

Fractal Deformation Using Displacement Vectors Based on Extended Iterated Shuffle Transformation

Tadahiro Fujimoto

Iwate University

fujimoto@cis.iwate-u.ac.jp

Yoshio Ohno

Keio University

ohno@on.cs.keio.ac.jp

Kazunobu Muraoka*

Iwate University

muraoka@cis.iwate-u.ac.jp

Norishige Chiba

Iwate University

nchiba@cis.iwate-u.ac.jp

Abstract

In this paper, we propose a framework of “fractal deformation” using displacement vectors based on “extended Iterated Shuffle Transformation (ext-IST)”. An ext-unit-IST is a one-to-one and onto mapping that is extended from a unit-IST, which we have proposed, and is basically defined on a code space. When the mapping is applied on a geometric space, a fractal-like repeated structure, which is referred to as “local resemblance in space/scale directions”, is constructed on the relationship between points on the domain and those on the range. By applying the mapping to displacement vectors given on a geometric shape, the shape can be deformed in the fractal-like repeated manner. This fractal deformation is easy to control by changing the displacement vectors intuitively. In addition, a continuous transition between a continuous deformation and a fractal deformation can be realized. We demonstrate how the fractal deformation technique produces attractive results by showing various examples.

Key words: computer graphics (CG), geometric model, shape deformation, fractal, Iterated Function System (IFS), attractor, Iterated Shuffle Transformation (IST)

1 Introduction

Shape deformation is a useful technique when we create or modify the shape of an object. A lot of deformation techniques have been proposed. For example, *free form deformation (FFD)*[11, 16] is one of the most familiar deformation techniques. In FFD, the shape of a 3D object is deformed by deforming a 3D lattice containing the object. Other useful techniques are 2D or 3D *morphing* and *warping*[11], which are one of the most active topics in current computer graphics research. The former produces a natural continuous transition between two objects, and the latter deforms a single object. Besides, as a kind of shape deformation, some *fractal deformation* techniques have been proposed[4, 5, 12, 14, 15, 17, 18, 19]. These techniques

treat fractal shapes and deform them (cf. Sec.2). Fractal deformation techniques achieve completely different deformation styles from usual continuous deformation techniques such as FFD; a fractal deformation technique deforms a shape in the way that each subpart of the shape in all scales is deformed recursively, while a continuous deformation technique deforms the whole shape continuously.

In this paper, we propose a new framework of fractal deformation using *displacement vectors* based on *extended Iterated Shuffle Transformation (ext-IST)*. So far we have proposed the idea of IST’s to define our parametric surfaces[7, 8]. The IST’s consist of two types: *unit-IST*’s and *connected-IST*’s. In this paper, we focus on the unit-IST’s and extend them to propose *extended unit-IST*’s (*ext-unit-IST*’s). A unit-IST is a one-to-one and onto mapping that is basically defined on a code space. The mapping can be applied on a ge-

*Presently, Tohoku Institute of Technology,
muraoka@tohtech.ac.jp

ometric space by mapping codes on the code space to points on the geometric space. On the geometric space, the unit-IST constructs a fractal-like repeated structure on the relationship between points on the domain and those on the range. The property of this structure is referred to as “*local resemblance in space/scale directions*”, which is considered to be the unification of “*locality in space directions*” in Euclidean geometry and “*self-similarity in scale directions*” in fractal geometry. We extend this original unit-IST to the ext-unit-IST in order to realize wider variety of local resemblance property. The fractal deformation technique proposed in this paper deforms a geometric shape in the fractal-like repeated manner based on this property. The proposed technique can be applied to any geometric shape if the points on the shape can be addressed appropriately. Then, the shape is deformed by applying the ext-unit-IST to displacement vectors given on the shape. The deformation can be controlled easily by manipulating the displacement vectors intuitively. Moreover, the ability of the ext-unit-IST realizes a continuous transition between a continuous deformation and a fractal deformation.

In Sec.2, we describe the previous work related to fractal deformation and state the difference between other techniques and the proposed technique. In Sec.3, we define ext-unit-IST’s. In Sec.4, we explain the local resemblance. In Sec.5, we propose the fractal deformation technique based on ext-unit-IST’s. In Sec.6, we describe the conclusion and future work.

2 Related Work

Some fractal deformation techniques that have been thus far proposed are related to *IFS theory*, in which a fractal shape is defined as an *attractor* obtained by applying a set of contraction mappings infinitely (cf. Sec.5.1). The techniques proposed in [4, 5, 15] treat IFS attractors and deform them by changing their contraction mappings. The method described in [15] produces a continuous transition between two IFS attractors by directly interpolating the coefficients of their contraction mappings. The technique in [4] deforms an IFS attractor by changing the fixed point and the strength of attraction of the point of each contraction mapping. The study in [5] proposed a method for maintaining the connectedness of a deformed IFS attractor during the deformation process. Besides, in [14], a fractal shape is defined based on a recursive functional equation on the complex plane and is analyzed topologically; using this definition form, a fractal metamorphosis of the shape is achieved by changing the functional equation continuously. In [12], a simple procedural technique for creat-

ing a moving fractal tree was proposed; the shape of a tree is first defined using a simple recursive procedure that gives proper scale factors to the branch length, and then is given motions by rotating the branches. Here, *fractal interpolation*[1, 3, 10, 13, 20] is one of the fields greatly related to fractal deformation. Among the studies categorized into fractal interpolation, the studies in [17, 18, 19] proposed a method that combined IFS and smooth interpolation techniques and achieved fractal deformation. A set of basis functions of smooth interpolation, such as Bézier interpolation, can be represented as the IFS attractor of a set of specific Markov matrices. Based on this fact, using general Markov matrices results in basis functions that interpolate control points into a curve, a tensor product surface, or an IFS attractor defined on a tensor product parameter space, each of which is given fractal appearance. Controlling the Markov matrices and control points achieves fractal deformations of these shapes. In [7, 8], we proposed a construction method of a fractal interpolation surface, which was referred to as *wrinkly surface (WR surface)*. We first defined the transformation referred to as Iterated Shuffle Transformation (IST) and then proposed a method for interpolating control points having height values into a WR surface using the IST. Changing the positions and the height values of the control points results in giving a fractal deformation, which is referred to as *avalanche deformation*, to the WR surface.

The techniques in [4, 5, 15] utilize the fundamental mechanism of an IFS attractor so well and deform its shape globally. However, these techniques cannot achieve a local deformation because each contraction mapping of the IFS affects the whole shape. Similarly, the technique in [14] can achieve only a global deformation because of the global influence of a functional equation. Moreover, these techniques are manipulated by treating parameter values such as the coefficients of contraction mappings or the strength of attraction of fixed points, and a user cannot control a shape directly. Thus some shapes are difficult to deform intuitively. The technique in [12] is restricted to a moving tree. The method in [17, 18, 19] is formulated quite well mathematically. In this method, when a control point is moved, the shape produced by interpolation is deformed locally according to the movement of the control point in the fractal manner determined by the Markov matrices. However, it is not so easy to obtain a desired deformation, because controlling the Markov matrices is not intuitive and the movement of the control point affects both the position and the fractal appearance of the shape simultaneously. The method in [7, 8] can deform the shape of a surface locally by manipulating the position and the height value of a control

point separately in a intuitive way. However, the application of this method is restricted to a surface shape.

In this paper, we expand the idea of [7, 8] to any geometric shape that can be addressed appropriately. The proposed technique is realized by separating an original shape to deform and displacement vectors given on the shape for a fractal deformation. This means that an original shape and additional information for a deformation are controlled separately in the proposed technique whereas both of them are treated together in a construction mechanism of a fractal shape in most of other techniques. As a result, the proposed technique enables easy local control of the fractal deformation by manipulating the displacement vectors intuitively. Moreover, the extension of unit-IST's to ext-unit-IST's enhances the variety of the deformation and achieves combining a continuous deformation and a fractal deformation continuously. The above effects distinctive of the proposed technique cannot be realized by other fractal deformation techniques.

3 Extended Unit Iterated Shuffle Transformation

In this section, we define ext-unit-IST's. First, let Σ_L denote the code space of L symbols, $L \geq 1$, given by

$$\Sigma_L = \{\alpha = \alpha_1\alpha_2\alpha_3 \dots \mid \alpha_j \in \mathbf{Z}_L, j = 1, 2, 3, \dots\}, \quad (1)$$

where $\mathbf{Z}_L = \{0, 1, \dots, L-1\}$. The space Σ_L is referred to as *unit code space of L symbols*. For example, we can take $\alpha = 24101\dots$ as a code on Σ_5 . Then, we define an ext-unit-IST on Σ_L as follows.

Definition 1. For $\alpha \in \Sigma_L$, let $D_{e-u}^L : \Sigma_L \rightarrow \Sigma_L$ for integers $k_b \geq 0$ and $k \geq k_b + 1$ be the mapping given by

$$\alpha = \alpha_1 \dots \alpha_{k_b} \alpha_{k_b+1} \dots \alpha_{k-1} \alpha_k \alpha_{k+1} \dots, \quad (2)$$

$$D_{e-u}^L(k_b, k, \alpha) = \alpha_1 \dots \alpha_{k_b} \alpha_k \alpha_{k_b+1} \dots \alpha_{k-1} \alpha_{k+1} \dots. \quad (3)$$

When $k_b = 0$, symbols $\alpha_1, \dots, \alpha_{k_b}$ are omitted from the equations.

Definition 2. An ext-unit-IST is defined as the mapping $D_{e-uni}^L : \Sigma_L \rightarrow \Sigma_L$ for integers k_b and k_e , $0 \leq k_b \leq k_e$, given by

$$D_{e-uni}^L(k_b, k_e, \alpha) = \begin{cases} \alpha & \text{if } k_b = k_e, \\ D_{e-u}^L(k_b, k_e, D_{e-uni}^L(k_b, k_e - 1, \alpha)) & \text{if } k_b < k_e. \end{cases} \quad (4)$$

The introduction of variable k_b is the extended point from the original unit-IST we have proposed in [7, 8]. The original unit-IST is a special case of the ext-unit-IST and is defined by setting $k_b = 0$. In Sec.5.2, we will

explain how the introduction of k_b works on geometric shapes using the examples in Fig.3.

We can obtain the following theorems.

Theorem 1. For fixed k_b and k_e , the mapping D_{e-uni}^L is one-to-one and onto.

Proof. Consider D_{e-u}^L . From Equations 2 and 3, if k_b and k are fixed, then D_{e-u}^L is one-to-one and onto for $\alpha \in \Sigma_L$. Now, consider D_{e-uni}^L . From Eq.4, if $k_b = k_e$, then this theorem is true. If $k_b < k_e$, then D_{e-uni}^L is defined as the composite mapping of a series of D_{e-u}^L , $k = k_b + 1, \dots, k_e - 1, k_e$, as follows:

$$\begin{aligned} & D_{e-uni}^L(k_b, k_e, \alpha) \\ &= D_{e-u}^L(k_b, k_e, D_{e-u}^L(k_b, k_e - 1, D_{e-u}^L(\dots \\ & \quad \dots D_{e-u}^L(k_b, k_b + 1, D_{e-uni}^L(k_b, k_b, \alpha)) \dots))) \\ &= D_{e-u}^L(k_b, k_e, D_{e-u}^L(k_b, k_e - 1, D_{e-u}^L(\dots \\ & \quad \dots D_{e-u}^L(k_b, k_b + 1, \alpha)) \dots))). \end{aligned}$$

Each mapping D_{e-u}^L is one-to-one and onto, as proved above. Therefore, the mapping D_{e-uni}^L is also clearly one-to-one and onto. ■

Theorem 2. If $k_b < k_e$, then the mapping D_{e-uni}^L is formulated by

$$\alpha = \alpha_1 \dots \alpha_{k_b} \alpha_{k_b+1} \alpha_{k_b+2} \dots \alpha_{k_e-1} \alpha_{k_e} \alpha_{k_e+1} \alpha_{k_e+2} \dots \quad (5)$$

$$\begin{aligned} & D_{e-uni}^L(k_b, k_e, \alpha) = \\ & \alpha_1 \dots \alpha_{k_b} \alpha_{k_e} \alpha_{k_e-1} \dots \alpha_{k_b+2} \alpha_{k_b+1} \alpha_{k_e+1} \alpha_{k_e+2} \dots \end{aligned} \quad (6)$$

When $k_b = 0$, symbols $\alpha_1, \dots, \alpha_{k_b}$ are omitted from the equations.

Proof. Equations 2, 3, and 4 complete the proof. ■

Theorem 3. For $\forall \alpha \in \Sigma_L$, if $\alpha' = D_{e-uni}^L(k_b, k_e, \alpha)$ then $\alpha = D_{e-uni}^L(k_b, k_e, \alpha')$.

Proof. The case of $k_b = k_e$ is trivial from Eq.4. If $k_b < k_e$, then Equations 5 and 6 complete the proof. ■

4 Local Resemblance in Space/Scale Directions

We have two types of directions on a geometric space: *space directions* and *scale directions*. The former are the directions along which a point of view changes, whereas the latter are the directions along which a field of view changes. Based on these directions, we can consider “locality in space directions” and “self-similarity in scale directions”. The former is the property that each local position on the space can be identified uniquely. The latter is the property that a similar structure appears repeatedly by expansion or reduction. “Local resemblance in space/scale directions”,

which we have proposed[7, 8], is the property that unifies the “locality” and “self-similarity”. An ext-unit-IST, as well as an original unit-IST, constructs this property on a geometric space as shown in the next section.

5 Fractal Deformation

In this section, we propose the fractal deformation technique using displacement vectors based on ext-unit-IST’s and show various examples. We here demonstrate only the cases of two-dimensional IFS attractors, although arbitrary dimension and other kinds of shapes can be essentially treated as well.

5.1 Addressing Points on Geometric Shape

In order to apply the proposed technique to a geometric shape S , we have to define an ext-unit-IST working on the shape. For this definition, first, we have to address the points on the shape appropriately. Concretely, each point $s \in S$ has to be uniquely given a code $\alpha \in \Sigma_L$ as its address. This is achieved by a one-to-one and onto mapping $M : S \rightarrow \Sigma_L$, which is referred to as *address mapping*.

$$M(s) = \alpha. \quad (7)$$

In general, each point s on the shape S is given various attributes such as a coordinate \mathbf{x} that determines the position of the point s in the space \mathbf{X} in which the shape S is defined. Here, for a convenient description, let $F_a : S \rightarrow Atr$ be an *attribute function* that gives an attribute value $a \in Atr$ to a point $s \in S$.

$$F_a(s) = a. \quad (8)$$

For example, a *coordinate function* $F_{\mathbf{x}} : S \rightarrow \mathbf{X}$ gives a coordinate $\mathbf{x} \in \mathbf{X}$ to a point $s \in S$.

$$F_{\mathbf{x}}(s) = \mathbf{x}. \quad (9)$$

We are allowed to construct different address mappings M for a geometric shape S . For example, when S is a D -dimensional unit cube, that is, $F_{\mathbf{x}}(S) = \{\mathbf{x} = (x_1, x_2, \dots, x_D) \mid x_i \in [0, 1], i = 1, 2, \dots, D\} \subset \mathbf{R}^D$, $D \geq 1$, we can easily construct M as shown in Fig.1. (a) and (b) are one-dimensional cases for $L = 2$ and 3. (c) and (d) are two-dimensional cases for $L = 4$ and 9. Each figure illustrates how the addresses α of points s are determined by the mapping M , where the first and second places of α given to each region of the unit cube are shown. This addressing rule proceeds to infinity to give α infinite places.

When S is an IFS attractor, we can use the idea described in [3]. *Iterated Function System (IFS)*[2, 3, 6]

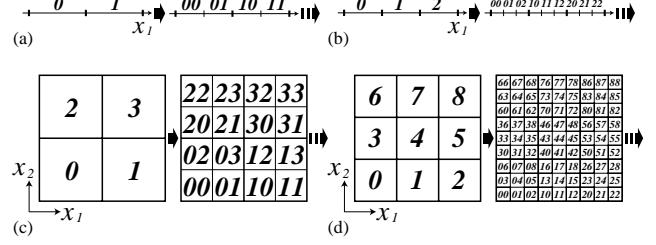


Figure 1: Address mappings on unit cubes.

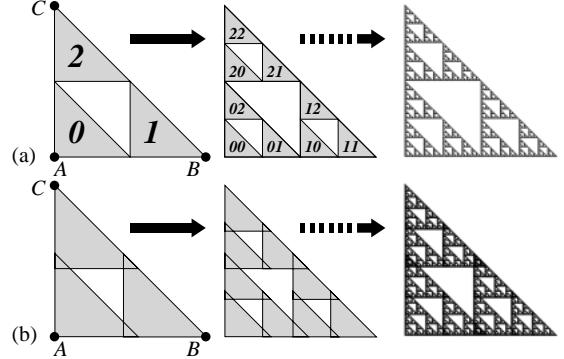


Figure 2: Address mappings on IFS attractors.

is a well-known framework to generate fractals. A (hyperbolic) IFS consists of a complete metric space \mathbf{X} and a finite set of contraction mappings $w_i : \mathbf{X} \rightarrow \mathbf{X}$, $i = 0, \dots, L - 1$. Its notation is $\{\mathbf{X}; w_i, i = 0, \dots, L - 1\}$. The set $S \subset \mathbf{X}$ is referred to as the attractor of the IFS if $S = \bigcup_{i=0}^{L-1} w_i(S)$. Each point $s \in S$ is given an address determined in terms of the sequence of mappings w_i applied. Figure 2 shows the cases of Sierpinski gaskets, $L = 3$, on \mathbf{R}^2 . The gasket of (a) is obtained as the attractor by the following IFS mappings:

$$\begin{aligned} w_0(\mathbf{x}) &= \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \\ w_1(\mathbf{x}) &= \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0.5 \\ 0 \end{bmatrix}, \\ w_2(\mathbf{x}) &= \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0.5 \end{bmatrix}, \end{aligned}$$

and that of (b) is obtained by the following:

$$\begin{aligned} w_0(\mathbf{x}) &= \begin{bmatrix} 0.55 & 0 \\ 0 & 0.55 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \\ w_1(\mathbf{x}) &= \begin{bmatrix} 0.55 & 0 \\ 0 & 0.55 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0.45 \\ 0 \end{bmatrix}, \\ w_2(\mathbf{x}) &= \begin{bmatrix} 0.55 & 0 \\ 0 & 0.55 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0.45 \end{bmatrix}. \end{aligned}$$

In each of (a) and (b), the two figures on the left show the initial two application steps of L contraction map-

pings from the initial triangle ABC , where the numbers show the addressing rule in the same way as Fig.1. The figures on the right are the resultant attractors after infinite steps.

Strictly, in [3], a point can be given more than one address. In [3], a “point” means its “coordinate”, and addresses are directly given to the coordinates $\mathbf{x} \in \mathbf{X}$ of points on an attractor. Therefore, if the areas mapped by its mappings w_i overlap on \mathbf{X} , then a point, that is, a coordinate on the attractor can be given multiple addresses, as shown in Fig.2 (b)^{†1}. On the other hand, in this paper, a point s and its coordinate \mathbf{x} are distinguished. Every point to which a different sequence of mappings w_i is applied is considered as a different point, even if the point overlaps with others. And its coordinate is considered as an attribute of the point.

5.2 Ext-unit-IST on Geometric Shape

Using the mappings D_{e-uni}^L in Eq.4 and M in Eq.7, we can define an ext-unit-IST working on a geometric shape S as the mapping $F_{e-uni}^L : S \rightarrow S$ for integers k_b and k_e given by

$$F_{e-uni}^L(k_b, k_e, s) = M^{-1}(D_{e-uni}^L(k_b, k_e, M(s))). \quad (10)$$

The mapping F_{e-uni}^L is also one-to-one and onto.

Attribute values a given to points s by the function F_a in Eq.8 are transformed on the shape S by the application of the ext-unit-IST. Using Eq.10, the resultant function $F_{e-uni,a}^L : S \rightarrow Atr$ is expressed as follows.

$$F_{e-uni,a}^L(k_b, k_e, s) = F_a(F_{e-uni}^{L-1}(k_b, k_e, s)). \quad (11)$$

Moreover, Eq.11 can be *superposed* as follows.

$$\begin{aligned} & F_{S-e-uni,a}^L(k_b, k_e, s) \\ &= \sum_{k=k_b}^{k_e} W_a(k, k_b, k_e, s) F_{e-uni,a}^L(k_b, k, s). \end{aligned} \quad (12)$$

The *weight function* W_a , $k_b \leq k \leq k_e$, is defined by using a *ratio function* $\delta_a(s) \geq 0$ ^{†2} as follows.

$$\begin{aligned} & W_a(k, k_b, k_e, s) \\ &= \begin{cases} \frac{1 - \delta_a(s)}{1 - \{\delta_a(s)\}^{k_e - k_b + 1}} \{\delta_a(s)\}^{k - k_b} & \text{if } \delta_a(s) \neq 1, \\ \frac{1}{k_e - k_b + 1} & \text{if } \delta_a(s) = 1. \end{cases} \end{aligned} \quad (13)$$

Equation 13 satisfies $\sum_{k=k_b}^{k_e} W_a(k, k_b, k_e, s) \equiv 1$. The weight function and ratio function defined above are given a point s as an argument. This means that the

^{†1}Actually, even in (a), the points that joint triangles have multiple addresses when infinite places of α are considered[3].

^{†2}In [7, 8], this range was defined as $0 < \delta_a(s) < 1$.

values of these functions are variable for different points s on the shape S .

Figure 3 shows examples for demonstrating how ext-unit-IST’s work on geometric shapes. These examples help readers understand the difference between the transformation patterns caused by different pairs of k_b and k_e in Eq.10. The geometric shape and applied address mapping of (a) are Fig.1 (c), and those of (b) are Fig.2 (a). The points s on each shape are given color values $\mathbf{c} = (red, green, blue) \in \mathbf{C}$ by a *color function* $F_c : S \rightarrow \mathbf{C}$ as attribute values a . In each of (a) and (b) in Fig.3, the image in the row labeled k_b and the column labeled k_e is colored by $F_{e-uni,c}^L$, which is the function obtained by replacing a with \mathbf{c} in Eq.11, given the values k_b and k_e . Similarly, each image in the column labeled “superposition” is colored by $F_{S-e-uni,c}^L$, which is obtained using Eq.12, given k_b of the row, $k_e = 5$, and the ratio function $\delta_c(s) = 1$ (constant for all s). We find that the color changes in these images caused by the ext-unit-IST’s construct fractal-like repeated structures. Here, we observe how different values of k_e work. In each of (a) and (b), for each k_b , the construction of the repeated structure proceeds in the scale direction by scattering the colors over the shape as k_e increases. That is, k_e works for controlling the level of detail of the repeated structure. Moreover, even though k_e increases, the upper left part on the shape remains redder, the lower right part remains bluer, and the lower left part or upper right part remains greener than other parts. That is, the proportion of red, blue, and green depends on the position in the space direction. The “superposition” case shows this situation more clearly. The property described above is the “local resemblance in space/scale directions” constructed by the ext-unit-IST’s. Next, we observe how different values of k_b work. As described in Sec.3, the introduction of k_b is the extended point from original unit-IST’s to ext-unit-IST’s proposed in this paper. In each of (a) and (b), the case of $k_b = 0$ shows an original unit-IST. When comparing the cases of $k_b = 0, 1$, and 2 , we find that the scattering of the colors is localized within smaller regions as k_b becomes greater. Actually, k_b works for restricting the extent within which a point s can be transformed by Eq.10. In Fig.3, for each k_b , the leftmost image shows that the transformation of a point s is restricted within the region that is enclosed by a white square or triangle and includes the point. This property realized by k_b is an important feature distinctive of ext-unit-IST’s and cannot be achieved by original unit-IST’s. In particular, this property is indispensable for realizing a continuous transition between a continuous deformation and a fractal deformation described in Sec.5.4.

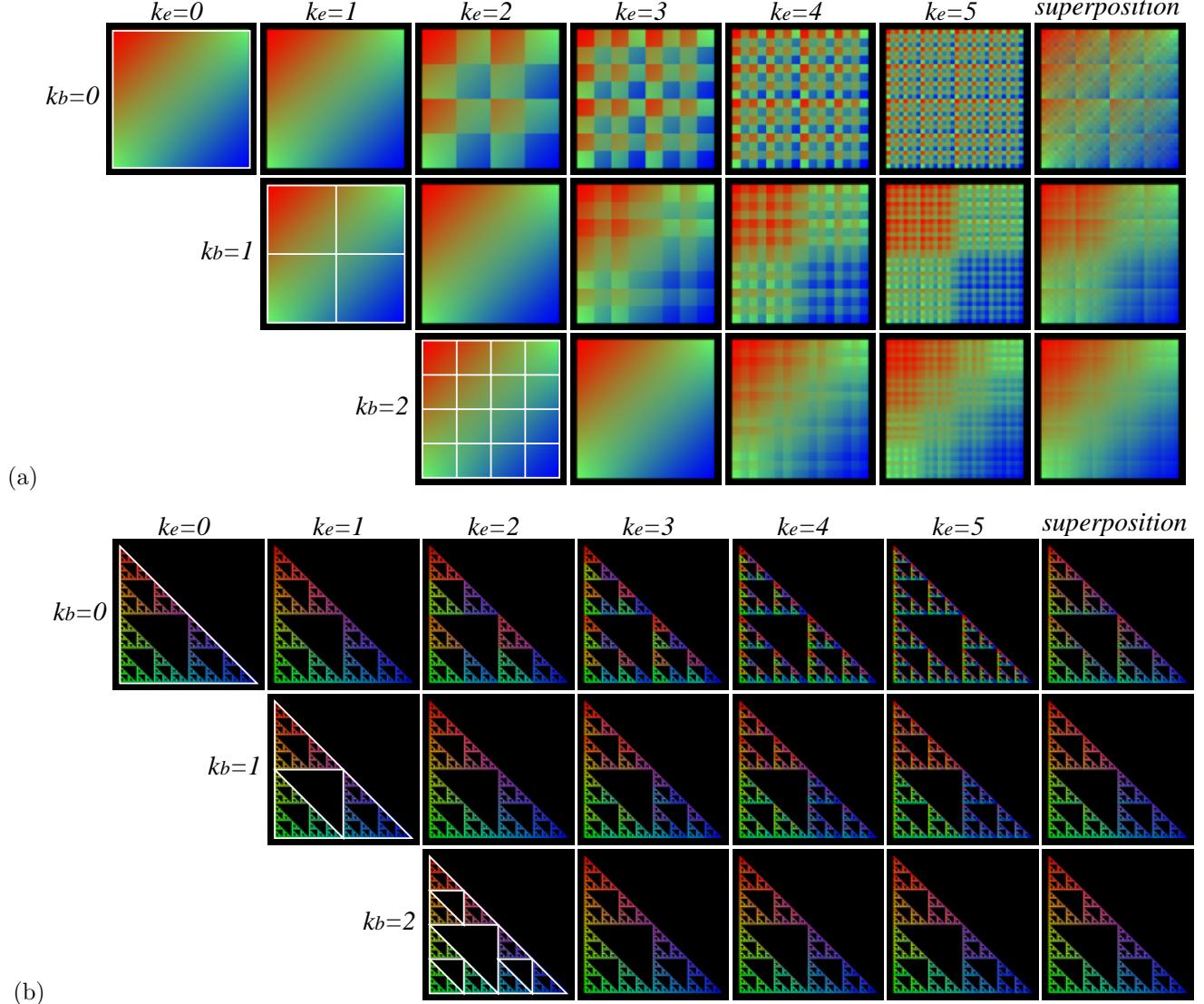


Figure 3: Examples of ext-unit-IST's on geometric shapes. (a) A unit cube. (b) A Sierpinski gasket.

5.3 Fractal Deformation Based on Ext-unit-IST

The fractal deformation proposed in this paper is achieved by giving displacement vectors to the points on a geometric shape to deform and applying an ext-unit-IST to the displacement vectors. This is formalized as follows.

First, let $F_{\mathbf{x}}$ in Eq.9 be the coordinate function of a geometric shape S to deform. Actually, the shape S is concretely determined on the coordinate space \mathbf{X} by defining the function $F_{\mathbf{x}}$. Then, each point s on the shape S is given a displacement vector $\mathbf{v} \in \mathbf{X}$. When the displacement vector is considered to be an attribute value, the operation of giving displacement vectors \mathbf{v}

to the points s is represented as a *displacement vector function* $F_{\mathbf{v}} : S \rightarrow \mathbf{X}$ given by

$$F_{\mathbf{v}}(s) = \mathbf{v}. \quad (14)$$

When the displacement vectors varies over time $t \in T$, Eq.14 is represented as the function $F_{\mathbf{v}} : S, T \rightarrow \mathbf{X}$ given by

$$F_{\mathbf{v}}(s, t) = \mathbf{v}. \quad (15)$$

This function form is used for producing a deformation over time t . In the following, for convenience, we consider only functions $F_{\mathbf{v}}$ that are continuous on both S and T . Here, if the functions $F_{\mathbf{x}}$ and $F_{\mathbf{v}}$ are directly added, the coordinate of a point s on the space \mathbf{X} at time t is obtained by the function $P_{\mathbf{x}, \mathbf{v}} : S, T \rightarrow \mathbf{X}$

given by

$$P_{\mathbf{x}, \mathbf{v}}(s, t) = F_{\mathbf{x}}(s) + F_{\mathbf{v}}(s, t). \quad (16)$$

In this case, the shape S is deformed continuously over time t because the function $F_{\mathbf{v}}$ is continuous on both S and T as described above. In this paper, we refer to this deformation as *continuous deformation* in comparison with the term “fractal deformation”. On the other hand, using Eq.11^{†3}, applying an ext-unit-IST to the function $F_{\mathbf{v}}$ results in the fractal deformation proposed in this paper. In this case, the coordinate of a point s at time t is obtained by the function $P_{e-uni, \mathbf{x}, \mathbf{v}}^L : S, T \rightarrow \mathbf{X}$ given by

$$\begin{aligned} P_{e-uni, \mathbf{x}, \mathbf{v}}^L(k_b, k_e, s, t) \\ = F_{\mathbf{x}}(s) + F_{\mathbf{v}}(F_{e-uni}^{L-1}(k_b, k_e, s), t) \\ = F_{\mathbf{x}}(s) + F_{e-uni, \mathbf{v}}^L(k_b, k_e, s, t). \end{aligned} \quad (17)$$

In Eq.17, the function F_{e-uni}^L rearranges the displacement vectors given by $F_{\mathbf{v}}$ on the shape S in a local resemblance manner, as shown in Sec.5.2. This results in a fractal deformation. Besides, using Eq.12, the superposed case is obtained by the function $P_{S-e-uni, \mathbf{x}, \mathbf{v}}^L : S, T \rightarrow \mathbf{X}$ given by

$$\begin{aligned} P_{S-e-uni, \mathbf{x}, \mathbf{v}}^L(k_b, k_e, s, t) \\ = F_{\mathbf{x}}(s) + \sum_{k=k_b}^{k_e} W_{\mathbf{v}}(k, k_b, k_e, s) F_{e-uni, \mathbf{v}}^L(k_b, k, s, t) \\ = F_{\mathbf{x}}(s) + F_{S-e-uni, \mathbf{v}}^L(k_b, k_e, s, t). \end{aligned} \quad (18)$$

The weight function $W_{\mathbf{v}}$ for displacement vectors \mathbf{v} is obtained using a ratio function $\delta_{\mathbf{v}}(s)$ for \mathbf{v} and Eq.13.

We show some examples of the fractal deformation in the following. These examples are two-dimensional IFS attractors defined on \mathbf{R}^2 . Therefore, the equations above are treated by considering that $\mathbf{X} = \mathbf{R}^2$, $\mathbf{x} = (x, y) \in \mathbf{R}^2$, and $\mathbf{v} = (v_x, v_y) \in \mathbf{R}^2$.

(The animations included in the supplementary works help readers understand the following explanation.)

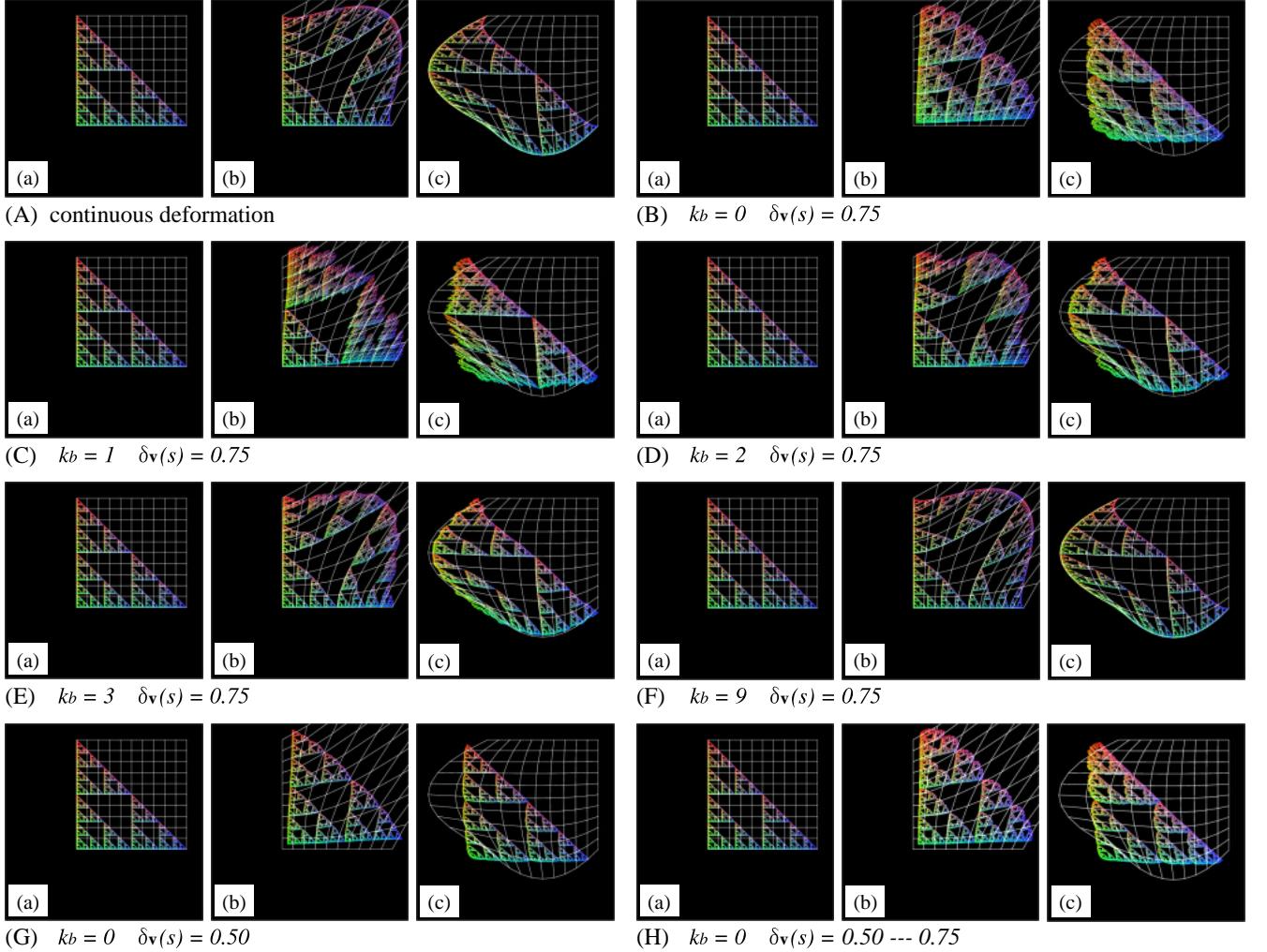
(1) Examples of a Sierpinski Gasket (Fig.4)

Figure 4 shows examples for explaining how the fractal deformation technique works on a Sierpinski gasket. *(See the animations of Figure 4 in the supplementary works.)* (A) is an example of continuous deformation obtained by Eq.16, while (B) to (H) are examples of fractal deformation obtained by Eq.18. (B) to (H) are given different pairs of k_b and $\delta_{\mathbf{v}}$, although all of them are given the same $k_e = 9$ that determines

^{†3}Time t is considered in Eq.17, although Eq.11 does not have t . The same applies to Equations 18 and 12.

the minimum scale of the deformations. In each case from (A) to (H), (a) is the original Sierpinski gasket by setting $F_{\mathbf{v}}(s, t) = (0, 0)$ for all s at $t = t_0 = 0$, while (b) and (c) are deformed gaskets by $F_{\mathbf{v}}(s, t)$ at $t = t_1$ and t_2 ($t_0 < t_1 < t_2$). All the cases from (A) to (H) are given the same function $F_{\mathbf{v}}$. Technically, in these examples, the function $F_{\mathbf{v}}$ was defined by giving displacement vectors changing over time t to 4×4 control points constituting a control mesh placed over the gasket and applying two-dimensional Bézier interpolation of degree 3 to the vectors. (The functions $F_{\mathbf{v}}$ used for Figures 5 and 6 were also defined using the same technique.) This means that actually the function $F_{\mathbf{v}}$ was defined as not $F_{\mathbf{v}}(s, t)$ but $F_{\mathbf{v}}(\mathbf{x}, t)$ for $\mathbf{x} \in \mathbf{Y} \subset \mathbf{X}$, where \mathbf{Y} is the square region covered with the Bézier interpolation on \mathbf{X} . Then, the coordinate $\mathbf{x} = F_{\mathbf{x}}(s)$ of each point s on the gasket was assigned the displacement vector $\mathbf{v} = F_{\mathbf{v}}(\mathbf{x}, t)$. The movement of the white grid drawn over the gasket shown in (a), (b), and (c) represents the resultant function $F_{\mathbf{v}}$ changing over time t . The movements of the grid points of this white grid correspond to $F_{\mathbf{v}}(\mathbf{x}, t)$ obtained at 11×11 points \mathbf{x} placed on \mathbf{Y} at equal parameter intervals. The position of each grid point indicates $\mathbf{x} + F_{\mathbf{v}}(\mathbf{x}, t)$ at t .

Comparing (A) and (B) shows us the fundamental difference between a continuous deformation and a fractal deformation. The continuous deformation (A) deforms the whole shape of the gasket continuously, while the fractal deformation (B) deforms the gasket in the way that each subpart of the gasket in all scales is deformed recursively (*see the animations in the supplementary works carefully*). Here, for detailed analyses, we focus on the deformed shape of (b) in (A) and that in (B). In (A)-(b), when the midpoint of the hypotenuse of the gasket is lifted up, the lifting is propagated continuously around the lifted point and produces the continuous deformation. The neighbourhood of the lifted point is deformed greater, and the regions far from the point are less deformed. The region of the lower left corner is hardly deformed. On the other hand, in (B)-(b), the lifting is scattered all over the gasket in a fractal-like repeated manner. We find that every triangle in all regions, even in the lower left corner, and in all scales is deformed although the triangles near the lifted point are still deformed greater than those in far regions. This deformation manner indicates the property of local resemblance (cf. Sections 4 and 5.2). The comparison between (A) and (B) above shows that the two deformations have completely different deformation styles each other. In general, other fractal deformation techniques such as those described in Sec.2 also deform shapes in such kinds of recursive manners; such deformations cannot be achieved using

Figure 4: Fractal deformations of a Sierpinski gasket. $k_e = 9$.

▷ The animations of this figure are in the supplementary works.

continuous deformation techniques such as FFD. However, the proposed technique provides easy control to the deformation by changing the displacement vector function intuitively, while other techniques are not so easy to manipulate for the reasons described in Sec.2. Concretely, if using the above-mentioned technique of defining a displacement vector function, the fractal deformation can be easily controlled by manipulating the displacement vectors given to the 4×4 control points intuitively.

The examples from (B) to (F) show us the influence of different values of k_b on the deformations. The effect produced by k_b is distinctive of the proposed technique and is not found in other previous techniques. In the examples, the change of k_b changes the proportion of the deformations of the triangles in all regions and all scales inside the gasket (see the animations in the sup-

plementary works carefully). For example, when considering the deformed shapes of (b) in (B) to (F), as k_b increases from 0 to 9 (= k_e), the deformations of the triangles near the lifted point get to be greater than those in far regions. This means that the scattering of the lifting gets to be localized into each original position on the gasket. When k_b reaches k_e , this localization converges; the fractal deformations by Eq.18 become the continuous deformation by Eq.16. This can be easily understood by Eq.4 in Definition 2; an ext-unit-IST gives no change to α if $k_b = k_e$. Actually, we find that the deformed gasket in (F) is the same as that in (A). This fact means that ext-unit-IST's enable a continuous transition between a continuous deformation and a fractal deformation by changing k_b continuously (cf. Sec.5.4).

The examples of (B), (G), and (H) show us the in-

fluence of different ratio functions δ_v . The function δ_v controls the “roughness” or “smoothness” of a deformed shape. (B) and (G) are given $\delta_v(s) = 0.75$ and 0.50, constant for all points s on the gasket, respectively. Comparing these two cases shows that the deformed shape becomes rougher as $\delta_v(s)$ becomes greater. On the other hand, (H) is given $\delta_v(s)$ that has different values at different points s : given 0.50 on the lower left part of the gasket, 0.67 on the lower right part, and 0.75 on the upper left part. In this case, the deformed shape has different roughness on each part.

In fact, using Eq.17 or Eq.18 as it is, the resultant deformed shape comes apart because an ext-unit-IST on a geometric space is essentially a discontinuous function. For example, the color figures in Fig.3 reveal the discontinuity as the recursively partitioned square or triangle regions. For avoiding the discontinuity, when we use Eq.17 or Eq.18, we take the following strategy: 1) average the displacement vectors, which are obtained by $F_{e-uni,v}^L$, of all the points s in each continuous region that is not separated by the ext-unit-IST, 2) give each point s in the whole shape S a displacement vector obtained by interpolating the averaged vectors in 1) using Delaunay triangulation.

(2) Examples of a Black Spleenwort Fern (Fig.5)

Figure 5 shows another example of IFS attractor, a Black Spleenwort fern, $L = 4$, on \mathbf{R}^2 . (*See the animations of Figure 5 in the supplementary works.*) This attractor is obtained by the following IFS mappings:

$$\begin{aligned} w_0(\mathbf{x}) &= \begin{bmatrix} 0.85 & 0.04 \\ -0.04 & 0.85 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1.6 \end{bmatrix}, \\ w_1(\mathbf{x}) &= \begin{bmatrix} 0.2 & -0.26 \\ 0.23 & 0.22 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1.6 \end{bmatrix}, \\ w_2(\mathbf{x}) &= \begin{bmatrix} -0.15 & 0.28 \\ 0.26 & 0.24 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0.44 \end{bmatrix}, \\ w_3(\mathbf{x}) &= \begin{bmatrix} 0 & 0 \\ 0 & 0.16 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \end{aligned}$$

(X) indicates its addressing rule. (A) is an example of continuous deformation by Eq.16. (B) to (E) are examples of fractal deformation by Eq.18 given different pairs of k_b and δ_v , and the same $k_e = 6$. (A) to (E) are given the same displacement vector function F_v . (B-1) to (B-3) are magnified images of (B); the magnified parts are enclosed with red, blue, and green rectangles.

In the fractal deformations of this model, in order to give a proper sway direction to each part of the fern, the displacement vector given a point s on the fern is rotated properly; the displacement vector $F_{e-uni,v}^L(k_b, k, s, t)$ in Eq.18 given on a point s is ro-

tated by the rotation factors of the IFS mappings from $w_{\alpha_{k_b+1}}$ to w_{α_k} , where $\alpha = M(s)$ by Eq.7. As a result of this rotation as well as the scattering of the displacement vectors by the ext-unit-IST’s, in (B) to (E), every branch or leaf element in all regions and all scales is deformed in its own manner having a proper sway direction. The displacement vector function F_v given to this model originally sways the upper part of the fern from side to side, as shown in (A). However, when observing (B), (B-1), (B-2), and (B-3), we find that the branch or leaf elements in each subpart of the fern are swaying to the right and left of the axis of the subpart as if the subpart were a single fern in itself (*see the animations in the supplementary works carefully*). This sway is not identical with the motion of the white grid representing the original motion of F_v . The important fact is that the same situation above is observed in every subpart of the fern in all regions and all scales recursively. In comparison, in the continuous deformation (A), the deformation of the whole shape is identical with the motion of the white grid; this results in a rigid impression of the whole shape. The above distinctive deformation manner of the proposed technique cannot be realized by other previous fractal deformation techniques as well as usual continuous deformation techniques. Similarly to Fig.4, manipulating F_v intuitively achieves easy control of the deformation.

The examples (B) to (E) show the effect caused by different pairs of k_b and δ_v . As described above, the displacement vector function F_v originally sways the upper part of the fern from side to side, as shown in (A). When $k_b = 0$ in (B), the sway is scattered all over the fern, and even the lowest part sways widely. As k_b increases from (B) to (E), the sway of the lower part decreases and that of the upper part increases. When k_b reaches 6 (= k_e), the fractal deformations converge to the continuous deformation of (A). In the same way as Fig.4, this is caused by the fact that the value of k_b controls the extent of the scattering of the displacement vectors given by F_v . In this model, the ratio function δ_v is given smaller value as k_b increases in order to make the deformation gentle.

(3) Examples of a Twin-Dragon (Fig.6)

Figure 6 shows examples of a Twin-Dragon, $L = 2$, on \mathbf{R}^2 . (*See the animations of Figure 6 in the supplementary works.*) This attractor is obtained by the following IFS mappings:

$$\begin{aligned} w_0(\mathbf{x}) &= \begin{bmatrix} 0.5 & 0.5 \\ -0.5 & 0.5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} -0.5 \\ 0.375 \end{bmatrix}, \\ w_1(\mathbf{x}) &= \begin{bmatrix} 0.5 & 0.5 \\ -0.5 & 0.5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0.5 \\ 0.625 \end{bmatrix}. \end{aligned}$$

(X) indicates its addressing rule. (A) is an example of continuous deformation by Eq.16. (B) to (E) are examples of fractal deformation by Eq.18 given different values of k_b , and the same $k_e = 8$ and $\delta_v(s) = 1$ for all points s on the dragon. (A) to (E) are given the same displacement vector function F_v . (B-1) to (B-3) are magnified images of (B); the magnified parts are enclosed with red, black, and green rectangles. (Y) is the color function F_c (cf. Sec.5.2). This function was defined in the same way as the displacement vector function F_v by giving color values c to the 4×4 control points and interpolating them. All of (A) to (E) are colored by applying Eq.12 to this color function with $k_b = 0$, $k_e = 8$, and $\delta_c(s) = 1$ for all s . This makes it easy to understand the fractal motion inside the dragon described below.

The displacement vector function F_v originally affects the dragon to draw its lower right part to the lower right direction, as shown in (A). In the fractal deformations (B) to (E), the displacement vector given a point s on the dragon by the function F_v is rotated by the rotation factors of the IFS mappings in the same way as Fig.5. Thus each spiral tip in all regions and all scales rounds inside and shrinks individually. The magnified images (B-1) to (B-3) show that each inside part as well as the outline of the dragon is deformed individually in such a manner (*see the animations in the supplementary works carefully*). This deformation is distinctive of the proposed technique.

The examples (B) to (E) show the effect caused by different k_b . When k_b is small, the drawing of the lower right part by F_v gives great effect even on the opposite upper left part. The increase of k_b makes this effect localized to the neighbourhood of the drawn part; the lower right part is more deformed while the upper left part is less deformed. Then, the spiral deformation effect described above is weakened.

5.4 Continuous Transition between Continuous and Fractal Deformations

The function $F_{e-uni,a}^L$ in Eq.11 can be easily extended for non-integers k_b^* and k_e^* , $0 \leq k_b^* \leq k_e^*$, by

$$\begin{aligned} F_{e-uni,a}^{*L}(k_b^*, k_e^*, s) &= \\ \bar{r}_b \bar{r}_e F_{e-uni,a}^L(k_b, k_e, s) + \bar{r}_b r_e F_{e-uni,a}^L(k_b, k'_e, s) \\ + r_b \bar{r}_e F_{e-uni,a}^L(k'_b, k_e, s) + r_b r_e F_{e-uni,a}^L(k'_b, k'_e, s). \end{aligned} \quad (19)$$

In Eq.19, k_b and r_b are the integral part and decimal part of k_b^* ; $k'_b = k_b + 1$; $\bar{r}_b = 1 - r_b$. The same notations are given for k_e^* . Equation 19 realizes perfectly a continuous transition between a continuous deformation and a fractal deformation. This technique can be also applied to the case of Eq.12.

(The supplementary works include the animation of an example of continuous transition in the case of Figure 4.)

6 Conclusion and Future Work

In this paper, we have proposed the fractal deformation technique using displacement vectors based on ext-unit-IST's. Using this technique, a geometric shape can be deformed in a fractal-like repeated manner by controlling a displacement vector function intuitively. Besides, this technique realizes various levels of fractal deformations and achieves a continuous transition between a continuous deformation and a fractal deformation. Although we have demonstrated only the cases of two-dimensional IFS attractors, arbitrary dimension and other kinds of shapes can be treated if the shapes can be given proper address mappings. However, actually, when considering the application of this technique to actual various object shapes, it is hard to find proper address mappings in most cases. In addition, in order to realize desired deformations for various complicated shapes, a more flexible control mechanism is needed. These are open problems to solve. As one application theme, we are currently planning to apply this technique to image deformation using real pictures. Besides, we are pursuing other application themes of ext-unit-IST's and the proposition of extended connected-IST's.

References

- [1] Barnsley, M. F., Fractal Functions and Interpolation, *Constructive Approximation*, 2, pp.303–329, 1986.
- [2] Barnsley, M. F., Jacquin, A., Malassenet, F., Reuter, L., and Sloan, A. D., Harnessing Chaos for Image Synthesis, *Computer Graphics (SIGGRAPH '88)*, Vol.22, No.4, pp.131–140, 1988.
- [3] Barnsley, M. F., *Fractals Everywhere*, 2nd ed., Academic Press, Boston, 1993.
- [4] Bowman, R. L., Fractal Metamorphosis: a Brief Student Tutorial, *Computers & Graphics*, Vol.19, No.1, pp.157–164, 1995.
- [5] Burch, B. and Hart, J. C., Linear Fractal Shape Interpolation, *Graphics Interface '97*, pp.155–162, 1997.
- [6] Demko, S., Construction of Fractal Objects with Iterated Function Systems, *Computer Graphics (SIGGRAPH '85)*, Vol.19, No.3, pp.271–278, 1985.
- [7] Fujimoto, T. and Ohno, Y., Formalization and Superposed Construction of Wrinkly Surface, *Transactions of Information Processing Society of Japan*, Vol.41, No.9, pp.2518–2535, 2000 (in Japanese).

- [8] Fujimoto, T., Ohno, Y., Muraoka, K., and Chiba, N., Wrinkly Surface Generated on Irregular Mesh by Using IST Generalized on Code Space and Multi-Dimensional Space : Unification of Interpolation Surface and Fractal, *IEICE Transactions on Information and Systems*, to appear.
- [9] Fujimoto, T., Ohno, Y., Muraoka, K., and Chiba, N., Fractal Deformation Based on Extended Iterated Shuffle Transformation, *NICOGRAF International 2002*, pp.79–84, 2002.
- [10] Geronimo, J. S. and Hardin, D., Fractal Interpolation Surfaces and a Related 2-D Multiresolution Analysis, *Journal of Mathematical Analysis and Applications*, 176, pp.561–586, 1993.
- [11] Gomes, J., Darsa, L., Costa, B., and Velho, L., *Warping and Morphing of Graphical Objects*, Morgan Kaufmann, 1999.
- [12] Gonzalez, J. A., A Tutorial and Recipe for Moving Fractal Trees, *Computers & Graphics*, Vol.22, No.2–3, pp.301–305, 1998.
- [13] Massopust, P. R., Fractal Surfaces, *Journal of Mathematical Analysis and Applications*, 151, pp.275–290, 1990.
- [14] Montiel, M. E., Aguado, A. S., and Zaluska, E. J., Topology in Fractals, *Chaos, Solitons and Fractals*, Vol.7, No.8, pp.1187–1207, 1996.
- [15] Peruggia, M., *Discrete Iterated Function Systems*, A K Peters, 1993.
- [16] Sederberg, T. and Parry, S., Free-Form Deformation of Solid Geometric Models, *Computer Graphics (SIGGRAPH '86)*, Vol.20, No.4, pp.151–160, 1986.
- [17] Zair, C. E. and Tosan, E., Fractal Modeling Using Free Form Techniques, *EUROGRAPHICS '96*, Vol.15, No.3, pp.269–278, 1996.
- [18] Zair, C. E. and Tosan, E., Unified IFS-based Model to Generate Smooth or Fractal Forms, *Surface Fitting and Multiresolution Methods*, Vanderbilt University Press, pp.345–354, 1997.
- [19] Zair, C. E. and Tosan, E., Computer Aided Geometric Design with IFS Techniques, *Fractal Frontiers (Proc. Fractals '97)*, pp.443–452, 1997.
- [20] Zhao, N., Construction and Application of Fractal Interpolation Surfaces, *The Visual Computer*, 12, pp.132–146, 1996.

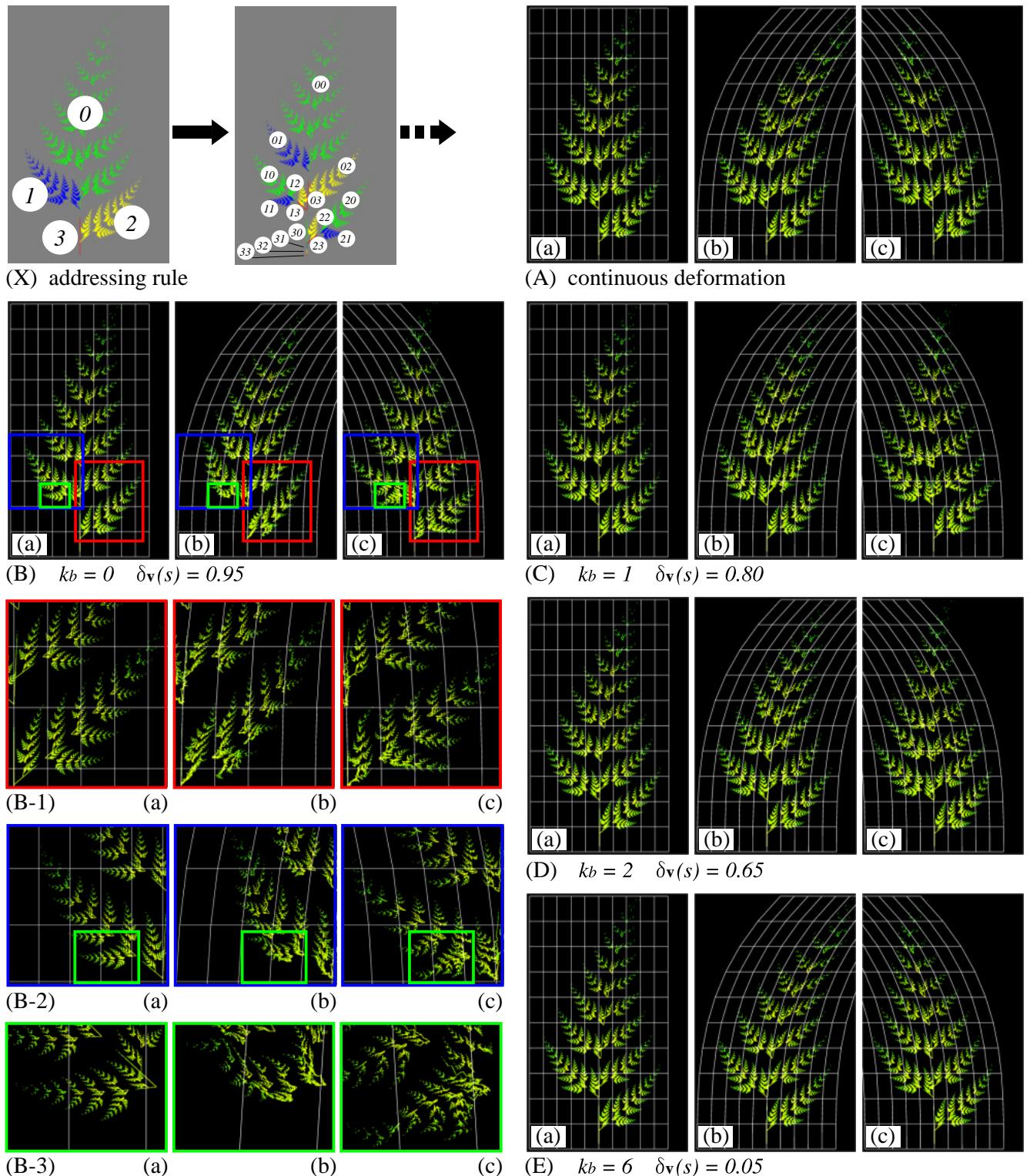


Figure 5: Fractal deformations of a Black Spleenwort fern. $k_e = 6$.
▷ The animations of this figure are in the supplementary works.

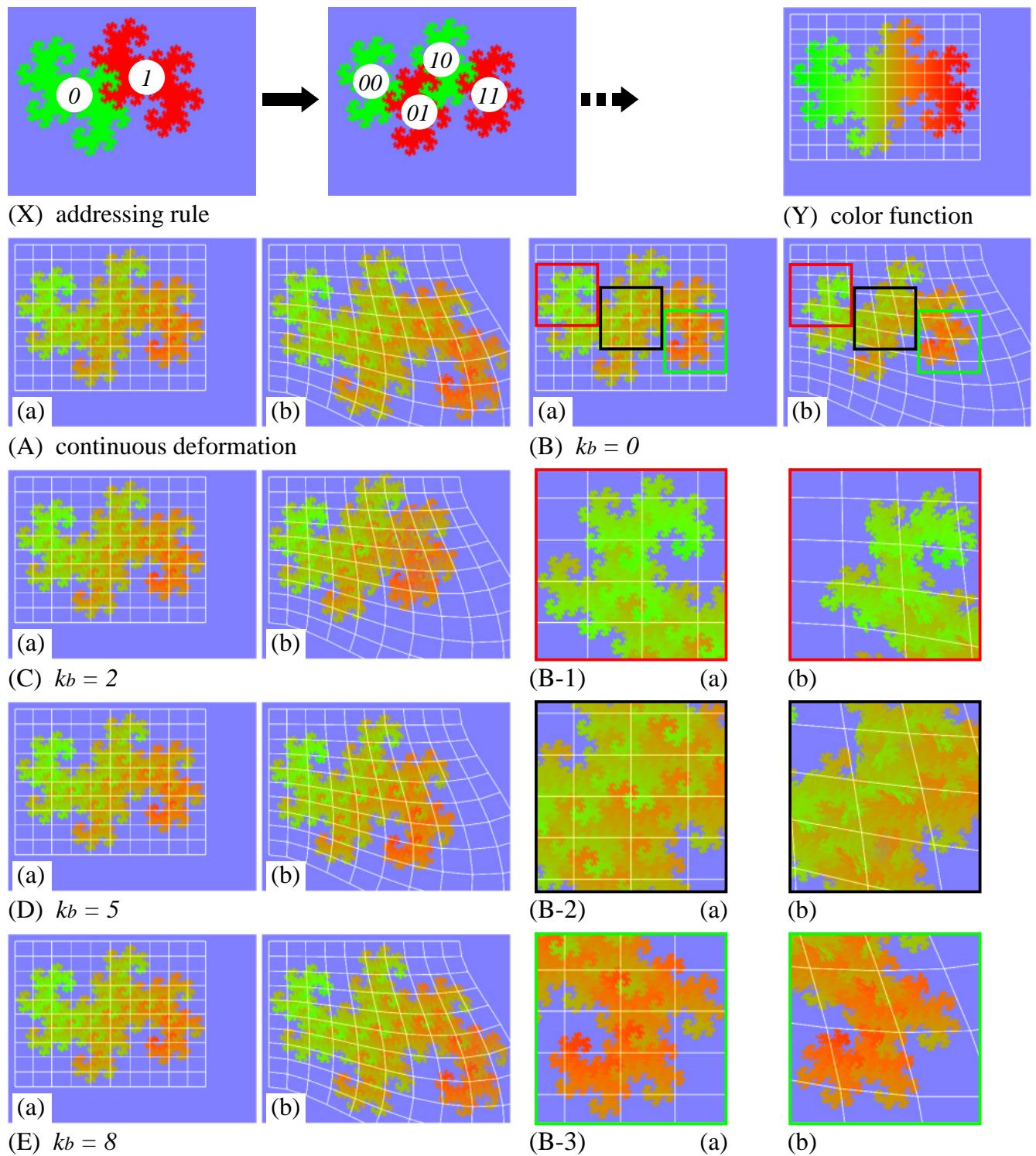


Figure 6: Fractal deformations of a Twin-Dragon. $k_e = 8$. $\delta_v(s) = 1$.
▷ The animations of this figure are in the supplementary works.