \ldots, N$, in the interval $[0,1]$ belongs to w_j, then the corresponding mapping is used to generate a point (x_i, y_i) from the previous one (x_{i-1}, y_{i-1}) so that $x_i = a_jx_{i-1} + b_jy_{i-1} + e_j +$, $y_i = c_jx_{i-1} + d_jy_{i-1} + f_j$. The process is repeated for all $i = 1, \ldots, N$. It is worth mentioning that to ensure a point-cloud generated by the process defined in equation (1) be a fractal (i.e., a set of points having non-integer Hausdorff-Besicovitch dimension), all the affine mappings in equation (1) must be strictly-contracting ones (i.e., the eigenvalues of the mappings have modulus less than one) [6].

$$\begin{aligned}
&\mathit{Input\ Setting}:\\
&\quad \mathit{Seed}: (x_0, y_0) \qquad \mathit{Iterations}: N\\
&\quad \mathit{Mapping}: (a_j, b_j, c_j, d_j, e_j, f_j) \quad j = 1, \ldots, n\\
&\quad \mathit{Probabilites}: (p_j | j = 1, \ldots, n)\\
&\mathit{Calculation}:\\
&\quad cp_j = p_1 + \ldots + p_j \qquad j = 1, \ldots, n\\
&\quad w_1 = [0, p_1), \ldots, w_j = [cp_{j-1}, cp_j), \ldots, w_n\\
&\qquad = [cp_{n-1}, cp_n)\\
&\mathit{Iteration}:\\
&\quad \mathit{For} \quad i = 1, \ldots, N\\
&\qquad \mathit{random\ number}: r_i \leftarrow [0, 1]
\end{aligned}\tag{1}$$

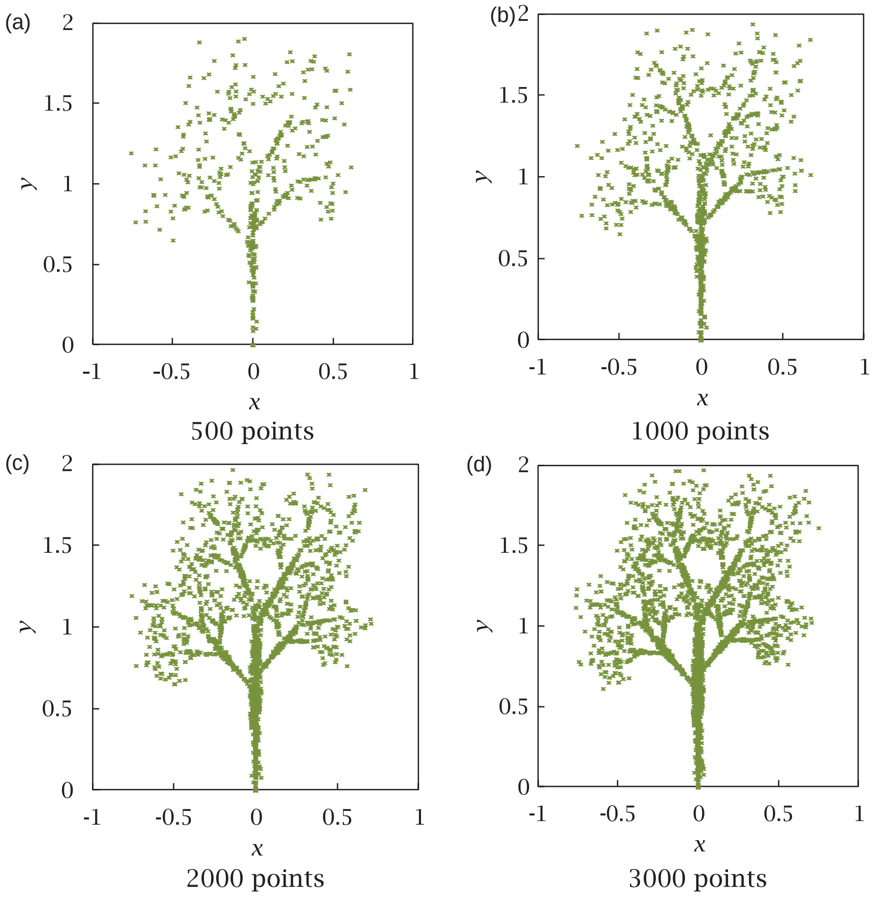
© 2014 CAD Solutions, LLC, http://www.cadanda.com

$$
\begin{aligned}
&\mathit{If}\ r_i = w_1 \quad x_i = a_1 x_{i-1} + b_1 y_{i-1} + e_1\\
&\quad y_i = c_1 x_{i-1} + d_1 y_{i-1} + f_1\\
&\ldots\\
&\mathit{If}\ r_i = w_j \quad x_i = a_j x_{i-1} + b_j y_{i-1} + e_j\\
&\quad y_i = c_j x_{i-1} + d_j y_{i-1} + f_j\\
&\ldots\\
&\mathit{If}\ r_i = w_n \quad x_i = a_n x_{i-1} + b_n y_{i-1} + e_n\\
&\quad y_i = c_n x_{i-1} + d_n y_{i-1} + f_n
\end{aligned}
$$

For example, Tab. 1 lists the inputs (mapping parameters and probabilities) of the IFS that create a point-cloud in accordance with equation (1) representing the shape of a tree. As listed in Tab. 1, six affine mappings are used to create a point-cloud $\{(x_i, y_i) \mid i = 0, \ldots, N\}$ for modeling the shape (tree). The relative weights of the mappings (p_j) are also listed in Tab. 1. Figure 1 shows four models (point-clouds) of the shape (tree). Needless to say, the models are created by applying equation (1) under the conditions listed in Tab. 1 and the seed $(x_{i=0}, y_{i=0})$ is equal to

	j					
	1	2	3	4	5	6
a_j	0.05	0.05	0.459677455	0.46985757	0.432912361	0.421371001
b_j	0	0	– 0.321343358	– 0.15396579	0.275191087	0.257074686
c_j	0	0	0.385612029	0.170979132	– 0.250173715	– 0.353477693
d_j	0.6	– 0.5	0.383064546	0.422841029	0.476203597	0.306451637
e_j	0	0	0	0	0	0
f_j	0	1	0.6	1.1	1	0.7
p_j	0.25	0.05	0.22	0.13	0.13	0.22

Tab. 1: Settings of IFS for creating models of a tree.



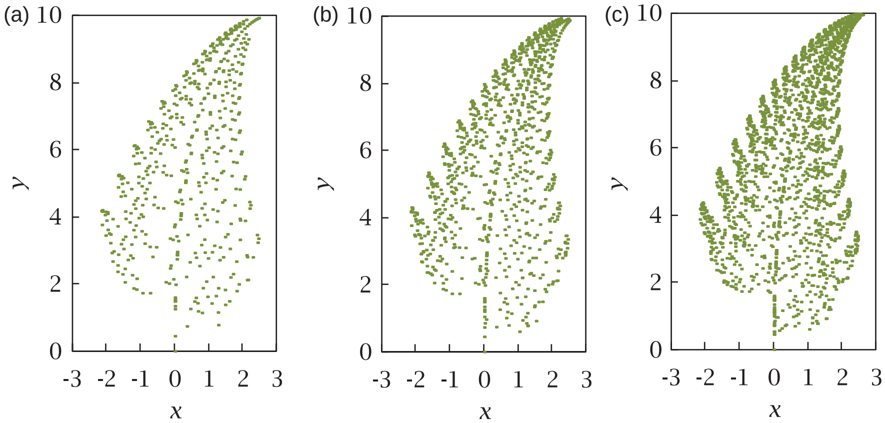
500 points

1000 points

2000 points

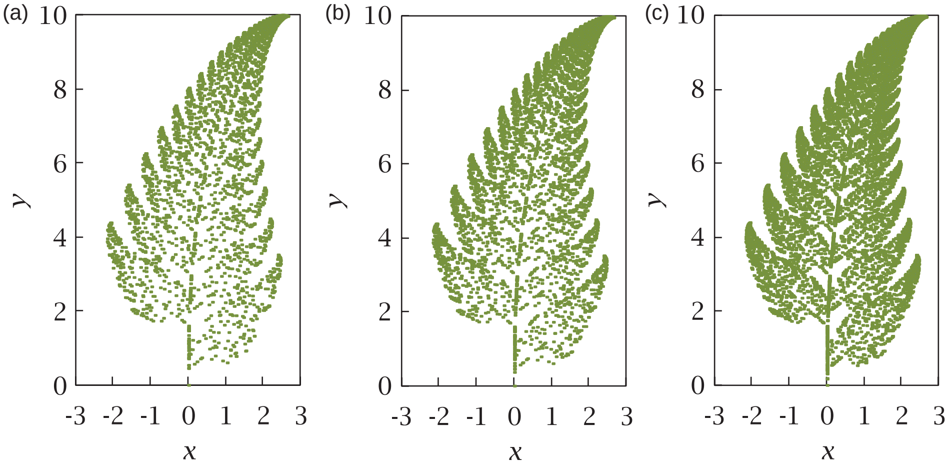
3000 points

Fig. 1: IFS-created models of a tree.



500 points 1000 points 2000 points

Fig. 2: IFS-created models of a fern-leaf for a small number of points.



3000 points 5000 points 10000 points

Fig. 3: IFS-created models of a fern-leaf for a large number of points.

(0,0) for all models. As seen from Fig. 1, the more the number of points is, the clearer the shape is (compare the trees consisting of 500, 1000, 2000, and 3000 points). However, the models of the tree shown in Fig. 1 provide some valuable insights into the IFS-created fractals. For example, all the trees in Fig. 1 are confined to a definite area on the x-y plane, i.e., intervals [-1,1] along the x-axis and [0,2] in the y-axis. This is because of the contracting nature of the mappings as defined by the parameters listed in Tab. 1. Another remarkable feature is the detailedness. Even though the visual appearance of the shape (tree) improves as the number of points increases, there is no guarantee that the detailedness of the shape improves with an increase in the number of points as perceived from the models in Fig. 1. The other remarkable feature is that the visual appearance of the shape becomes independent of the number of points after a certain number of points. For example, compare the models of the tree corresponding to 2000 and 3000 points. These two models look almost the same.

However, the abovementioned features are seen in the cases of other IFS-created fractals. For example, see the case of an IFS-created fractal called fern-leaf introduced by Barnsley [2]. Table 2 lists the settings of the parameters and Figs. 2–3 illustrate four models of the shape (fern-leaf).

In the case of a fern-leaf, four affine mappings are needed instead and the seed $(x_{i=0},y_{i=0})$ is equal to (0,0). As seen from Figs. 2–3, the more the number of points is, the clearer the shape is (compare the fern-leaf models consisting of 500, 1000, 2000, and 3000, 5000, and 10000 points). The shape is always confined to an interval [-3,3] along the x-axis and [0,10] along the y-axis, irrespective of the number of points. This is because of the contracting nature of the mappings as defined by the parameters in Tab. 2. For a small number of points, the shape is hard to recognize to be a fern-leaf, whereas for a large number of points, the shape is easy to recognize to be a fern-leaf (Fig. 3). From the models corresponding to 5000 and 10000 points in Fig. 3, it is clear that the whole

J	a_j	b_j	c_j	d_j	e_j	f_j	p_j
1	0	0	0	0.16	0	0	0.01
2	0.85	0.04	– 0.04	0.85	0	1.6	0.85
3	0.2	– 0.26	0.23	0.22	0	1.6	0.07
4	– 0.15	0.28	0.26	0.24	0	0.44	0.07

Tab. 2: Settings of IFS for creating models of a fern-leaf.

leaf can be subdivided into parts, each of which is a reduced copy of the whole, and the process continues in an endless manner. However, when the subdivision continues beyond a certain degree of contraction, the shape is hard to recognize and becomes a point-filled area rather than a reduced copy of the whole.

4. DFM OF FERN-LEAF

This section describes some of the useful results regarding the DFM of the described shape (fern-leaf). Here, DFM means creating a set of points that models a fern-leaf that is easy to manufacture either by subtractive manufacturing (e.g., milling) or additive manufacturing (e.g., 3-D printing). It is worth mentioning that the main inspiration of the DFM described below is taken from some of the previous works of the authors and others [5, 15, 23, 29–31, 36].

Before describing the proposed DFM, consider the manufacturing of the shape (fern-leaf) by using end-milling operations, as schematically illustrated in Fig. 4. As seen from Fig. 4, tool-paths have been generated from a point-cloud of fern-leaf (Figs. 2–3), i.e., from the points (x_i,y_i), $i = 0,1,\ldots$. A typical result of the physical model is shown in the bottom section of Fig. 4. The tool diameter and the depth of cut have been varied to see how they affect the accuracy of the physical model. It is difficult to manufacture the intended shape (fern-leaf) due to the presence of the closely packed points in some areas in the model (point-cloud) as seen from Fig. 4.

Therefore, an appropriate approach is needed to manage the distribution of the points in the point-cloud of a fern-leaf without destroying its fractalness (i.e., the whole leaf can be subdivided into small leafs, each of which is a reduced copy of the whole). One of the useful approaches is to find out a definite data structure in the point-cloud, as pursued by some authors (e.g., see the works in [5, 15, 29–31]). The other alternative is to redefine the underlying affine mappings [36]. However, the ultimate goals remain the same, as follows:

- How to control the distribution of points in the point-cloud
- How to control the levels of subdivisions
- How to create a set of non-crossing tool-paths
- How to create the accurate convex-hulls

In order to achieve the abovementioned goals, some mapping experiments have been conducted. Some of the useful results are described below.

Recall the four mappings underlying Barnsley's fern-leaf. Table 2 lists their parameters. Each mapping can be used to define an iterative process. Equations (2)-(5) define the four possible iterative processes. In particular, equation (2) defines the iterative process for creating a point-cloud solely by using the first mapping in accordance with the parameters listed in Tab. 2. Equation (3) defines the iterative process for creating a point-cloud solely by using the second mapping in accordance with the parameters listed in Tab. 2. Equation (4) defines the iterative process for creating a point-cloud solely by using the third mapping in accordance with the parameters listed in Tab. 2. Equation (5) defines the iterative process for creating a point-cloud solely by using the last mapping in accordance with the parameters listed in

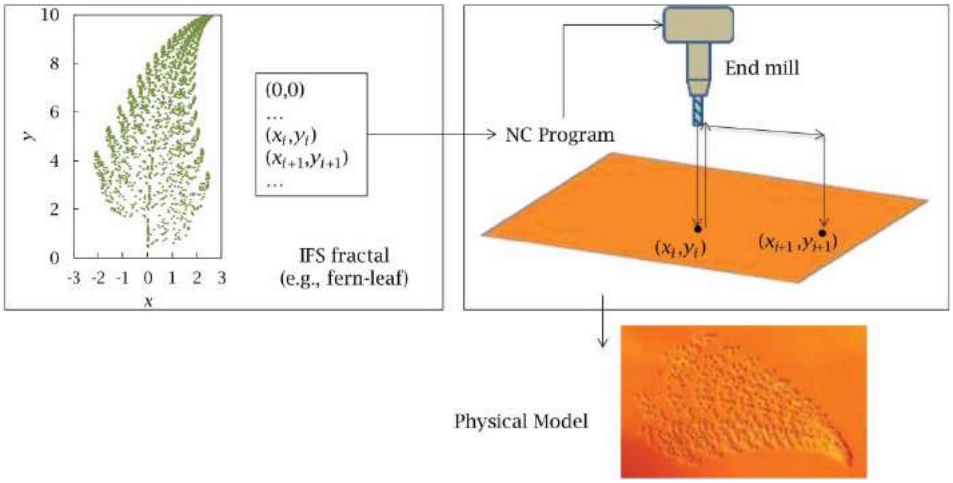


Fig. 4: Physical model building process of a fern-leaf using milling [36].

Tab. 2.

$$\begin{aligned} &\mathit{Input\ setting}: \\ &\quad \mathit{seed}: (x_0, y_0) = (0,0) \quad \mathit{iterations}:\ N \\ &\mathit{Iteration}: \\ &\quad \mathit{For} \quad i = 1, \dots, N \quad x_i = 0 \quad y_i = 0.16 y_{i-1} \end{aligned} \tag{2}$$

$$\begin{aligned} &\mathit{Input\ setting}: \\ &\quad \mathit{seed}:\ (x_0, y_0) = (0,0) \quad \mathit{iterations}:\ N \\ &\mathit{Iteration}: \\ &\quad \mathit{For} \quad i = 1, \dots, N \\ &\qquad x_i = 0.85 x_{i-1} + 0.04 y_{i-1} \\ &\qquad y_i = -0.04 x_{i-1} + 0.85 y_{i-1} + 1.6 \end{aligned} \tag{3}$$

$$\begin{aligned} &\mathit{Input\ setting}: \\ &\quad \mathit{seed}:\ (x_0, y_0) = (0,0) \quad \mathit{iterations}:\ N \\ &\mathit{Iteration}: \\ &\quad \mathit{For} \quad i = 1, \dots, N \\ &\qquad x_i = 0.23 x_{i-1} - 0.26 y_{i-1} \\ &\qquad y_i = 0.23 x_{i-1} + 0.22 y_{i-1} + 1.6 \end{aligned} \tag{4}$$

$$\begin{aligned} &\mathit{Input\ setting}: \\ &\quad \mathit{seed}:\ (x_0, y_0) = (0,0) \quad \mathit{iterations}:\ N \\ &\mathit{Iteration}: \\ &\quad \mathit{For} \quad i = 1, \dots, N \\ &\qquad x_i = -0.15 x_{i-1} + 0.28 y_{i-1} \\ &\qquad y_i = 0.26 x_{i-1} + 0.24 y_{i-1} + 0.44 \end{aligned} \tag{5}$$

Figure 5 shows the scatter plots of the point-clouds that have been created in accordance with the equations (2)-(5), respectively, for 100 iterations. As seen from Fig. 5, all the mappings are strict contractions onto them. As seen from Fig. 5(a), the first mapping (equation (2)) has a fixed point (0,0), and after the first iteration it reaches its fixed point. As seen from Fig. 5(b), the second mapping (equation (3)) has a fixed point near (2.654,9.962), and it takes a relatively large number of iterations to reach its fixed point. The point-cloud remains in the region $x \geq 0$. As seen from Fig. 5(c), the third mapping (equation (4)) has a fixed point near (-0.608,1.872), and it takes a relatively less number of iterations to reach its fixed point. The point-cloud remains in the region $x \leq 0$. As seen from Fig. 5(d), the last mapping (equation (5)) has a fixed point near (0.154,0.631), and it takes a relatively less number of iterations to reach its fixed point. The point-cloud remains in the region $x \geq 0$.

From the described mapping experiments, it is clear that the second-, third-, and fourth-mappings are the most informative mappings, whereas the other mapping is less informative and can be ignored for the sake of DFM. The third- and fourth-mappings keep the points in two particular regions (see the location of the points in Fig. 5(c) (the left-hand-side of $x = 0$) and Fig. (d) (the right-hand-side of $x = 0$)). In order to see the interactions among the mappings, some other mapping experiments defined in equation (6) have been conducted.

$$\begin{aligned} &\mathit{Input\ Setting}: \\ &\quad \mathit{Seed}:\ (x_0, y_0) = (0,0) \quad \mathit{Iterations}:\ N \\ &\quad \mathit{Mappings}:\ M_k = (a_k, b_k, c_k, d_k, e_k, f_k) \quad k = 1, \dots, K \\ &\mathit{Generator}: \\ &\qquad \mathit{For} \quad i = 1, \dots, N \quad x_i = 0.85 x_{i-1} + 0.04 y_{i-1} \\ &\qquad\qquad y_i = -0.04 x_{i-1} + 0.85 y_{i-1} + 1.6 \\ &\qquad \mathit{For} \quad i = 0, \dots, N \quad x_{gi} = x_i \quad y_{gi} = y_i \\ &\mathit{Trans} - 1: \\ &\quad \mathit{Selection}: \quad M_{1\,k} = M_k \quad \exists k\{1, \dots, K\} \\ &\qquad M_{1\,k} = (a_{1\,k}, b_{1\,k}, c_{1\,k}, d_{1\,k}, e_{1\,k}, f_{1\,k}) \\ &\qquad \mathit{For} \quad i = 0, \dots, N \quad x_{1i} = a_{1\,k} x_{gi} + b_{1\,k} y_{gi} + e_{1\,k} \\ &\qquad\qquad y_{1i} = c_{1\,k} x_{gi} + d_{1\,k} y_{gi} + f_{1\,k} \\ &\dots \\ &\mathit{Trans} - q: \\ &\quad \mathit{Selection}: \quad M_{qk} = M_k \quad \exists k\{1, \dots, K\} \\ &\qquad M_{qk} = (a_{qk}, b_{qk}, c_{qk}, d_{qk}, e_{qk}, f_{qk}) \\ &\qquad \mathit{For} \quad i = 0, \dots, N \quad x_{qi} = a_{qk} x_{q-1\,i} + b_{qk} y_{q-1\,i} + e_{qk} \\ &\qquad\qquad y_{qi} = c_{qk} x_{q-1\,i} + d_{qk} y_{q-1\,i} + f_{qk} \\ &\dots \end{aligned} \tag{6}$$

As defined in equation (6), the second-mapping of the original settings (Tab. 2) is applied recursively (onto basis) (exactly like it is in equation (3)) to create a set of points called generator-points denoted as (x_{gi},y_{gi}), $i = 0,\dots,N$. Afterwards, a series of transformations denoted as *Trans-q*, $q = 1,2,\dots,N$ continues, wherein the points are created on the one-to-one basis (not on the onto basis). The three sets of mapping parameters used in these transformations are listed in Tab. 3. Thus, it is true that $M_{qk} \in \{M_{k=1}, M_{k=2}, M_{k=3}\}$, $\forall q \in \{1,\dots,N\}$ so that $M_{k=1}$ = (0.85,0.04,-0.04,0.85,0,1.6) (the parameters of the second-mapping in the original settings), $M_{k=2}$ = (0.2,-0.26,0.23,0.22,0,1.6) (the parameters of the third-mapping in the original settings), or $M_{k=3}$ = (-0.15,0.28,0.26,0.24,0,0.44) (the parameters of the fourth-mapping in the original settings).

Let the sets of points created by the process defined in equation (6) be $A_0 = \{(x_{gi},y_{gi}) \mid i = 0,\dots,N\}$ and $A_q = \{(x_{qi},y_{qi}) \mid i = 0,\dots,N\}$, $q = 1,\dots,N$, and $\boldsymbol{T} = (t_0, t_1,\dots,t_q,\dots,t_N)$ be the vector representing the sequence of the sets of mapping parameters, wherein $t_0 = 1$ and $t_q = 1$, 2, or 3 if $M_{qk} = M_{k=1}$, $M_{k=2}$, or $M_{k=3}$, respectively, for all $q = 1,\dots,N$. Let the set of points $A = A_0 \cup \dots A_q \cup \dots \cup A_N$

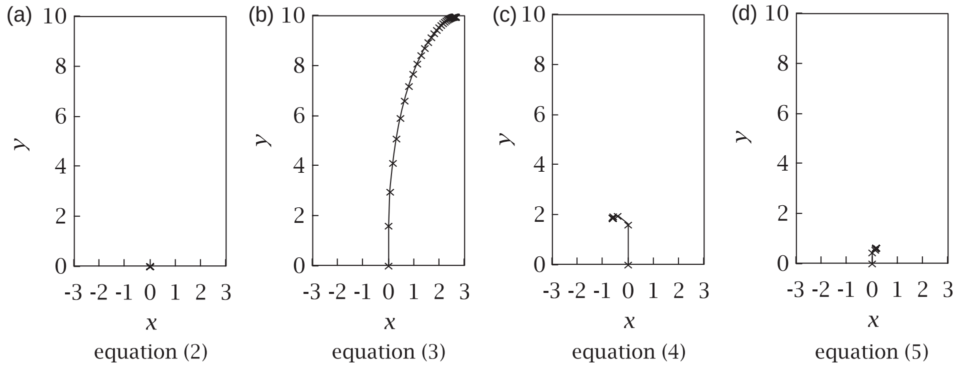


Fig. 5: Natures of the four mappings underlying Barnsley's fern-leaf.

k	a_k	b_k	c_k	d_k	e_k	f_k
1	0.85	0.04	-0.04	0.85	0	1.6
2	0.2	-0.26	0.23	0.22	0	1.6
3	-0.15	0.28	0.26	0.24	0	0.44

Tab. 3: Mapping parameters for the DFM of a fern-leaf.

be the point-cloud. (The points corresponding to $A_q \cup \ldots \cup A_N$ can be renamed depending on their relative position with respect to A_0 as describe below.) The sequential order of the sets of mapping parameters determines whether or not the point-cloud (i.e., A) will be useful enough for the intended purpose (i.e., DFM of the shape (fern-leaf)). Keeping this in mind, numerous forms of $\boldsymbol{T}$ have been considered and the underlying point-clouds have been observed. It has been found that the sequence of the sets of mapping parameters $\boldsymbol{T} = (1, 2 \text{ or } 3, 1, 1, \ldots)$ helps create useful point-clouds as shown in Fig. 6.

As seen from Fig. 6(a), $\boldsymbol{T} = (1, 2, 1, 1, \ldots)$ results the sets of points A_0, $A_{1\ \mathrm{L}}$, $A_{2\ \mathrm{L}}$, $A_{3\ \mathrm{L}}$, ... This means that the points of A_0 move to the lowest-left corner of A_0 due to the application of the set of mapping parameters corresponding to $M_{k=2}$, creating a set of points $A_{1\ \mathrm{L}}$ ($A_{q=1}$). These points move upward due to the successive applications the same set of mapping parameters corresponding to $M_{k=1}$, creating the sets of points $A_{2\ \mathrm{L}}$ ($A_{q=2}$), $A_{3\ \mathrm{L}}$ ($A_{q=3}$), ..., respectively. On the other hand, $\boldsymbol{T} = (1, 3, 1, 1, \ldots)$ results the sets of points A_0, $A_{1\mathrm{R}}$, $A_{2\mathrm{R}}$, $A_{3\mathrm{R}}$, ..., as seen from Fig. 6(b). This means that the points of A_0 move to the lowest-right corner of A_0 due to the application of the mapping parameters corresponding to $M_{k=3}$, creating a set of points $A_{1\mathrm{R}}$ ($A_{q=1}$). These points move upward due to the successive applications of the same set of mapping parameters corresponding to $M_{k=1}$, creating the sets of points $A_{2\mathrm{R}}$ ($A_{q=2}$), $A_{3\mathrm{R}}$ ($A_{q=3}$), ..., respectively.

The points in the set A_0 serve as the level-1 model of a fern-leaf, whereas the points in the sets $A_{1\ \mathrm{L}}$, $A_{2\ \mathrm{L}}$, ..., A_{NL} and $A_{1\mathrm{R}}$, $A_{2\mathrm{R}}$, ..., A_{NR} collectively serve

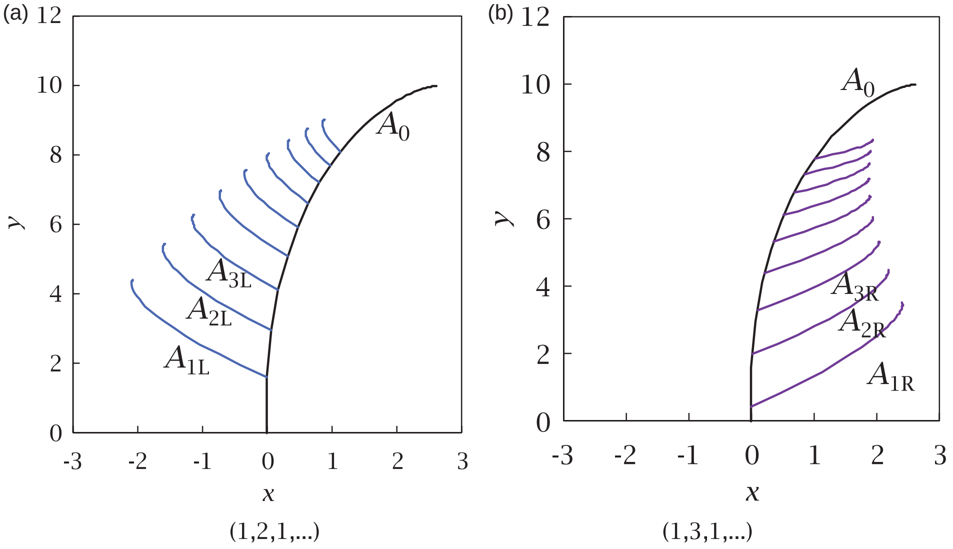


Fig. 6: Point-clouds corresponding to $\boldsymbol{T} = (1, 2 \text{ or } 3, 1, \ldots)$.

© 2014 CAD Solutions, LLC, http://www.cadanda.com

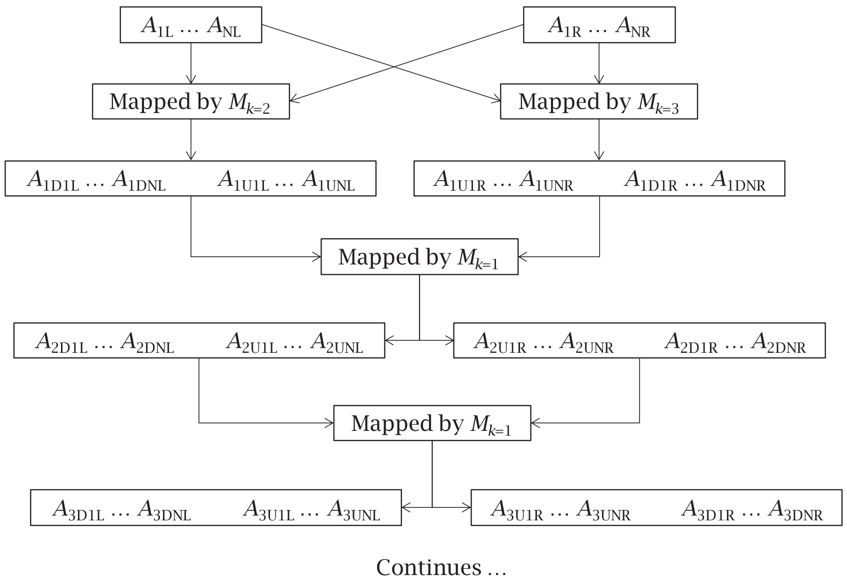

$A_{1L} \dots A_{NL}$
$A_{1R} \dots A_{NR}$
Mapped by $M_{k=2}$
Mapped by $M_{k=3}$
$A_{1D1L} \dots A_{1DNL}$ $A_{1U1L} \dots A_{1UNL}$
$A_{1U1R} \dots A_{1UNR}$ $A_{1D1R} \dots A_{1DNR}$
Mapped by $M_{k=1}$
$A_{2D1L} \dots A_{2DNL}$ $A_{2U1L} \dots A_{2UNL}$
$A_{2U1R} \dots A_{2UNR}$ $A_{2D1R} \dots A_{2DNR}$
Mapped by $M_{k=1}$
$A_{3D1L} \dots A_{3DNL}$ $A_{3U1L} \dots A_{3UNL}$
$A_{3U1R} \dots A_{3UNR}$ $A_{3D1R} \dots A_{3DNR}$
Continues ...

Fig. 7: Creating level-3 model of a fern-leaf.

as the level-2 model of a fern-leaf. It is worth mentioning that the each set of points (points on each curved line shown in Fig. 6) contains only 35 points, i.e., $N = 34$. It is observed that the distance between two consecutive points becomes very short after the 10-th iteration. One may use any number of points to create the level-1 and level-2 models, but the results will be similar to that of in Fig. 6, unless a different formulation of $\boldsymbol{T}$ is chosen. In synopsis, the sequences of mapping parameters $\boldsymbol{T} = (1{,}2 \text{ or } 3{,}1{,}1, \dots)$ can control the shape of a fern-leaf and the distributions of the points in the point-cloud (i.e., model).

However, it is not difficult to create the level-3 model of a fern-leaf and beyond. As an example, Fig. 7 schematically illustrates the procedure for creating the level-3 model of a fern-leaf.

As seen from Fig. 7, the points in the sets $A_{1\,L}$, $A_{2\,L}, \dots, A_{NL}$ and $A_{1R}, A_{2R}, \dots, A_{NR}$ must be mapped by the mapping $M_{k=2}$ in a one-to-one basis for creating the points of the lowest-left (or the first-left) leaf (around the points of $A_{1\,L}$) of the level-3 model. The sets of points of the first-leaf of the level-3 model are denoted as $A_{1D1\,L} \dots A_{1DNL}$ and $A_{1U1\,L} \dots A_{1UNL}$. In the subscript, the letter "D" denotes "down" and "U" denotes "up" with respect to the points of $A_{1\,L}$ (i.e., around the points in the first-left leaf of the level-2 model). This convention is true for other cases described below. The points $A_{1D1\,L} \dots A_{1DNL}$ and $A_{1U1\,L} \dots A_{1UNL}$ must successively be mapped by the same mapping $M_{k=1}$ to get the points of the other leafs of the level-3 model. This means that the sequence of the sets of mapping parameters $\boldsymbol{T} = (1{,}2 \text{ or } 3{,}2{,}1{,}1, \dots)$ creates the points of the level-3 model in the left-hand-side of the level-1 model. On the other hand, the sequence of the sets of mapping parameters $\boldsymbol{T} = (1{,}3 \text{ or } 2{,}3{,}1{,}1, \dots)$ creates the points of the leafs of the level-3 model in the right-hand-side leaf of the level-1 model.

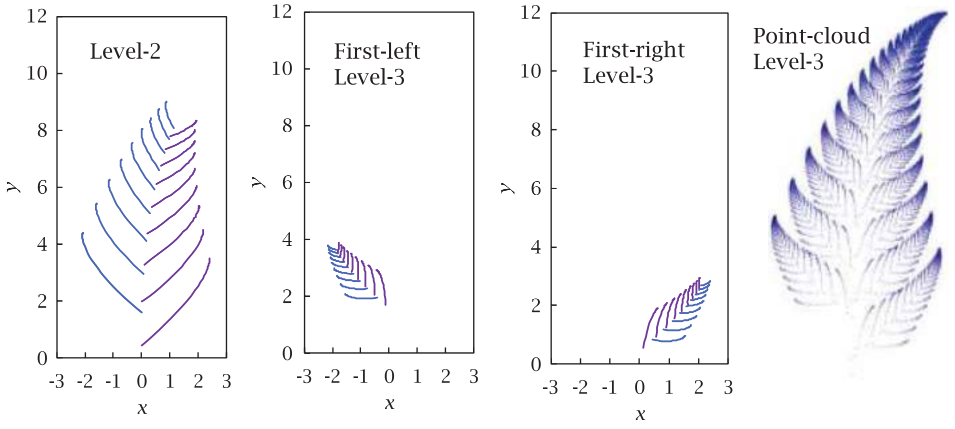


Fig. 8: DFM of a fern-leaf up to level-3.

© 2014 CAD Solutions, LLC, http://www.cadanda.com

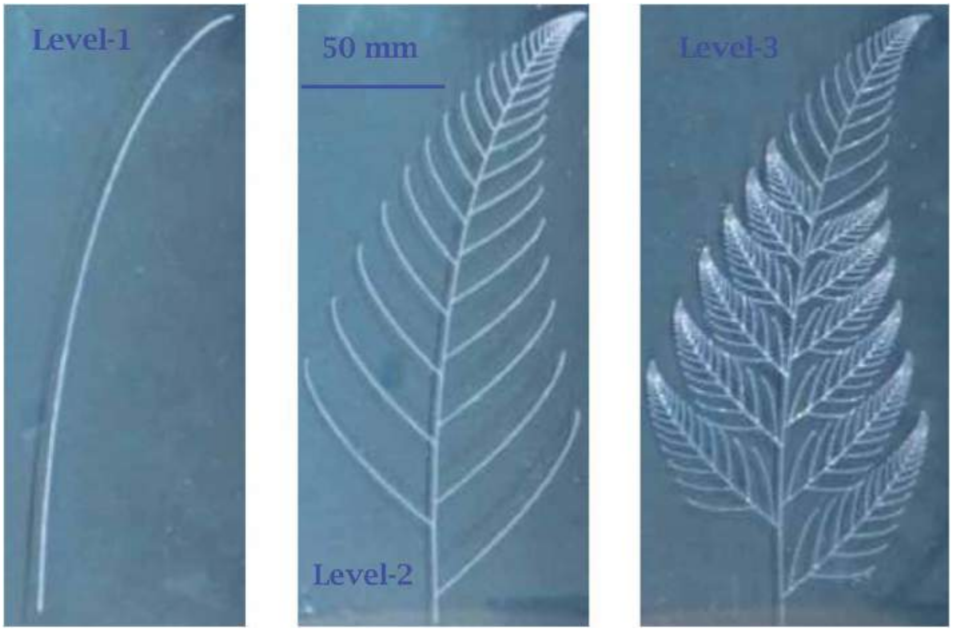


Fig. 9: Physical models of a fern-leaf up to level-3 produced by milling ($N = 34$).

The modeling result obtained by applying the described procedure (Fig. 7) is shown in Fig. 8. Particularly, the scatter plot in the left-hand-side of Fig. 8 shows some of the points of the level-2 model, whereas the two scatter plots in the middle of Fig. 8 show the corresponding points of the level-3 model. The correspondence of the points in the respective levels is shown by the purple and blue colors. The right-hand-side plot in Fig. 8, on the other hand, is the entire point-cloud, i.e., the level-3 model, of a fern-leaf for $N = 34$. As seen from the point-cloud, some of the regions in the model are still covered by points. Reducing the number of points (i.e., keeping N as small as possible), can help reduce the point-filled regions. (See the next section for the details.) In other words, not only the sequential order of the mappings (i.e., the nature of $\boldsymbol{T}$) but also the number of iterations (N) is an important issue for the DFM of a fern-leaf, in particular, and IFS-created fractals, in general.

5. MANUFACTURING

This section describes some of the results obtained while creating the physical models of the redesigned fern-leaf with the aid of both subtractive manufacturing (i.e., milling) and additive manufacturing (i.e., 3-D printing). It is worth mentioning that sometimes the additive manufacturing becomes highly energy intensive (i.e., less environmentally-friendly) and the alternative process, i.e., the subtractive manufacturing, can be used to manufacture the model instead [35]. Thus, both options, additive and subtractive manufacturing, must be explored for getting a better insight into the physical model building process of IFS-created fractals.

The model-building results corresponding to the subtractive manufacturing (i.e., milling) are described first as follows. Figures 9–10 shows some of the physical models that have been built by a desktop CNC milling machine available at the Kitami Institute of Technology. Each set of points as described in the previous section (A_0, $A_{1\,L}$, ... ,A_{NL}, A_{1R}, ... ,A_{NR}, $A_{1D1\,L}$, ... ,A_{1DNL}, $A_{1U1\,L}$, ... ,A_{1UNL}, A_{1D1R}, ... ,A_{1DNR}, A_{1U1R}, ... ,A_{1UNR}, ...) has been used to create a single tool-path. Figure 9 shows the physical models of the level-1 and level-2 (full) and level-3 (partial) models corresponding to $N = 34$ (see Fig. 8). It has been observed (Fig. 9) that the cutter diameter affects the models when the cutter approaches the end of each tool-path. As a result, the physical models are accurate to some extend and defective to some other extend. Nevertheless, the overall quality of the model is much better than that of in Fig. 4. However, reducing the number of points in each set of points (reducing N) can eliminate the areas affected by the cutter diameter. In order to see this possibility, several physical models have been produced by choosing different values of N. Two examples of such results are shown in Fig. 10 wherein the point-clouds and the corresponding physical models are shown side by side for $N = 7$ (7-point models) and 9 (9-point models). As seen from Fig. 10, the cutter diameter affected regions still exist in the case of 9-point model, whereas there are no such regions in the case of 7-point model. Thus, in order to produce a defect-free physical model of the shape (fern-leaf), one needs to choose a suitable scale of magnification and the number of points

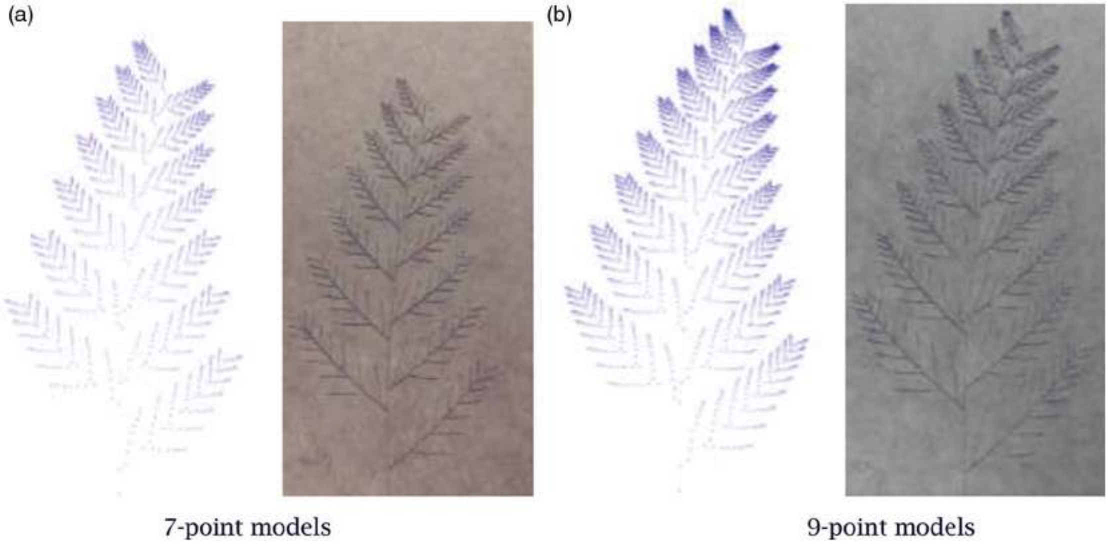


Fig. 10: Level-3 physical models of a fern-leaf produced by milling for $N = 7$ and 9.

underlying each set of points or tool-path (i.e., the value of N).

The results shown up to this point do not involve any commercial CAD/CAM systems. However, one may take the help of commercial CAD systems for creating the 3-D CAD models from the point-clouds. If the created CAD models are accurate enough, then the models can be used for manufacturing physical models using the additive manufacturing or other means. In order to explore this possibility, a set of experiments has been conducted. In these experiments, a commercially available software package has been used. The package can operate on a given point-cloud and creates a 3-D surface/mesh (3-D CAD model). Figure 11 shows some of the point-clouds and the corresponding 3-D CAD models of a fern-leaf.

Two point-clouds placed at different heights in the z-axis have been used to create a surface or mesh (convex or concave) as seen from Fig. 11. Irrespective of the number of points in a point-cloud, the mesh/surface models are not as accurate as expected. Sometimes, the distribution of the points affects the 3-D CAD model. For example, consider the second case from the top in Fig. 11. The model not even includes all points of the given point-cloud while creating a concave mash model. In all cases, however, the right-hand-side segments of the shape remain relatively less accurate. The accuracy can be improved either by using a point-cloud up to the level-2 (the first case from the top in Fig. 11) or by using a model having reduced number of points up to the level-3 (the second case from the top in Fig. 11). As shown in the last case in Fig. 11, instead of using the point-cloud directly, a grid of rectangular cells on the x-y plane can be considered. The cell height can be determined by whether or not at least a point of the point-cloud belongs to it. This grid-based modeling also produces less accurate 3-D CAD model, as seen from the last case from the top in Fig. 11. One advantage of the grid-based modeling is, however, that the detailedness in the inner section of the leaf can somewhat be preserved.

The above mentioned results imply that the point-cloud as such is not good enough for creating accurate 3-D CAD model of a fern-leaf. Further modifications are needed. One of the options is to representing the outermost boundary of the point-cloud using a set of triangles. The triangles can be organized as a collection of facets [12]. The facets can be used to create an STL file [12]. The STL file can further be processed using a commercially available CAD or RP system. The triangulation can be applied to the level-2 or -3 models as preferred.

In order to see the applicability of the abovementioned triangulation process, an experiment has been conducted, as schematically illustrated in Fig. 12. As seen from Fig. 12, triangles have been created representing the top, bottom, and side surfaces defined by the two sets of point-clouds placed at two different heights along the z-axis. The points shown in Fig. 12 correspond to all points in the set A_0 and the last points in the sets $A_{1\text{ L}}, A_{2\text{ L}}, \ldots, A_{\text{NL}}$ and $A_{1\text{R}}, A_{2\text{R}}, \ldots, A_{\text{NR}}$, for $N = 34$ (see Section 4). The triangles have been organized in the form of facets and an ASCII-STL file has been created using the facets. A CAD package has been used that reads the STL file displays the underlying 3-D model as shown in the upper-right corner in Fig. 12. As seen from Fig. 12, this time the 3-D CAD model is much more accurate compared to those shown in Fig. 11. Particularly, the right-hand-side of the 3-D model does not suffer the problem that is observed in the 3-D CAD models shown in Fig. 11. This means that this time the convex hull is much more accurate. The data stored in the

© 2014 CAD Solutions, LLC, http://www.cadanda.com

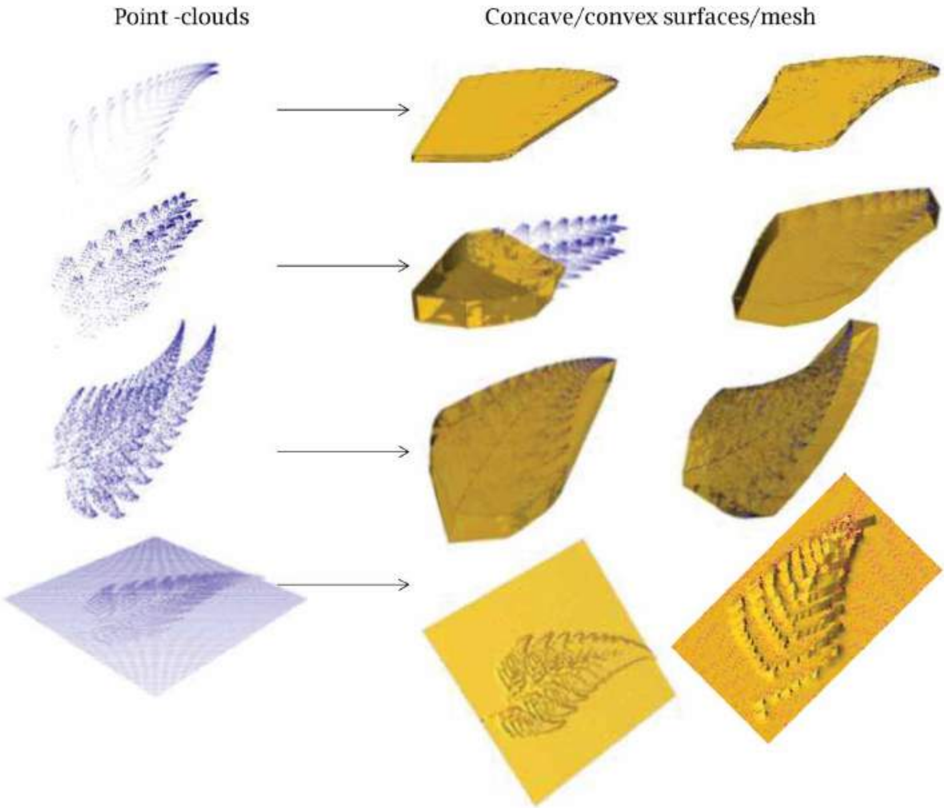


Fig. 11: 3-D CAD models of a fern-leaf.

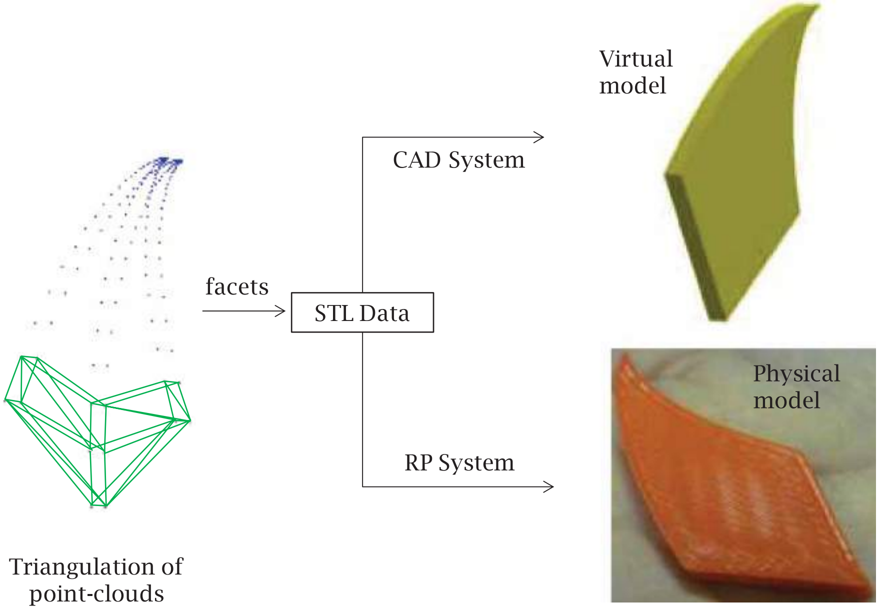


Fig. 12: Physical model-building process of a fern-leaf using additive manufacturing.

STL file has also been used to manufacture a physical model using a 3-D printer available at the Kitami Institute of Technology, as shown in the bottom-left corner in Fig. 12. As seen from Fig. 12, the RP model resembles the 3-D CAD model.

However, the (shape) accuracy of the physical model can further be improved by adding some selected points taken from the level-3 model. For example, the outermost points in the bottom-left/right leafs of the level-3 model (i.e., the last points in the sets $A_{1D1\ L} \ldots A_{1DNL}$ and $A_{1D1R} \ldots A_{1DNR}$)) (see Fig. 7) can be added to the points shown in Fig. 12, to improve the shape-accuracy of the bottom segment of the physical model. This possibility is open for further research. One can produce multiple physical models of a fern-leaf. The scales of the models can be varied as preferred. This issue also remains open for further study.

6. CONCLUDING REMARKS

A set of mapping experiments reveals some data structures in an IFS-created fractal shape called fern-leaf. These data structures help develop an effective procedure for DFM. Particularly the levels of subdivision of a fern-leaf can easily be controlled by the presented DFM procedure. Both one-to-one and onto mappings are involved in the presented DFM procedure. The presented DFM procedure is shown equally effective for both additive manufacturing technology (3-D printing) and subtractive manufacturing technology (milling) in order to manufacture physical models of a fern-leaf. The data points belonging to all three levels of the redesigned virtual model of a fern-leaf are needed to manufacture its physical models. Although the scope of this work is limited to the shape of a fern-leaf, the presented DFM procedure can be applied to other IFS-created fractals (e.g., tree, dragon) for manufacturing more accurate physical models. In this case the mapping procedure defined in equation (6) (i.e., a combination of onto and one-to-one mappings among some selected ones) will help solve the problem. Further research can be carried out to delve into this possibility.

Unlike the design and manufacturing systems that deal with the Euclidian geometry based shapes, the design and manufacturing systems that deal with the fractal geometry based shapes are in the phase of development. Thus, it is worth delving into the development of cost-effective and user-friendly systems intended for the design and manufacturing of fractal geometry based shapes. This study is one of the endeavors toward materializing such systems.

ACKNOWLEDGMENTS

The first author acknowledges Mr. Ryuta Omori with whom the author initiated the work described in this article. The authors thank the reviewers for their constructive suggestions.

REFERENCES

[1] Barnsley, M.F.; Demko, S.: Iterated Function Systems and the Global Construction of Fractals, Proceedings of the Royal Society of London Series A: Mathematical and Physical Sciences, 399(1817), 1985, 243–275. http://dx.doi.org/10.1098/rspa.1985.0057

[2] Barnsley, M.F.: Fractals Everywhere (New Edition), Dover Publications, Mineola, New York, 2012.

[3] Brown, C.A.; Johnsen, W.A.; Butland, R.M.; Bryan, J.: Scale-Sensitive Fractal Analysis of Turned Surfaces, CIRP Annals-Manufacturing Technology, 45(1), 1996, 515–518. http://dx.doi.org/10.1016/S0007-8506(07)63114-X

[4] Chan, P.L.; Hui, K.C.: Surface Pattern Generation Using L-system, Computer-Aided Design and Applications, 10(2), 2013, 339–353. http://dx.doi.org/10.3722/cadaps.2013.339-353

[5] Chiu, W.K.; Yeung, Y.C.; Yu, K.M.: Toolpath generation for layer manufacturing of fractal objects, Rapid Prototyping Journal, 12(4), 2006, 214–221. http://dx.doi.org/10.1108/13552540610682723

[6] Demko, S.; Hodges, L.; Naylor, B.: Construction of fractal objects with iterated function systems, Computer Graphics, 19(3), 1985, 271–278. http://dx.doi.org/10.1145/325165.325245

[7] Eglash, R.: Fractals in African settlement architecture, Complexity, 4(2), 1998, 21–29. http://dx.doi.org/10.1002/(SICI)1099-0526(199811/12)4:2 < 21::AID-CPLX6 > 3.0.CO;2-F

[8] Falconer, K.: Fractal Geometry: Mathematical Foundations and Applications (Second Edition), Wiley, New York, 2003. http://dx.doi.org/10.1002/0470013850

[9] Griffiths, J.G.: Toolpath based on Hilbert's curve, Computer-Aided Design, 26(11), 1994, 839–844. http://dx.doi.org/10.1016/0010-4485(94)90098-1

[10] Hohlfeld, R.G.; Cohen, N.: Self-similarity and the geometric requirements for frequency independence in antennae, Fractals, 7(1), 1999, 79–84. http://dx.doi.org/10.1142/S0218348X99000098

[11] Hutchinson, J.E.: Fractals and self-similarity, Indiana University Mathematics Journal, 30(5), 1981, 713–747. http://dx.doi.org/10.1512/iumj.1981.30.30055

[12] ISO Standard.: Standard specification for additive manufacturing file format (AMF) Version 1.1, ISO/ASTM 52915:2013, 2013.

[13] Kerbrat, O.; Mognol, P.; Jean-Yves Hascoët, J.-Y.: A new DFM approach to combine machining and additive manufacturing, Computers in Industry, 62(7), 2011, 684–692. http://dx.doi.org/10.1016/j.compind.2011.04.003

[14] Kumar, G.S.; Pandithevan, P.; Ambatti, A.R.: Fractal raster tool paths for layered manufacturing of porous objects, Virtual and Physical Prototyping, 4(2), 2009, 91–104. http://dx.doi.org/10.1080/17452750802688215

[15] Li, C.L.; Yeung, Y.C.; Chiu, W.K.; Yu, K.M.: Modelling of complex fractal objects for aesthetic product development, International Journal of

© 2014 CAD Solutions, LLC, http://www.cadanda.com

Product Development, 4(3-4), 2007, 207–224. http://dx.doi.org/10.1504/IJPD.2007.012492

[16] Mandelbrot, B.B.: Fractals: form, chance, and dimension, W.H. Freeman and Co., San Francisco, CA, 1977.

[17] Mandelbrot, B.B.: How long is the coast of Britain? Statistical self-similarity and fractional dimension, Science, 156, 1967, 636–638. http://dx.doi.org/10.1126/science.156.3775.636

[18] Mandelbrot, B.B.: The Fractal Geometry of Nature, W. H. Freeman: San Francisco, 1982.

[19] Martyn, T.: Realistic rendering 3D IFS fractals in real-time with graphics accelerators, Computers & Graphics, 34(2), 2010, 167–175. http://dx.doi.org/10.1016/j.cag.2009.10.001

[20] Martyn, T.: The attractor-wrapping approach to approximating convex hulls of 2D affine IFS attractors, Computers & Graphics, 33(1), 2009, 104–112. http://dx.doi.org/10.1016/j.cag.2008.08.003

[21] Mishkinis, A.; Gentil, C.; Lanquetin, S.; Sokolov, D.: Approximate convex hull of affine iterated function system attractors, Chaos, Solitons & Fractals, 45(11), 2012, 1444–1451. http://dx.doi.org/10.1016/j.chaos.2012.07.015

[22] Mizugaki, Y.; Sakamoto, M.; Sata, T.: Fractal Path Generation for a Metal-Mold Polishing Robot System and Its Evaluation by the Operability, CIRP Annals-Manufacturing Technology, 41(1), 1992, 531–534. http://dx.doi.org/10.1016/S0007-8506(07)61261-X

[23] Omori, R.; Ullah, A.M.M.S.; Tamaki, J.; Kubo, A.: Toward Design for Manufacturing of IFS Fractals, Proceedings of The 5th International Conference on Manufacturing, Machine Design and Tribology (ICMDT2013), Busan, South Korea, 2013, 22–25.

[24] Pang, W.; Hui, K.C.: Interactive Evolutionary 3D Fractal Modeling with Modified IFS, Computer-Aided Design and Applications, 6(1), 2009, 55–67. http://dx.doi.org/10.3722/cadaps.2009.55-67

[25] Purintrapiban, U.; Kachitvichyanukul, V.: Detecting patterns in process data with fractal dimension, Computers & Industrial Engineering, 45(4), 2003, 653–667. http://dx.doi.org/10.1016/j.cie.2003.09.004

[26] Rochman, D.; Vázquez, S.: A Conceptual Framework Based on Fractal Geometry for Design, Modeling and Rapid Prototyping of Complex Geometric Shapes, Computer-Aided Design and Applications, 10(2), 2013, 307–319. http://dx.doi.org/10.3722/cadaps.2013.307-319

[27] Ruschin-Rimini, N.; Maimon, O.; Romano, R.: Visual analysis of quality-related manufacturing data using fractal geometry, Journal of Intelligent Manufacturing, 23(3), 2012, 481–495. http://dx.doi.org/10.1007/s10845-010-0387-2

[28] Shin, M.; Mun, J.; Jung, M.: Self-evolution framework of manufacturing systems based on fractal organization, Computers & Industrial Engineering, 56(3), 2009, 1029–1039. http://dx.doi.org/10.1016/j.cie.2008.09.014

[29] Soo, S.C.; Yu, K.M.; Chiu, W.K.: Modeling and fabrication of artistic products based on IFS fractal representation, Computer-Aided Design, 38(7), 2006, 755–769. http://dx.doi.org/10.1016/j.cad.2006.04.003

[30] Soo, S.C.; Yu, K.M.: Rapid prototyping for self-similarity design, Journal of Materials Processing Technology, 139(1–3), 2003, 219–225. http://dx.doi.org/10.1016/S0924-0136(03)00223-1

[31] Soo, S.C.; Yu, K.M.: Tool-Path Generation for Fractal Curve Making, The International Journal of Advanced Manufacturing Technology, 19(1), 2002, 32–48. http://dx.doi.org/10.1007/PL00003966

[32] Stewart, D.E.: Towards numerically estimating Hausdorff dimensions, The ANZIAM Journal, 42, 2001, 451–461. http://dx.doi.org/10.1017/S1446181100012207

[33] Sun, W., Starly, B.: Internal Scaffold Architecture Designs using Lindenmayer Systems, Computer-Aided Design and Applications, 4(1-4), 2007, 395–403. http://dx.doi.org/10.1080/16864360.2007.10738559

[34] Ullah, A.M.M.S.; Harib, K.H.: Manufacturing Process Performance Prediction by Integrating Crisp and Granular Information, Journal of Intelligent Manufacturing, 16(3), 2005, 319–332. http://dx.doi.org/10.1007/s10845-005-7026-3

[35] Ullah, A.M.M.S.; Hashimoto, H.; Kubo, A.; Tamaki, J.: Sustainability analysis of rapid prototyping: material/resource and process perspectives, International Journal of Sustainable Manufacturing, 3(1), 2013, 20–36. http://dx.doi.org/10.1504/IJSM.2013.058640

[36] Ullah, A.M.M.S.; Omori, R.; Nagara, Y.; Kubo, A.; Tamaki, J.: Toward Error-Free Manufacturing of Fractals, Procedia CIRP, 12(2013), 2013, 43–48. http://dx.doi.org/10.1016/j.procir.2013.09.009

[37] Ullah, A.M.M.S.; Rahman, M.R.; Kachitvichyanuluk, V.; Harib, K.H.: Fractal Dimension: A New Machining Decision-Making Parameter, Intelligence in a Small Materials World, DEStech Publications, Lancaster, PA, 2005.

[38] Wannarumon, S.; Bohez, E.L.J., Annanon, K.: Aesthetic evolutionary algorithm for fractal-based user-centered jewelry design, Artificial Intelligence for Engineering Design, Analysis and Manufacturing: AIEDAM, 22(1), 2008, 19–39. http://dx.doi.org/10.1017/S0890060408000024

© 2014 CAD Solutions, LLC, http://www.cadanda.com

[39] Wannarumon, S.; Bohez, E.L.J.: A New Aesthetic Evolutionary Approach for Jewelry Design, Computer-Aided Design and Applications, 3(1-4), 2006, 385–394. http://dx.doi.org/10.1080/16864360.2006.10738477

[40] Wannarumon, S.: An Aesthetics Driven Approach to Jewelry Design, Computer-Aided Design and Applications, 7(4), 2010, 489–503. http://dx.doi.org/10.3722/cadaps.2010.489–503

[41] Yeung, Y.C.; Yu, K.M.: Manufacturability of Fractal Geometry, Materials Science Forum, 471–472, 2004, 722–726. http://dx.doi.org/10.4028/www.scientific.net/MSF.471-472.722

© 2014 CAD Solutions, LLC, http://www.cadanda.com