PAPER

Wrinkly Surface Generated on Irregular Mesh by Using IST Generalized on Code Space and Multi-Dimensional Space: Unification of Interpolation Surface and Fractal

Tadahiro FUJIMOTO[†], Yoshio OHNO[††], Kazunobu MURAOKA[†*], *and* Norishige CHIBA[†], *Regular Members*

SUMMARY Interpolation surfaces, such as Bézier or B-spline surface, are usually used for representing smooth man-made objects and provide an excellent ability to control the shape of a surface by intuitively moving control points. In contrast, the fractal technique is used for creating various complex shapes, mainly of natural objects, that have self-similarity using simple procedures. We have proposed the "wrinkly surface (WR surface)" for combining the advantages of interpolation surfaces and fractals. In this paper, we propose the expansion of the construction scheme of the WR surface to irregular meshes. Control points of a WR surface are interpolated using the "Iterated Shuffle Transformation (IST)." Therefore, in order to achieve the expansion, we first generalize the IST on code spaces, and then propose multi-dimensional IST defined on geometric spaces. By creating various shape model examples, we demonstrate the usefulness of the WR surface as a modeling tool.

key words: geometric modeling, interpolation surface, fractal, subdivision, iterated function system (IFS)

1. Introduction

Interpolation surfaces, such as Bézier or B-spline surface, are useful techniques to represent smooth man-made objects. The primary advantage of these techniques is to control the shape of a surface easily by intuitively moving control points. In contrast, the *fractal* technique is used to create self-similar complex shapes that appear primarily in natural objects. Fractals are pretty useful because they can be used to create such shapes automatically by simple procedures.

As an attractive geometric model, we have proposed the "*wrinkly surface* (*WR surface*)" [9]–[11], which is intended to be a unified model of the interpolation surface and the fractal. A WR surface is a parametric surface defined on a mesh of control points, and can manage complex self-similar shapes easily by intuitively moving control points. These control points are interpolated using the "*Iterated Shuffle Transformation* (*IST*)" [9]–[11]. An IST is a one-to-one and onto mapping that maps one point to another based on recursive subdivision in a fashion similar to the construction of the Cantor set [3], [13]. Thus, the WR surface may be considered to be a kind of subdivision surface, although the surface shape does not have a smooth appearance such as that of the usual subdivision surface [7]. The IST constructs the structure of *"local resemblance in space/scale direction,"* which unifies *"locality in space direction"* and *"self-similarity in scale direction"* (cf. Sect. 3.3). Since "locality" is the essence of the interpolation surface (cf. Sect. 3.5) and "self-similarity" is that of the fractal, "local resemblance" becomes the essential property for combining the interpolation surface and the fractal into the WR surface.

The WR surface has thus far been developed in two ways. One method was to define WR surfaces on regular meshes in tensor product form [9], [11], and the other method was to define them on irregular meshes by connecting tensor product patches in which the connection pattern was strongly restricted by a definite constrained condition [10]. These methods were all based on one-dimensional IST on geometric spaces. In this paper, we have two primary goals: 1) to generalize the IST on code spaces in abstractive form and propose multi-dimensional IST defined on geometric spaces, 2) to propose the construction scheme of the WR surface on irregular meshes by allowing arbitrary connection patterns of tensor product patches based on 1).

In Sect. 2, we discuss previous works. In Sect. 3, we propose the IST on code spaces and multi-dimensional geometric spaces. In Sect. 4, we propose the WR surface on irregular meshes, and in Sect. 5, we present conclusions.

2. Related Work

Several studies on unifying interpolation surfaces and fractals have been reported. This field of study is often referred to as *fractal interpolation surface.* Most research on this field is based on the theory of *Iterated Function System* (*IFS*) [1]–[3], [6]. Barnsley [1] proposed the fundamental concept of fractal interpolation based on the IFS. Massopust [14] studied fractal interpolation surfaces, and Geronimo, Hardin [12], Zhao [23] generalized them considering more general boundary

Manuscript received August 6, 2001.

Manuscript revised January 21, 2002.

[†]The authors are with the Faculty of Engineering, Iwate University, Morioka-shi, 020–8551 Japan.

[††]The author is with the Faculty of Science and Technology, Keio University, Yokohama-shi, 223–8522 Japan.

[*]Presently, with Tohoku Institute of Technology.

data and domains. Zair and Tosan [21], [22] proposed a unified IFS-based model by representing basis functions of interpolation surfaces as IFS attractors. In contrast to these studies, Szeliski's research [19] does not concern the IFS. Rather, he proposed the constrained fractal based on the controlled-continuity spline.

We have developed the IST based on the construction of the Cantor set, which is one of the most typical IFS's. However, our approach is completely different from the above mentioned studies. The primary difference is explained by "local resemblance." We will describe our approach in the following sections.

3. Iterated Shuffle Transformation

In this section, we describe the definition of the IST and its properties. Although most of the statement in this section is made in mathematical ways, we adopt such description manner in order to bring clear information about the IST to readers.

An IST is a one-to-one and onto mapping that maps one point to another in a fashion similar to the construction of the Cantor set, and constructs the structure of "local resemblance" on a space. We have two IST types: *unit-IST* and *connected-IST*. The former is the basic unit of the IST, while the latter is constructed from a set of unit-IST's.

In the following, we first present the definition of IST on code spaces in Sect. 3.1. Based upon this, we then propose multi-dimensional IST defined on geometric spaces in Sect. 3.2. We explain the property of local resemblance in Sect. 3.3, and finally, we compare the IST to the IFS in Sect. 3.4, and to the traditional continuous interpolation in Sect. 3.5.

3.1 Definition of IST on Code Space

3.1.1 Unit-IST on Code Space

Let Σ_L denote the code space of L symbols by

$$\Sigma_L = \{\alpha = \alpha_1\alpha_2\alpha_3\cdots \mid \alpha_j \in \mathbf{Z}_L,\ j = 1, 2, 3, \ldots\},$$

where $\mathbf{Z}_L = \{0, 1, \ldots, L-1\}$ for integer $L \geq 1$. A unit-IST is the mapping defined on Σ_L as follows.

Definition 1. *For $\alpha \in \Sigma_L$, let $D_U^{L,k} : \Sigma_L \to \Sigma_L$ be the mapping for integer $k \geq 1$ such that*

$$\alpha = \alpha_1\alpha_2\cdots\alpha_{k-1}\alpha_k\alpha_{k+1}\cdots, \tag{1}$$

$$D_U^{L,k}(\alpha) = \alpha_k\alpha_1\alpha_2\cdots\alpha_{k-1}\alpha_{k+1}\cdots. \tag{2}$$

We refer to $D_U^{L,k}$ as the internal-shuffle (in-shuffle) mapping. Then, a unit-IST is defined as the mapping $D_{UNI}^{L,K} : \Sigma_L \to \Sigma_L$ for integer $K \geq 0$ such that

$$D_{UNI}^{L,K}(\alpha) = \begin{cases} \alpha & \text{if } K = 0, \\ D_U^{L,K}(D_{UNI}^{L,K-1}(\alpha)) & \text{if } K \geq 1. \end{cases} \tag{3}$$

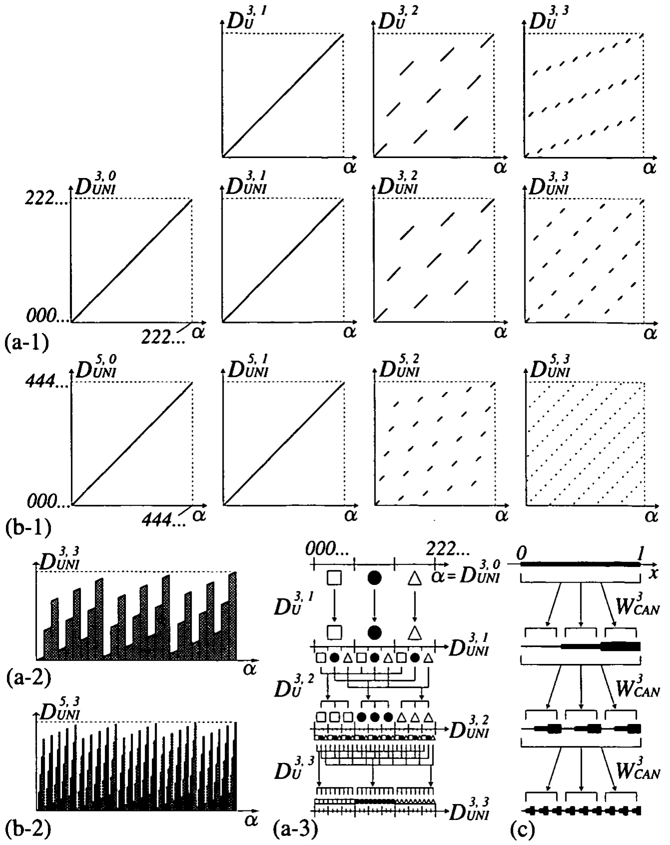


Fig. 1 Graphs and illustrations of unit-IST's on code spaces.

Figure 1 (a-1) and (b-1) show the graphs of $D_U^{L,k}$, $D_{UNI}^{L,K}$ for $L = 3$ and 5, respectively†. (a-2) and (b-2) are the graphs of $D_{UNI}^{3,3}$ and $D_{UNI}^{5,3}$; two axes of each graph are given a different ratio from that of (a-1) or (b-1). (a-3) illustrates the iteration steps of (a-1). The movements of squares, circles, and triangles show the transformation of codes α based on Eq. (2). As the iteration proceeds, the axis is subdivided into smaller intervals. (c) illustrates the construction of an N-adic Cantor set for $N = 3$, where $W_{CAN}^N = \{w_i(x) = (1/N)x + i/N,\ x \in [0,1],\ i = 0, \ldots, N-1\}$. It is shown for comparison with (a-3) (cf. Sect. 3.4).

We can obtain the following theorems for $D_{UNI}^{L,K}$.

Theorem 1. *The mapping $D_{UNI}^{L,K}$ is one-to-one and onto.*

Proof. Equations (1), (2), and (3) complete the proof. □

Theorem 2. *For $K \geq 1$, the mapping $D_{UNI}^{L,K}$ is formulated as the scale reverse formula of unit-IST by*

$$\alpha = \alpha_1\alpha_2\cdots\alpha_{K-1}\alpha_K\alpha_{K+1}\alpha_{K+2}\cdots, \tag{4}$$

$$D_{UNI}^{L,K}(\alpha) = \alpha_K\alpha_{K-1}\cdots\alpha_2\alpha_1\alpha_{K+1}\alpha_{K+2}\cdots. \tag{5}$$

†To be exact, codes α do not have any order and a code space Σ_L does not have any topology. In Fig. 1 and the explanation above, in order to help readers understand the behavior of an IST intuitively, a code space is given the topology based on the order of numbers in $\mathbf{Z}_L$.

Proof. Equations (1), (2), and (3) complete the proof. □

Theorem 3. *For all $\alpha \in \Sigma_L$, if $\alpha' = D_{UNI}^{L,K}(\alpha)$, then $\alpha = D_{UNI}^{L,K}(\alpha')$. A pair of α and α' is referred to as the reverse pair of unit-IST.*

Proof. The case of $K = 0$ is trivial. For $K \geq 1$, Theorem 2 completes the proof. □

3.1.2 Connected-IST on Code Space

Let $\hat{\Sigma}_L$ denote the code space expanded from Σ_L by

$$\hat{\Sigma}_L = \{\hat{\alpha} = \alpha_0.\alpha = \alpha_0.\alpha_1\alpha_2\alpha_3\cdots \mid \alpha_0 \in \mathbf{Z},\ \alpha_j \in \mathbf{Z}_L,\ j = 1,2,3,\ldots\},$$

where $\mathbf{Z} = \{\ldots,-2,-1,0,1,2,\ldots\}$. A connected-IST is the mapping defined on $\hat{\Sigma}_L$ as follows.

Definition 2. *For $\hat{\alpha} \in \hat{\Sigma}_L$, let $G_C : \hat{\Sigma}_L \rightarrow \hat{\Sigma}_L$ be a one-to-one and onto mapping such that*

$$\hat{\alpha} = \alpha_0.\alpha = \alpha_0.\alpha_1\alpha_2\alpha_3\cdots, \tag{6}$$

$$G_C(\hat{\alpha}) = \alpha'_0.\alpha' = \alpha'_0.\alpha'_1\alpha'_2\alpha'_3\cdots, \tag{7}$$

$$\begin{cases} \alpha'_0 = B_C(\alpha_0,\alpha_1), \\ \alpha'_1 = E_C(\alpha_0,\alpha_1), \\ \alpha'_j = R_C(\alpha_0,\alpha_1,\alpha_j),\ j \geq 2. \end{cases}$$

We refer to G_C as the external-shuffle (ex-shuffle) mapping. The functions B_C, E_C, and R_C must satisfy the following two conditions: 1) for B_C and E_C, the mapping $M_{BE} : (\alpha_0,\alpha_1) \rightarrow (\alpha'_0,\alpha'_1)$ is one-to-one and onto, 2) for R_C, when both α_0 and α_1 are fixed, the mapping $R_C : \alpha_j \rightarrow \alpha'_j$ is one-to-one and onto[†]. *Now, using G_C and $D_U^{L,k}$ in Def.1, let $D_C^{L,k} : \hat{\Sigma}_L \rightarrow \hat{\Sigma}_L$ be the mapping for $k \geq 1$ such that*

$$D_C^{L,k}(\hat{\alpha}) = G_C(\alpha_0.D_U^{L,k}(\alpha)). \tag{8}$$

Then, a connected-IST is defined as the mapping $D_{CON}^{L,K} : \hat{\Sigma}_L \rightarrow \hat{\Sigma}_L$ for $K \geq 0$ such that

$$D_{CON}^{L,K}(\hat{\alpha}) = \begin{cases} \hat{\alpha} & \text{if } K = 0, \\ D_C^{L,K}(D_{CON}^{L,K-1}(\hat{\alpha})) & \text{if } K \geq 1. \end{cases} \tag{9}$$

Figure 2 shows two different connected-IST's for the same $L = 3$ [††]. (a-1) and (b-1) show the graphs of G_C, $D_{CON}^{L,K}$ of each IST. In (b-1), we use the notations G_F, $D_{FUL}^{L,K}$ rather than G_C, $D_{CON}^{L,K}$. This will be explained in Sect. 3.2.3 (1). The numbers $3,4,\ldots,7$ written along the axes of each graph represent α_0. (a-2) and (b-2) illustrate the iteration steps of (a-1) and (b-1), respectively, based on Eq. (8). In each case, the transformation of codes $\hat{\alpha}$ only on the interval of $\alpha_0 = 5$ is illustrated in a similar way to Fig. 1 (a-3). The transformations on the intervals of other α_0 are the same, which can be easily understood by shifting the illustration along the axis. The set of the three arrows shown as G_C (or G_F) represents the one-to-one and onto mapping M_{BE} by functions B_C and E_C described in Def.2. In both cases, the positions of $\hat{\alpha}$ inside each interval having an arrow are not changed, that is, $R_C(\alpha_0,\alpha_1,\alpha_j) = \alpha_j$, $j \geq 2$. In general, functions B_C and E_C work to exchange the intervals that are determined by α_0, α_1 and have $1/L$ length on the axis as transferred units. On the other hand, function R_C works to rearrange the positions of $\hat{\alpha}$ inside each of those intervals by changing α_j, $j \geq 2$, from one number to another in $\{0,1,\ldots,L-1\}$ in one-to-one and onto manner. Function R_C is indispensable for multi-dimensional connected-IST's on irregular constructive spaces (cf. Sect. 3.2.2). In-shuffle mapping $D_U^{L,k}$ works inside the interval of each α_0 to execute each unit-IST, and is given a definite transformation rule defined by Eq. (2). On the other hand, ex-shuffle mapping G_C can be given various transformation rules as long as it satisfies the conditions described in Def.2, and works to connect the unit-IST's that are executed on the inter-

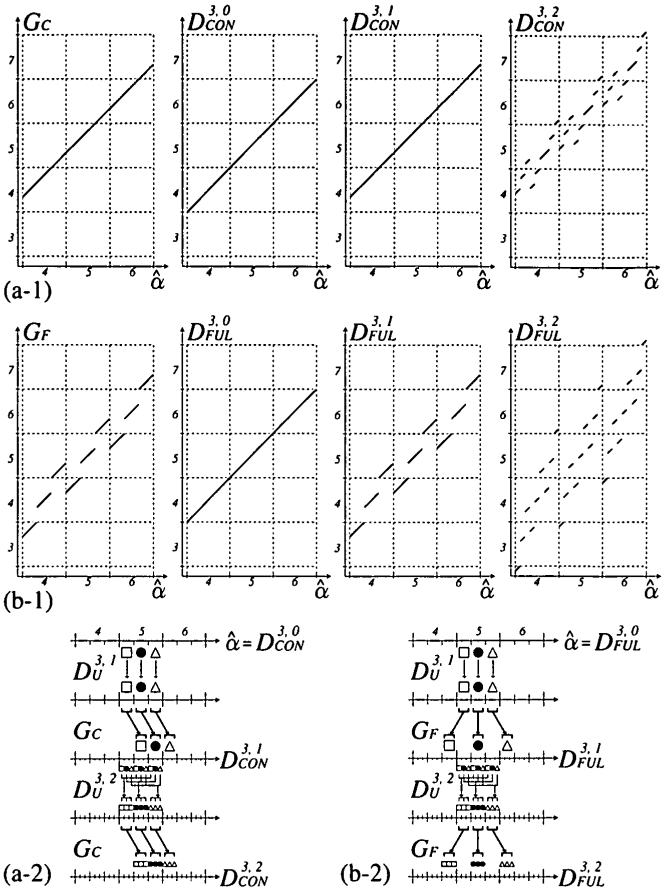


Fig. 2 Graphs and illustrations of connected-IST's on code spaces.

[†] Actually, satisfying both 1) and 2) is a sufficient condition for G_C to be one-to-one and onto.

[††] For the same reason as described in the footnote in Sect. 3.1.1, in Fig. 2 and the explanation above, a code space $\hat{\Sigma}_L$ is given the topology based on the order of numbers in $\mathbf{Z}$ and $\mathbf{Z}_L$.

vals of different α_0. As a result, the transformation rule of a connected-IST is determined by G_C.

We can obtain the next theorem for $D_{CON}^{L,K}$.

Theorem 4. *The mapping $D_{CON}^{L,K}$ is one-to-one and onto.*

Proof. The mapping $D_U^{L,k}$ is one-to-one and onto from Eq. (2). The mapping G_C is also one-to-one and onto from Def. 2. Hence, from Eq. (8), $D_C^{L,k}$ is one-to-one and onto. Then, Eq. (9) completes the proof. □

3.2 Definition of IST on Geometric Space

3.2.1 Unit-IST on Geometric Space

(1) Unit Cube

D-dimensional unit cube $\mathbf{U} \subset \mathbf{R}^D$, $D \geq 1$, is the geometric space defined as

$$\mathbf{U} = \{\mathbf{u} = (u_1, u_2, \ldots, u_D) \mid u_i \in \mathbf{I},\ i = 1, 2, \ldots, D\},$$

where $\mathbf{I} = [0, 1] \subset \mathbf{R}$. We can construct the one-to-one and onto mapping $\phi : \Sigma_L \to \mathbf{U}$ if there exists integer $N \geq 1$ such that $L = N^D$, where N is referred to as the *interval division number*:

$$\mathbf{u} = \phi(\alpha). \tag{10}$$

To construct Eq. (10), we introduce intermediate value $\beta = (\beta_1, \beta_2, \ldots, \beta_D)$, where $\beta_i = \beta_{i1}\beta_{i2}\beta_{i3}\cdots$, $\beta_{ij} \in \mathbf{Z}_N$, $i = 1, 2, \ldots, D$, $j = 1, 2, 3, \ldots$. The relationship between β_i and u_i is obtained by the mapping φ:

$$u_i = \varphi(\beta_i) = \textstyle\sum_{j=1}^{\infty} \beta_{ij} \left(\frac{1}{N}\right)^j,\ i = 1, 2, \ldots, D. \tag{11}$$

The relationship between α_j and $(\beta_{1j}, \beta_{2j}, \ldots, \beta_{Dj})$ is obtained by the following procedure:

```
$x = \alpha_j$;
for( $i = 1, 2, \ldots, D$ ) {
    $\beta_{ij} = x$ mod $N$;   $x = x/N$;
}
```

Let ψ denote the above procedure as follows:

$$(\beta_{1j}, \beta_{2j}, \ldots, \beta_{Dj}) = \psi(\alpha_j),\quad j = 1, 2, 3, \ldots. \tag{12}$$

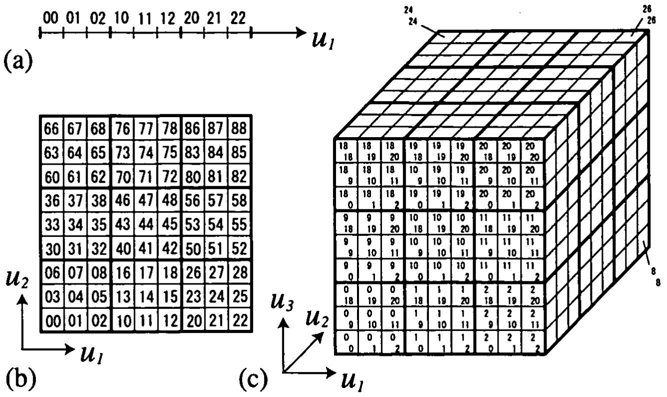
We can easily construct Eq. (10) using Eqs. (11) and (12). Figure 3 shows the cases of $N = 3$, $D = 1, 2, 3$.

(2) Unit-IST on Unit Cube

We can define the mapping $F_{UNI}^{D,N,K} : \mathbf{U} \to \mathbf{U}$ such that

$$F_{UNI}^{D,N,K}(\mathbf{u}) = \phi\left(D_{UNI}^{N^D,K}(\phi^{-1}(\mathbf{u}))\right). \tag{13}$$

Equation (13) implies that the unit-IST works on $\mathbf{U}$. Figure 4 shows the two-dimensional case for $N = 3$. The monochrome figures in the left half are the illustrations of $D_U^{N^D,k} = D_U^{9,k}$, $k = 1, 2, 3$, on $\mathbf{U}$ (cf. Eq. (2)). Triangles, squares, and circles show the movement of each coordinate region. The color figures in the right half show how each coordinate on $\mathbf{U}$ is transformed by $F_{UNI}^{D,N,K} = F_{UNI}^{2,3,K}$, $K = 0, \ldots, 5$. The continuous RGB color pattern of $F_{UNI}^{2,3,0}$ represents the initial arrangement of the coordinates. A series of color changes from $F_{UNI}^{2,3,1}$ to $F_{UNI}^{2,3,5}$ shows the transformation from the initial arrangement. We find that the unit cube is subdivided into smaller regions as the iteration proceeds and the regions are shuffled in a recursive manner.



66	67	68	76	77	78	86	87	88
63	64	65	73	74	75	83	84	85
60	61	62	70	71	72	80	81	82
36	37	38	46	47	48	56	57	58
33	34	35	43	44	45	53	54	55
30	31	32	40	41	42	50	51	52
06	07	08	16	17	18	26	27	28
03	04	05	13	14	15	23	24	25
00	01	02	10	11	12	20	21	22

18 18	18 19	18 20	19 18	19 19	19 20	20 18	20 19	20 20
18 9	18 10	18 11	10 9	19 10	19 11	20 9	20 10	20 11
18 0	18 1	18 2	19 0	19 1	19 2	20 0	20 1	20 2
9 18	9 19	9 20	10 18	10 19	10 20	11 18	11 19	11 20
9 9	9 10	9 11	10 9	10 10	10 11	11 9	11 10	11 11
9 0	9 1	9 2	10 0	10 1	10 2	11 0	11 1	11 2
0 18	0 19	0 20	1 18	1 19	1 20	2 18	2 19	2 20
0 9	0 10	0 11	1 9	1 10	1 11	2 9	2 10	2 11
0 0	0 1	0 2	1 0	1 1	1 2	2 0	2 1	2 2

Fig. 3 Relationship between Σ_{N^D} and $\mathbf{U}$. $N = 3$. The number written on each region represents $\alpha_1\alpha_2$. (a) $D = 1$. (b) 2. (c) 3.

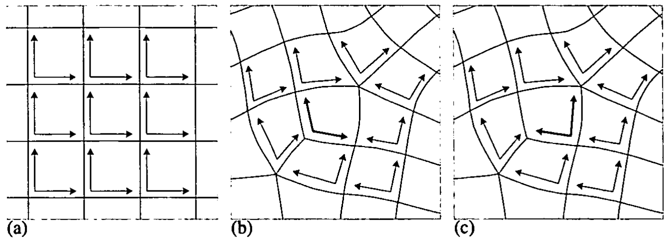


Fig. 5 Arrangements of local coordinate systems on two-dimensional constructive spaces. Each pair of arrows represents the local coordinate system $\mathbf{u} = (u_1, u_2)$ of each unit cube. (a) A regular space. (b), (c) An irregular space.

3.2.2 Connected-IST on Geometric Space

(1) Constructive Space

Consider a geometric space $\hat{\mathbf{U}}$ constructed by a set of D-dimensional unit cubes $\mathbf{U}_s$ such that

$$\begin{aligned}\hat{\mathbf{U}} &= \{\mathbf{U}_s \mid s \in \mathbf{Z}\} \\ &= \{\hat{\mathbf{u}} = (s, \mathbf{u}) = (s, (u_1, u_2, \ldots, u_D)) \mid \\ &\qquad s \in \mathbf{Z},\ u_i \in \mathbf{I},\ i = 1, 2, \ldots, D\}.\end{aligned}$$

We refer to $\hat{\mathbf{U}}$ as *D-dimensional constructive space*. Two-dimensional examples are shown in Fig. 5, where each four-sided area that has a pair of arrows is a unit cube. In particular, consider the case of arranging all unit cubes regularly in all D directions, as shown in (a). We refer to such a space as *regular constructive space*, which is denoted by $\hat{\mathbf{U}}_R$. Whereas a space having an irregular arrangement, as shown in (b), (c), is referred to as *irregular constructive space*.

Similar to the case of unit-IST, if there exists N such that $L = N^D$, then we can obtain a one-to-one and onto mapping $\hat{\phi} : \hat{\Sigma}_L \to \hat{\mathbf{U}}$ using ϕ in Eq. (10) such that

$$\hat{\mathbf{u}} = \hat{\phi}(\hat{\alpha}) = (\phi_0(\alpha_0), \phi(\alpha)), \tag{14}$$

where $s = \phi_0(\alpha_0)$ and $\mathbf{u} = \phi(\alpha)$. The mapping $\phi_0 : \mathbf{Z} \to \mathbf{Z}$ is a one-to-one and onto mapping.

(2) Connected-IST on Constructive Space

We can define the mapping $F_{CON}^{D,N,K} : \hat{\mathbf{U}} \to \hat{\mathbf{U}}$ such that

$$F_{CON}^{D,N,K}(\hat{\mathbf{u}}) = \hat{\phi}\left(D_{CON}^{N^D,K}(\hat{\phi}^{-1}(\hat{\mathbf{u}}))\right). \tag{15}$$

Equation (15) implies that the connected-IST works on $\hat{\mathbf{U}}$. Figure 6 shows two-dimensional cases, which have all different transformation patterns for $N = 3$. In each case of (a), (b), (c), and (d), the monochrome figure is the illustration of ex-shuffle mapping G_C on $\hat{\mathbf{U}}$. Bold lines indicate the boundaries between unit cubes. Triangles, squares, and circles show the movements of the nine coordinate regions of the unit cube located in the middle by G_C. Each of the nine regions is only translated and not rotated. Here, one problem shown in Fig. 5 should be considered. On a regular constructive space $\hat{\mathbf{U}}_R$, the local coordinate systems $\mathbf{u}$ of all unit cubes $\mathbf{U}_s$ can be arranged in the same direction all over $\hat{\mathbf{U}}_R$ as shown in (a). In this case, all α are also arranged in the same direction according to Eqs. (10) and (14). On the other hand, on an irregular constructive space $\hat{\mathbf{U}}$, all local coordinate systems $\mathbf{u}$, and α, cannot be arranged in the same direction as shown in (b) and (c), where the direction of $\mathbf{u}$ of the unit cube located in the middle cannot be determined. In this case, the same positions in adjacent unit cubes are not always indicated by the same number $\alpha_j \in \{0, 1, \ldots, N^D - 1\}$, $j \geq 1$. Therefore, when an ex-shuffle mapping G_C exchanges the regions that are determined by α_0, α_1 and have $1/N^D$ area (volume) over different unit cubes, the mapping G_C needs not only a mapping M_{BE} constructed by functions B_C and E_C but also a function R_C. The former works to move the regions determined by α_0, α_1 to new regions determined by $B_C(\alpha_0, \alpha_1)$, $E_C(\alpha_0, \alpha_1)$, and the latter works to change α_j to $R_C(\alpha_0, \alpha_1, \alpha_j)$, $j \geq 2$, that are adjusted to the new regions. The color figures in each of Fig. 6 show how each coordinate on $\hat{\mathbf{U}}$ is transformed by $F_{CON}^{D,N,K} = F_{CON}^{2,3,K}$, $K = 0, \ldots, 3$, in a similar way to Fig. 4. We use the notations G_F, $F_{FUL}^{D,N,K}$ in (a) and G_Q, $F_{QFUL}^{D,N,K}$ in (b) rather than G_C, $F_{CON}^{D,N,K}$. This will be explained in Sect. 3.2.3. Figure 6 tells us that different ex-shuffle mappings create different transformation patterns although each in-shuffle mapping works in the same manner defined by Eq. (2), as described in Sect. 3.1.2.

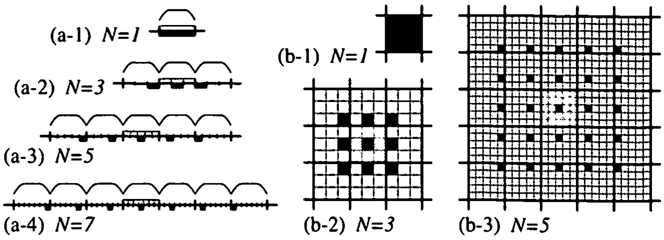


Fig. 7 External-shuffle mappings G_F of fully-connected-IST's. (a-1) – (a-4) $D = 1$. Each upper bracket ⌒ on the axis indicates the interval of a one-dimensional unit cube. (b-1) – (b-3) $D = 2$. Bold lines indicate the boundaries between two-dimensional unit cubes.

(3) Connected-IST on Finite and Open Space

When a constructive space $\hat{\mathbf{U}}$ is finite and open, that is, has the boundaries to the outside, then it is possible that a transformed point $F_{CON}^{D,N,K}(\hat{\mathbf{u}})$ from $\hat{\mathbf{u}} \in \hat{\mathbf{U}}$ goes out of $\hat{\mathbf{U}}$. In order to avoid such a situation, we add the following condition: if $F_{CON}^{D,N,K}(\hat{\mathbf{u}}) \notin \hat{\mathbf{U}}$, then $F_{CON}^{D,N,K}(\hat{\mathbf{u}}) = \hat{\mathbf{u}}$.

3.2.3 Special Types of Connected-IST

Now, we explain two special types of connected-IST defined by ex-shuffle mappings G_F and G_Q on D-dimensional geometric spaces: *fully-connected-IST* and *quasi-fully-connected-IST*.

(1) Fully-connected-IST

On regular constructive spaces $\hat{\mathbf{U}}_R$, consider ex-shuffle mappings G_F for odd numbers N, as shown in Fig. 7. In each case of Fig. 7, each gray region is translated to each black region radially by G_F. The connected-IST of Fig. 2 (b-1,2) is defined by Fig. 7 (a-2), and Fig. 6 (a) is defined by Fig. 7 (b-2). We can say that each G_F in Fig. 7 connects unit-IST's most regularly and uniformly among all types of connected-IST's defined on $\hat{\mathbf{U}}_R$ for each pair of D and N. For example, comparing four cases in Fig. 6 makes us understand the situation. The cases of greater D and N are easy to consider. The connected-IST defined by such an ex-shuffle mapping G_F on $\hat{\mathbf{U}}_R$ is referred to as fully-connected-IST.

Let $\mathbf{t} \in \mathbf{T}$ indicate the global position of a unit cube $\mathbf{U}_s$ on $\hat{\mathbf{U}}_R$, where $\mathbf{T} = \{\mathbf{t} = (t_1, t_2, \ldots, t_D) \mid t_i \in \mathbf{Z},\ i = 1, 2, \ldots, D\}$. We can easily construct a one-to-one and onto mapping $\lambda : \mathbf{Z} \to \mathbf{T}$ such that $\mathbf{t} = \lambda(s)$. Then, using ϕ_0 in Eq. (14), we can obtain a one-to-one and onto mapping $\chi : \mathbf{Z} \to \mathbf{T}$ such that

$$\mathbf{t} = \chi(\alpha_0) = \lambda(\phi_0(\alpha_0)). \tag{16}$$

Here, we arrange all local coordinate systems $\mathbf{u}$ of $\mathbf{U}_s$ in the same direction all over $\hat{\mathbf{U}}_R$, as shown in Fig. 5 (a). Then, using Eqs. (12) and (16), for $\hat{\alpha} \in \hat{\Sigma}_{N^D}$, we can describe G_F for a fully-connected-IST such that

$$G_F(\hat{\alpha}) = \alpha'_0 . \alpha' = \alpha'_0 . \alpha'_1 \alpha'_2 \alpha'_3 \cdots, \tag{17}$$

where

$$\begin{aligned}\alpha_0' &= B_F(\alpha_0, \alpha_1)\\ &= \chi^{-1}(\chi(\alpha_0) + \psi(\alpha_1) - M\mathbf{e}), \quad (18)\\ \alpha_1' &= E_F(\alpha_0, \alpha_1) = N^D - 1 - \alpha_1, \quad (19)\\ \alpha_j' &= R_F(\alpha_0, \alpha_1, \alpha_j) = \alpha_j,\ j \geq 2. \quad (20)\end{aligned}$$

In Eq. (18), integer $M = (N-1)/2$ and D-dimensional vector $\mathbf{e} = (1, 1, \ldots, 1)$. A fully-connected-IST is denoted by $D_{FUL}^{D,N,K}$. When $D = 1$, it is also denoted by $D_{FUL}^{N,K}$. We also use the notations $F_{FUL}^{D,N,K}$, $F_{FUL}^{N,K}$.

We can obtain the following theorems for $D_{FUL}^{D,N,K}$.

Theorem 5. *For $K \geq 1$, the mapping $D_{FUL}^{D,N,K}$ is formulated by*

$$\begin{aligned}\hat{\alpha} &= \alpha_0.\alpha \ = \alpha_0.\alpha_1\alpha_2\alpha_3\cdots, \quad (21)\\ D_{FUL}^{D,N,K}(\hat{\alpha}) &= \alpha_0'.\alpha' = \alpha_0'.\alpha_1'\alpha_2'\alpha_3'\cdots, \quad (22)\end{aligned}$$

where

$$\begin{aligned}\alpha_0' &= B_{FUL}^{D,N,K}(\hat{\alpha})\\ &= \chi^{-1}(\chi(\alpha_0) + \textstyle\sum_{j=1}^{K}(\psi(\alpha_j) - M\mathbf{e})), \quad (23)\\ \alpha_j' &= E_{FUL}^{D,N,K}(\hat{\alpha}, j)\\ &= N^D - 1 - \alpha_{K-j+1} \quad \text{if } 1 \leq j \leq K, \quad (24)\\ \alpha_j' &= \alpha_j \quad \text{if } j \geq K+1. \quad (25)\end{aligned}$$

The set of the formulas from Eqs. (22) to (25) is referred to as the scale reverse formula of fully-connected-IST.

Proof. Definition 2 and the equations from Eqs. (17) to (20) complete the proof. □

Theorem 6. *For all $\hat{\alpha} \in \hat{\Sigma}_{N^D}$, if $\hat{\alpha}' = D_{FUL}^{D,N,K}(\hat{\alpha})$, then $\hat{\alpha} = D_{FUL}^{D,N,K}(\hat{\alpha}')$. A pair of $\hat{\alpha}$ and $\hat{\alpha}'$ is referred to as the reverse pair of fully-connected-IST.*

Proof. Theorem 5 completes the proof. □

(2) Quasi-fully-connected-IST

A fully-connected-IST can be defined only on a regular constructive space. On the other hand, a quasi-fully-connected-IST, which we explain here, can be defined on all kinds of constructive spaces, although it is restricted to $N = 3$. The two-dimensional case is shown in Fig. 6 (b). From its ex-shuffle mapping G_Q illustrated there, this IST can be considered to connect unit-IST's most regularly and uniformly among all types of connected-IST's defined on arbitrary constructive spaces for $N = 3$. This can be generalized for all $D \geq 1$. When $D = 1$, it is equal to a fully-connected-IST. We use the notations G_Q, $D_{QFUL}^{D,N,K}$, and $F_{QFUL}^{D,N,K}$.

We cannot formulate a quasi-fully-connected-IST in a similar way to a fully-connected-IST for the following problems: 1) we cannot indicate the global position of a unit cube $\mathbf{U}_s$ on arbitrary $\hat{\mathbf{U}}$ in a definite way, 2) we cannot arrange all local coordinate systems $\mathbf{u}$ in the same direction (cf. Figure 5 (b), (c)). Therefore, we use a procedural way, when we implement it.

3.3 Local Resemblance in Space/Scale Direction

3.3.1 Concept of Local Resemblance

We have two types of directions: *space direction* and *scale direction*. The former is the direction along which a point of view changes, such as “moves upward,” “downward,” “to the right,” or “to the left.” The latter is the direction along which a field of view changes, such as “expands” or “reduces.” Based on them, we can consider “locality in space direction” and “self-similarity in scale direction.” The former is the property that each local position on a space can be identified individually and the shape or other attributes of an object can be recognized based on it. The latter is the property that similar shapes or structures appear repeatedly in every scale. An IST constructs the structure of “local resemblance in space/scale direction,” which unifies “locality” and “self-similarity” above. In brief, “local resemblance” is the property that satisfies the following: 1) when you move along the space direction you can identify each local position individually, 2) when you move along the scale direction you can see similar structures appear repeatedly. We discuss the local resemblances constructed by IST's below.

3.3.2 Local Resemblance by Unit-IST

From Fig. 1 (a-2), (b-2), and Fig. 4, a unit-IST seems to construct a kind of repeated structure, similar to self-similarity in fractals, as the scale changes. The structure constructed by $D_{UNI}^{L,K}$ is explained analytically as follows. For $l = 1, 2, \ldots, K$, let $\Delta_l = \delta_1\delta_2\delta_3\cdots$, where $\delta_l = 1$ and $\delta_{j \neq l} = 0$. Given fixed $\bar{\alpha}_1, \bar{\alpha}_2, \ldots, \bar{\alpha}_{l-1}$ (when $l = 1$, there is no fixed $\bar{\alpha}_j$). Then, for $\alpha_l \in \mathbf{Z}_L$, consider $D_{UNI}^{L,K}$ defined on L domains $S_{\alpha_l} = \{\alpha \mid \alpha = \bar{\alpha}_1\bar{\alpha}_2\cdots\bar{\alpha}_{l-1}\alpha_l\alpha_{l+1}\alpha_{l+2}\cdots\}$. By Eq. (5), on each of these L domains, we obtain

$$D_{UNI}^{L,K}(\alpha + \alpha_l\Delta_l) = D_{UNI}^{L,K}(\alpha) + \alpha_l\Delta_{K-l+1}, \quad (26)$$

where $\alpha = \bar{\alpha}_1\cdots\bar{\alpha}_{l-1}\,0\,\alpha_{l+1}\cdots \in S_0$ and $\alpha + \alpha_l\Delta_l = \bar{\alpha}_1\cdots\bar{\alpha}_{l-1}\alpha_l\alpha_{l+1}\cdots \in S_{\alpha_l}$. Equation (26) tells us that the graph on the domain S_{α_l} is constructed by translating the graph on the domain S_0 by $\alpha_l\Delta_l$ horizontally and by $\alpha_l\Delta_{K-l+1}$ vertically. This construction manner works well for all $l = 1, 2, \ldots, K$, and creates a repeated structure in the scale direction. Actually, we can easily find such structures in Fig. 1 (a-2) and (b-2). It is noticeable that the value of $D_{UNI}^{L,K}(\alpha)$ for $\alpha \in \Sigma_L$ on the graph not only varies upward and downward in such a repeated manner but also increases gradually as a whole when α increases continuously along α axis.

This means that $D^{L,K}_{UNI}$ differs in the space direction. The color figures in Fig. 4 show us the two-dimensional case. Those figures clearly have repeated structures in the scale direction. Moreover, even though the iteration K increases, the lower left part of $\mathbf{U}$ is redder, the upper right part is bluer, and the middle part is greener than other parts, respectively. That is, the proportion of red, blue, and green depends on the position in the space direction. The statement above explains the local resemblance by a unit-IST.

3.3.3 Local Resemblance by Connected-IST

Now, the structure constructed by a fully-connected-IST is explained as follows. For $l = 1, 2, \ldots, K$, given fixed $\bar{\alpha}_0, \bar{\alpha}_1, \bar{\alpha}_2, \ldots, \bar{\alpha}_{l-1}$. Then, for $\alpha_l \in \mathbf{Z}_L$, consider $D^{D,N,K}_{FUL}$ defined on each of L domains $\hat{S}_{\alpha_l} = \{\hat{\alpha} \mid \hat{\alpha} = \bar{\alpha}_0.\bar{\alpha}_1\bar{\alpha}_2\cdots\bar{\alpha}_{l-1}\alpha_l\alpha_{l+1}\alpha_{l+2}\cdots\}$. Here, we describe

$$D^{D,N,K}_{FUL}(\hat{\alpha}) = \alpha'_0.\alpha'_1\alpha'_2\cdots, \tag{27}$$

$$D^{D,N,K}_{FUL}(\hat{\alpha} + \alpha_l\Delta_l) = \alpha''_0.\alpha''_1\alpha''_2\cdots, \tag{28}$$

for $\hat{\alpha} = \bar{\alpha}_0.\bar{\alpha}_1\cdots\bar{\alpha}_{l-1}\,0\,\alpha_{l+1}\cdots \in \hat{S}_0$ and $\hat{\alpha} + \alpha_l\Delta_l = \bar{\alpha}_0.\bar{\alpha}_1\cdots\bar{\alpha}_{l-1}\alpha_l\alpha_{l+1}\cdots \in \hat{S}_{\alpha_l}$. From Theorem 5, we obtain the relationship between Eqs. (27) and (28) such that

$$\alpha''_0 = \chi^{-1}(\chi(\alpha'_0) + \psi(\alpha_l)), \tag{29}$$

$$\alpha''_{K-l+1} = \alpha'_{K-l+1} - \alpha_l, \tag{30}$$

$$\alpha''_j = \alpha'_j,\ j \geq 1,\ j \neq K - l + 1. \tag{31}$$

In the case of $D = 1$, if $\chi(\alpha_0) = \alpha_0$, then the equations from Eqs. (27) to (31) are united to one equation by

$$D^{N,K}_{FUL}(\hat{\alpha} + \alpha_l\Delta_l) = D^{N,K}_{FUL}(\hat{\alpha}) + \alpha_l(\Delta_0 - \Delta_{K-l+1}),$$

where $\Delta_0 = 1.00\cdots$. In this case, we find that the graph of $D^{N,K}_{FUL}$ on each domain $\hat{S}_{\alpha_l}$ is constructed based on the graph on the domain $\hat{S}_0$ in a similar fashon to the case of $D^{L,K}_{UNI}$ described in Sect. 3.3.2. Moreover, for all $\hat{\alpha} \in \hat{\Sigma}_N$, we find the following relationship:

$$D^{N,K}_{FUL}(\hat{\alpha} + \Delta_0) = D^{N,K}_{FUL}(\hat{\alpha}) + \Delta_0.$$

The statement above tells us that a fully-connected-IST constructs a local resemblance in a similar fashion to a unit-IST. Figure 2 (b-1) and Fig. 6 (a) show the one-dimensional and two-dimensional cases, respectively. In fact, any connected-IST constructs a local resemblance. Figure 6 (b) shows the case of a two-dimensional quasi-fully-connected-IST, and Fig. 2 (a-1), Fig. 6 (c), and (d) show other cases, respectively. Here, comparing Fig. 4 and Fig. 6 tells us one difference between a unit-IST and a connected-IST. That is, although the proportion of red, blue, and green depends on the position on the space in each IST, the figures generated by the connected-IST's in Fig. 6 have greater locality, which means that each color tends to remain around the original local position even after the iteration proceeds, than those generated by the unit-IST in Fig. 4. We will mention this property (global/local effectiveness) in Sect. 3.5.

3.4 Comparison between IST and IFS

Comparing (a-3) and (c) in Fig. 1, although their recursive construction steps look similar, a unit-IST and a Cantor set are quite different as described below. A Cantor set is created by applying each w_i in W^N_{CAN} to the whole interval $[0, 1]$ in every step. In general, each affine mapping w_i in an IFS $\{\mathbf{X}; w_i, i = 0, \ldots, N - 1\}$ is applied to the whole space $\mathbf{X}$ to create its attractor [3]. Such a construction scheme leads to the multiplication of each point that derives from the initial space, as shown in (c). On the other hand, a unit-IST subdivides a unit cube into a number of regions and exchanges their positions, as shown in (a-3). Thus, a unit-IST maintains every point without multiplication, which is actually proved as Theorem 1. This makes it possible to identify each point on the space, and leads to introducing locality into self-similarity.

We here mention that the IFS has the idea of addressing points, where an IFS addresses points on its attractor in terms of the sequences of w_i [3]. This idea may be used to identify each point. The difference between this idea and that of the IST is that an IST constructs a one-to-one and onto mapping defined on a space and makes it possible to identify each point on the space by using α or $\hat{\alpha}$ as an indicator. Moreover, the concept of "connection," which means that unit-IST's are connected into a connected-IST, is an original important property of the IST.

3.5 Comparison between IST and Traditional Continuous Interpolation

Traditional continuous interpolation methods interpolate control points smoothly. Moving a control point produces greater deformation on the curve or surface part nearer the control point. This property is based on "locality" (cf. Sect. 3.3.1) and is essential for continuous interpolation.

A Bézier curve of degree $n \geq 1$ is generated from $n + 1$ Bézier control points by using Bernstein polynomials $B^n_i(u)$, $i = 0, 1, \ldots, n$, for parameter $u \in \mathbf{I}$ [8]. Bézier interpolation is considered to be the basic unit of continuous interpolation. A composite Bézier curve is defined by connecting Bézier curves and has continuity C^r or G^r determined by the connection condition. In particular, a composite Bézier curve becomes a B-spline curve when the continuity is the maximum C^{n-1}. All the interpolation curves described above can be easily expanded to interpolation surfaces.

Considering the relationship between continuous interpolation and the IST, we can conclude the following. First, Bézier interpolation corresponds to a unit-

IST for the following reasons. 1) Both of them work as basic units. 2) Every control point affects the entire curve or surface in Bézier interpolation, while every $\mathbf{u}$ can be moved anywhere on $\mathbf{U}$ in a unit-IST. This property is referred to as *global effectiveness*. 3) The recursive manner of the de Casteljau algorithm [8] for Bézier interpolation is similar to the iteration steps of a unit-IST. Besides, we can also list the correspondent items: degree n — interval division number N; Bernstein polynomials $B_i^n(u)$ for $u \in \mathbf{I}$ — mapping $F_{UNI}^{D,N,K}(\mathbf{u})$ for $\mathbf{u} \in \mathbf{U}$. Secondly, composite Bézier interpolation corresponds to a connected-IST for the following reasons. 1) Both of them are constructed by connecting basic units. 2) Each control point affects only the local part of the entire curve or surface in composite Bézier interpolation, while each $\hat{\mathbf{u}}$ can be moved only on the local region of $\hat{\mathbf{U}}$ by one iteration step in a connected-IST. This property is referred to as *local effectiveness*. 3) Continuity C^r or G^r of composite Bézier interpolation corresponds to ex-shuffle mapping G_C of a connected-IST, because they determine the connection conditions and the entire structures. Finally, B-spline interpolation corresponds to a fully-connected-IST, because the maximum connection conditions are satisfied.

In general, it is difficult to guarantee continuity C^r of large r on an arbitrary irregular mesh. Therefore, we are often forced to adopt smaller r, and sometimes adopt G^r. In addition, as degree n is larger, it tends to be more difficult to make the continuity close to the maximum C^{n-1}. These situations are similar to those of connected-IST's. That is, on an arbitrary D-dimensional irregular constructive space, it is difficult to define an ex-shuffle mapping that is close to G_F and satisfies the conditions in Def.2. The larger N is, the more difficult this becomes.

4. Wrinkly Surface

In this section, we propose the construction scheme of the WR surface on irregular meshes by applying two-dimensional IST's described in Sect. 3. We first present the definition of the WR surface on irregular meshes by arbitrarily connecting tensor product WR surfaces. Then, we present various examples and shape models created using WR surfaces.

4.1 Definition of Wrinkly Surface

A WR surface is a parametric surface defined on a mesh of control points $\mathbf{r}_i = (\mathbf{p}_i, h_i, \mathbf{n}_i)$, where $\mathbf{p}_i = (x_i, y_i, z_i)$ is a three-dimensional coordinate, h_i is a *height value*, and $\mathbf{n}_i = (nx_i, ny_i, nz_i)$ is a *displacement vector*. A WR surface $\mathbf{R}$ is defined for parameter $\mathbf{u}^*$ on a two-dimensional parameter space $\mathbf{U}^*$ such that

$$\mathbf{R}(\mathbf{u}^*) = \mathbf{P}(\mathbf{u}^*) + H\left(F_{IST}^{2,N,K^{-1}}(\mathbf{u}^*)\right)\mathbf{N}(\mathbf{u}^*). \quad (32)$$

In Eq. (32), $F_{IST}^{2,N,K}$ denotes $F_{UNI}^{2,N,K}$ or $F_{CON}^{2,N,K}$. And also, $\mathbf{u}^*$ and $\mathbf{U}^*$ denote $\mathbf{u}$ and $\mathbf{U}$ in the case of a unit cube, or $\hat{\mathbf{u}}$ and $\hat{\mathbf{U}}$ in the case of a constructive space. The *base surface* $\mathbf{P}$ is defined by interpolating $\mathbf{p}_i$ by continuous interpolation, such as Bézier or B-spline interpolation. The *height function* H and the *displacement function* $\mathbf{N}$, which satisfies $|\mathbf{N}| \equiv 1$, are defined based on h_i and $\mathbf{n}_i$, respectively, in the same way as $\mathbf{P}$. Equation (32) indicates that $\mathbf{R}$ is constructed by the displacement mapping of H in the direction of $\mathbf{N}$ on $\mathbf{P}$. We can also supply to the WR surface various other attributes, such as color or transparency, and apply an IST to these attributes in the same way as to H. For example, when color values $\mathbf{c}_i = (r_i, g_i, b_i)$ are given, a *color function* $\mathbf{C}$ is obtained from them, and the color pattern $\mathbf{C}_{\mathbf{R}}$ given on the WR surface $\mathbf{R}$ is determined as follows:

$$\mathbf{C}_{\mathbf{R}}(\mathbf{u}^*) = \mathbf{C}\left(F_{IST}^{2,N,K^{-1}}(\mathbf{u}^*)\right).$$

The mapping $F_{IST}^{2,N,K}$ works to spread height values, or other attributes, with local resemblance over the surface. This produces the essential feature of the WR surface.

We can define another WR surface type $\mathbf{R}_{sp}$, which is referred to as the *superposed wrinkly surface* (*SP-WR surface*), by superposing height functions H of different iteration steps $K = 0, 1, 2, \ldots, K_{max}$ such that

$$\begin{aligned}&\mathbf{R}_{sp}(\mathbf{u}^*)\\ &= \mathbf{P}(\mathbf{u}^*) + \left\{\textstyle\sum_{K=0}^{K_{max}} W^K(\mathbf{u}^*) H\!\left(F_{IST}^{2,N,K^{-1}}(\mathbf{u}^*)\right)\right\}\mathbf{N}(\mathbf{u}^*).\end{aligned} \quad (33)$$

The maximum iteration K_{max} is usually determined based on the screen resolution when the surface is rendered [9]–[11]. The *weight function* W^K is defined for each K using a *ratio function* $0 < \delta(\mathbf{u}^*) < 1$ such that

$$W^K(\mathbf{u}^*) = \frac{1 - \delta(\mathbf{u}^*)}{1 - \{\delta(\mathbf{u}^*)\}^{K_{max}+1}}\{\delta(\mathbf{u}^*)\}^K. \quad (34)$$

Equation (34) satisfies $\sum_{K=0}^{K_{max}} W^K(\mathbf{u}^*) \equiv 1$. This surface can be also given other attributes. For color values $\mathbf{c}_i$, the color pattern $\mathbf{C}_{\mathbf{R}_{sp}}$ given on the surface $\mathbf{R}_{sp}$ is determined using a weight function $W_{\mathbf{C}}^K$ as follows:

$$\mathbf{C}_{\mathbf{R}_{sp}}(\mathbf{u}^*) = \textstyle\sum_{K=0}^{K_{max}} W_{\mathbf{C}}^K(\mathbf{u}^*)\mathbf{C}\left(F_{IST}^{2,N,K^{-1}}(\mathbf{u}^*)\right).$$

In [11], we proposed the *IST-Bézier surface* and the *IST-B-spline surface* as concrete representations of the WR surface (we use the term “WR surface” as a generic name). The former was defined by applying Bézier interpolation to $\mathbf{p}_i$, h_i, and $\mathbf{n}_i$, and adopting the tensor product of two one-dimensional unit-IST's as $F_{IST}^{2,N,K}$ in Eq. (32). The latter was defined using B-spline interpolation and one-dimensional connected-IST's. Both of them were defined on regular meshes. In

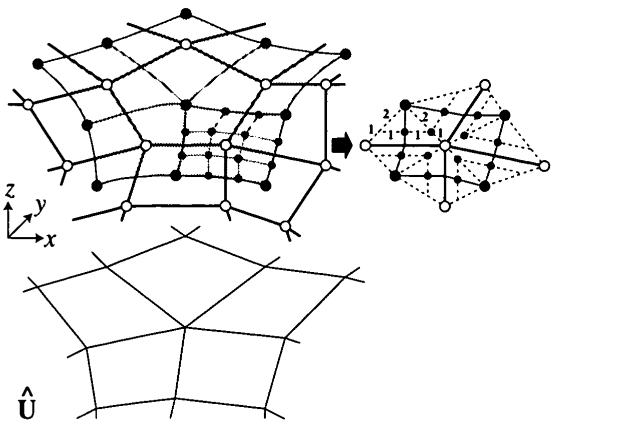


Fig. 8 Composite Bézier interpolation on an irregular control mesh.

this paper, we propose the *composite IST-Bézier surface* defined on an irregular mesh. We adapt composite Bézier interpolation to cover an irregular mesh by connecting tensor product Bézier patches as shown in Fig. 8, which describes the construction of a base surface $\mathbf{P}$ for degree $n = 3$. The white points ○ denote $\mathbf{p}_i$, the black points ● denote Bézier control points, and the Bézier patches are colored gray. The large black points are the corner points of the Bézier patches, and each of their positions is determined as the centroid of points $\mathbf{p}_i$ that surround it. The small black points are the other Bézier control points, and their positions are determined by the internal division of the ratio shown in Fig. 8. All the Bézier patches have their own parameter spaces $\mathbf{U}_s = \mathbf{I} \times \mathbf{I}$, which compose a constructive space $\hat{\mathbf{U}} = \{\mathbf{U}_s\}$ for a connected-IST. A height function H and a displacement function $\mathbf{N}$ are also constructed from h_i and $\mathbf{n}_i$ in a similar fashion, and so are functions of other attributes.

4.2 Examples of Wrinkly Surface

Figure 9 shows some examples of *superposed IST-Bézier surfaces* (*SP-IST-Bézier surfaces*). (a,b,c-1) show control meshes of $\mathbf{r}_i = (\mathbf{p}_i, h_i, \mathbf{n}_i)$ and base surfaces, which are generated by Bézier interpolation of degree $n = 3$. Each stick having a round tip that runs out from grid point $\mathbf{p}_i$ indicates height value h_i having direction $\mathbf{n}_i$. Each control point $\mathbf{r}_i$ is given a color value $\mathbf{c}_i$, which leads to creating the continuous RGB color pattern[†]on each base surface in (a,b,c-1), as well as h_i for applying an IST in order to make readers easily understand the surface features. (a,b,c-2) are continuous interpolation surfaces based on (a,b,c-1). They are generated without applying IST's, correctly speaking, by setting $F_{IST}^{2,N,K-1}(\mathbf{u}^*) = \mathbf{u}$ in Eq. (33) and coloring them using their color functions $\mathbf{C}$ directly. In comparison, (a-3,4), (b-3,4), (c-3,4) are SP-IST-Bézier surfaces. They are generated by applying two-dimensional unit-IST's with $N = 3, 4, 5$, $\delta(\mathbf{u}) = 0.66, 0.75, 0.50$ (constant for all $\mathbf{u}$), $K_{max} = 4, 3, 3$, respectively. (a,b,c-3) are colored using $\mathbf{C_R}$ of $K = K_{max}$, whereas (a,b,c-4) are colored using $\mathbf{C}_{\mathbf{R}_{sp}}$, $\delta_{\mathbf{C}}(\mathbf{u}) = 0.99$. The repeated structures of height bumps and color patterns in (a,b,c-3,4) show the local resemblances by the unit-IST's (cf. Sect. 3.3.2).

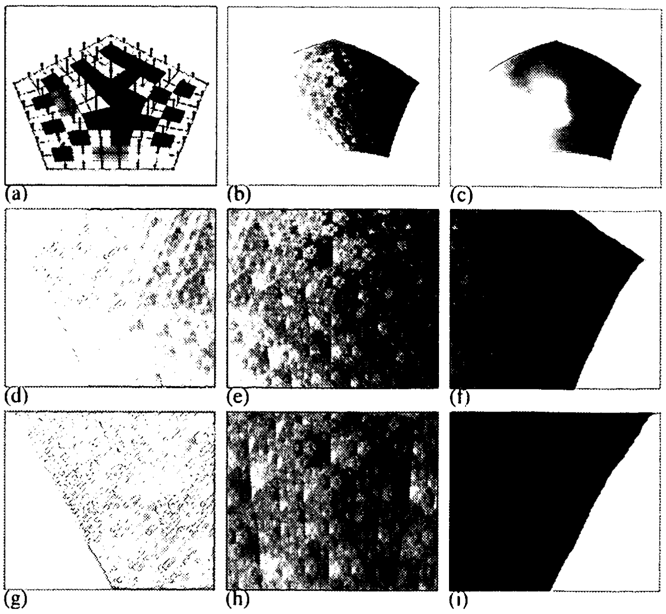


Fig. 10 A composite superposed IST-Bézier surface on an irregular pentagonal mesh.

Figure 10 shows a *composite SP-IST-Bézier surface* on an irregular pentagonal mesh (such a mesh could not be treated by the method of [10]). This surface is considered to be constructed by connecting SP-IST-Bézier surfaces. (a) shows a control mesh and a base surface covered by Bézier patches having different colors. (b) is a composite SP-IST-Bézier surface based on (a). It is generated by a quasi-fully-connected-IST, $\delta(\hat{\mathbf{u}}) = 0.50$, $K_{max} = 4$, and composite Bézier interpolation of degree $n = 3$. It is colored using $\mathbf{C_R}$ of $K = K_{max}$. (c) is a continuous interpolation surface without applying the IST. (d), (e), (f) are three-fold magnified images generated by zooming in on the left, central, right parts of (b), respectively ($K_{max} = 5$). Moreover, (g), (h), (i) are three-fold magnified images of (d), (e), (f), respectively ($K_{max} = 6$). These figures clearly show a local resemblance, that is, similar structures appear repeatedly under magnification in the scale direction, while these structures differ depending on the position in the space direction with regard to height bumps and colors (cf. Sect. 3.3.3).

Figure 11 shows the effect of changing height values h_i. (a,b-1) are control meshes and continuous interpolation surfaces without applying IST's. (a,b-2) are composite SP-IST-Bézier surfaces produced by a quasi-fully-connected-IST, $\delta(\hat{\mathbf{u}}) = 0.4$, $K_{max} = 3$. (a,b-3) are

[†] Figures 9 to 13 are originally color images. For Figs. 9, 10, and 12, only the red intensity is extracted so that readers can understand the explanations through even the black-and-white printing. The original color images of (b), (e), and (h) of Fig. 10 are shown in the color printed page.

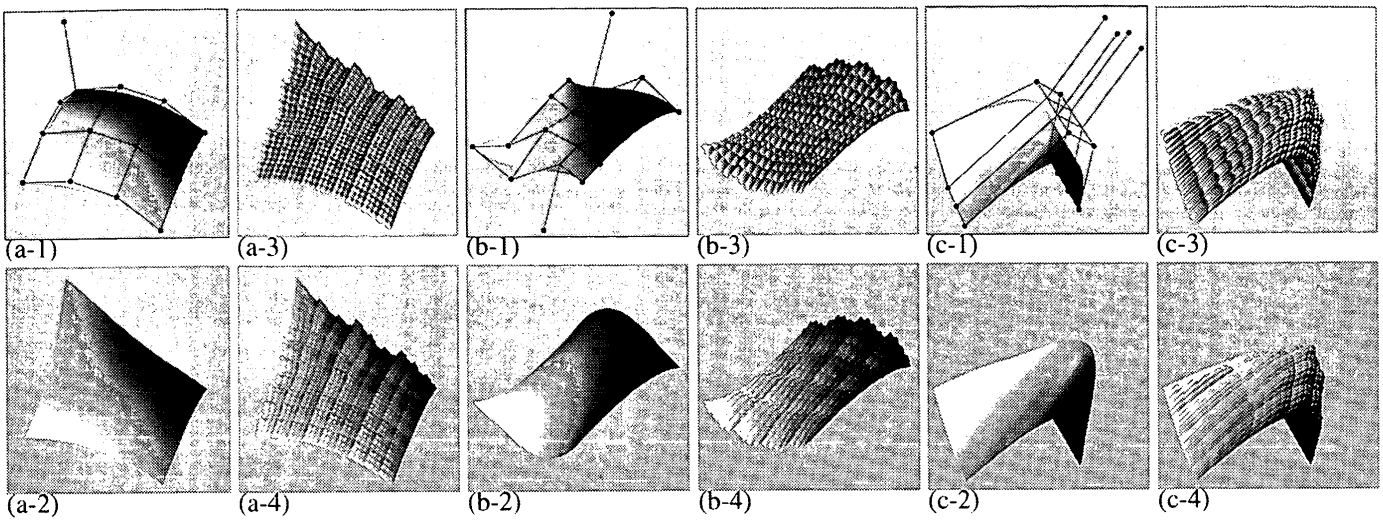


Fig. 9 Superposed IST-Bézier surfaces.

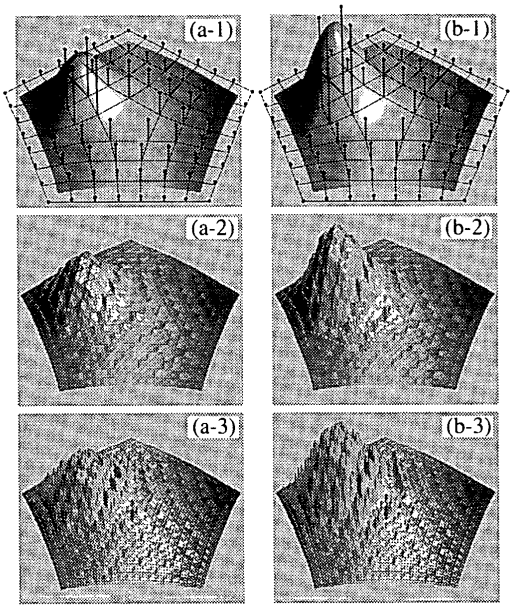


Fig. 11 Avalanche shape deformation.

the same, except for $\delta(\hat{\mathbf{u}}) = 0.8$. Comparing (a-1,2,3) with (b-1,2,3), we find that the change of height values h_i results in an attractive shape deformation, which is referred to as the *avalanche shape deformation*. This deformation is distinctive of the WR surface and is considered to be useful for shape modeling because the shape of a surface can be modeled and deformed in fractal-like fashion by intuitive and easy control. All a user has to do is to change the data of control points $\mathbf{r}_i$ intuitively: move the positions of $\mathbf{p}_i$, change height values h_i and their directions $\mathbf{n}_i$, and give desired attributes, such as colors $\mathbf{c}_i$ or transparencies t_i, to $\mathbf{r}_i$ and change them. In addition, the influence of the ratio function δ is understood by comparing (a-2) to (a-3), and (b-2) to (b-3). The larger the value of δ is, the rougher the surface becomes. It is naturally possible to make the value of δ differ on each position and give the roughness varying on the surface (actually, δ is defined as a function of $\mathbf{u}^*$ in Eq. (34)).

Comparing Fig. 9 with Figs. 10 and 11 shows us the difference between global and local effectiveness (cf. Sects. 3.3.2, 3.3.3, and 3.5). The latter tend to make the height bumps and the colors, which are given to the surfaces before IST iterations, stay around the original local positions even after IST iterations, whereas the former tends to spread them all over the surface globally. Based on this, a composite (SP-)IST-Bézier surface can be deformed locally on the part which is desired to be deformed as shown in Fig. 11.

Figure 12 shows the effect produced by different ex-shuffle mappings. The figures from (A-a) to (A-f) are the cases on a regular mesh, while those from (B-a) to (B-e) are on an irregular mesh. (A,B-a) show control meshs and base surfaces covered by differently colored Bézier patches. (A,B-b) show continuous interpolation surfaces without applying IST's. (A-c,d,e,f) are composite SP-IST-Bézier surfaces generated by ex-shuffle mappings of Fig. 6 (a,b,c,d), respectively. (B-c,d,e) are generated by Fig. 6 (b,c,d), respectively. They are all based on $\delta(\hat{\mathbf{u}}) = 0.66$, $K_{max} = 4$. From these figures, we can find that each ex-shuffle mapping determines the strength of the connection of the SP-IST-Bézier surfaces. (A-c) is generated by a fully-connected-IST and is given the most regular and uniform connection. On the other hand, (A-f) and (B-e) do not have any connection.

4.3 Shape Modeling Using Wrinkly Surface

We here present some shape model examples created by using composite SP-IST-Bézier surfaces. In order to produce their CG images, we converted the models to polygons and rendered them using Z buffer algorithm. When the polygons were created, we applied *adaptive sampling* [9]–[11] and *hierarchical sampling* [11]. In brief, the former determines the maximum iteration

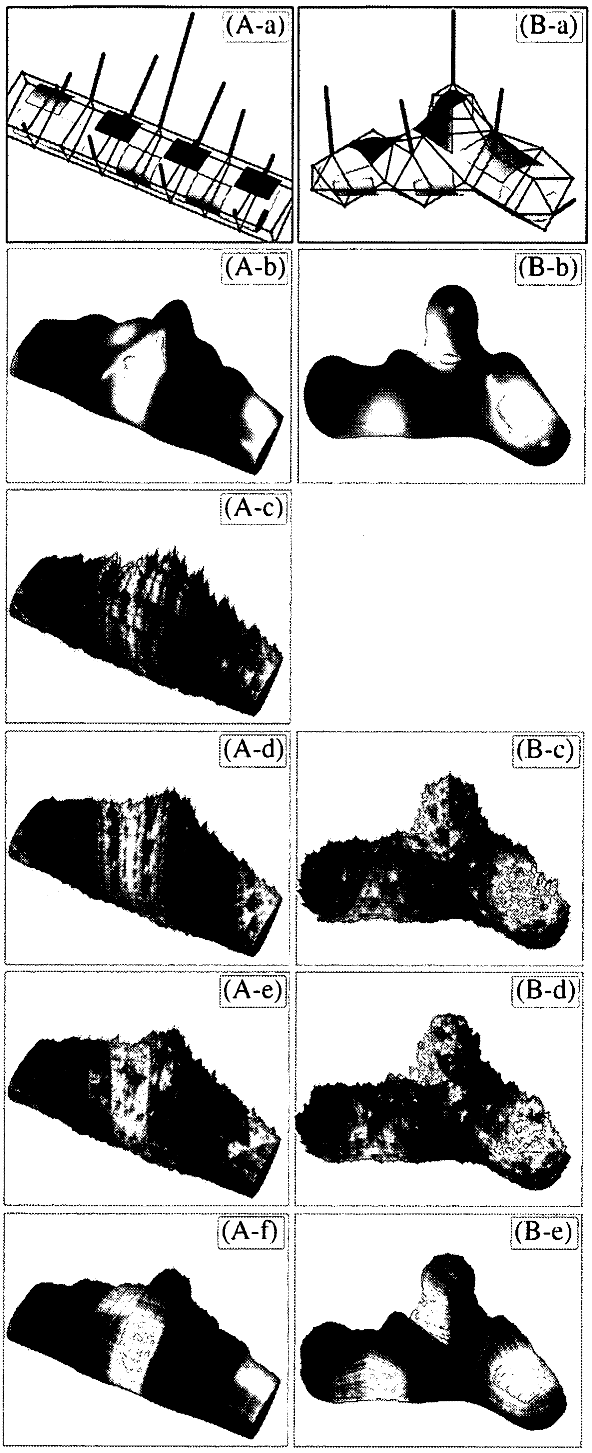


Fig. 12 Composite superposed IST-Bézier surfaces by different ex-shuffle mappings.

K_{max} in Eq. (33) based on the screen resolution for LOD display, and the latter samples a SP-WR surface based on every iteration step.

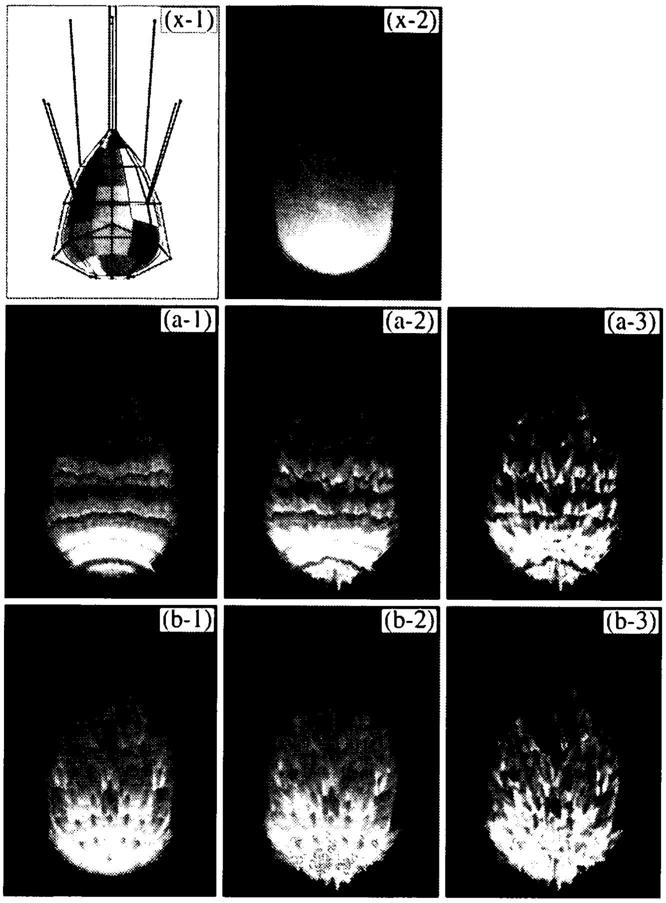


Fig. 13 "Fire."

Figure 13 is an example for showing an advantage due to the generalization of IST proposed in this paper compared with the previous method. (x-1) shows a regular control mesh of $\mathbf{r}_i$ and its base surface†. Each control point $\mathbf{r}_i$ is given a color value $\mathbf{c}_i$ and a transparency value t_i as well as a height value h_i and a displacement vector $\mathbf{n}_i$. Each surface in the remainder of Fig. 13 is rendered using these values. In particular, the transparency value of a point on the surface is calculated from t_i and the distance by which the ray from the eye point passes through the closed surface. (x-2) is a continuous interpolation surface based on (x-1) without applying an IST. (a-1,2,3) are composite SP-IST-Bézier surfaces generated by applying a fully-connected-IST to (x-2), while (b-1,2,3) are those by a quasi-fully-connected-IST. These are all given $\delta(\hat{\mathbf{u}}) = 0.75$ and $K_{max} = 3$. The color and transparency values on the surfaces are not superposed, that is, are determined by applying the IST's for only $K = K_{max}$. (a,b-2) are generated by giving randomness to sampled height values in the sampling process when polygons are created. Moreover, for (a,b-3), sampled transparency values are also given randomness. (a,b-1) are not given any randomness. Comparing (a-1,2,3) with (b-1,2,3), we find that the latter images look natural than the former ones. This is caused by the difference of the applied IST's. As shown in Fig. 6 (a), a two-dimensional fully-connected-IST essentially creates a regular and uniform lattice pattern, which results in making an artifact as shown in Fig. 13 (a-1,2,3). In these figures, the latitude and longitude directional pattern caused by the

†The Bézier patches on the top and bottom parts degenerate into triangles.

IST remains even after giving randomness as shown in (a-2,3). A two-dimensional fully-connected-IST can be defined as the tensor product of two one-dimensional fully-connected-IST's, which were proposed in the previous method [11]. Actually, in the previous method, the artifact had to be avoided by appropriately arranging the control points of a mesh and giving suitable displacement vectors. In contrast, (b-1,2,3) show that a quasi-fully-connected-IST, which is newly proposed in this paper, seems to be more suitable for the "fire" model. Although it also makes an artificial pattern, giving randomness reduces the impression of the pattern as shown in (b-2,3). We find that the local resemblance constructed on these surfaces contributes to the natural appearance of fire; a bigger part consists of smaller similar subparts, and this structure is repeated in the scale direction. In general, the variety of the transformation patterns achieved by the IST's generalized on multi-dimensional spaces makes it possible to create the local resemblance patterns suitable for various desired impressions. On the model of Fig. 13, the control points on the upper part are given redder color values and larger transparency values, while those on the lower part are given yellower ones and smaller ones. As a result, the fire-looking color pattern is created by the applied IST shuffling these color and transparency values in the local resemblance manner. This means that giving color and transparency values as well as height values and displacement vectors for an IST is so important for creating characteristic appearance. In addition, giving randomness in the sampling process is also important for natural appearance.

Figure 14 is an example of shape models generated on irregular meshes. (a,b,c-1) show control meshes, which are deformed from the same original mesh, and their base surfaces. In (a,b,c-2,3), the fire shapes are generated by the displacement of height values from the base surfaces that are represented as the dark objects in the fire. (a,b,c-2) are generated without applying IST's, while (a,b,c-3) are generated with applying a quasi-fully-connected-IST by giving the same conditions as Fig. 13 (b-1,2,3). Comparing the figures from (a-1,2,3) to (c-1,2,3), we find that changing height values h_i and displacement vectors $\mathbf{n}_i$ intuitively, shown as the sticks running out from the control meshes (a,b,c-1), brings local avalanche deformation on the surfaces with the appearance of fire maintained. Moreover, the deformation of the fire shape follows that of the base surface controlled by $\mathbf{p}_i$, which is traditional continuous deformation. In addition, as described about Fig. 13, the color pattern change controlled by color values $\mathbf{c}_i$ and transparency values t_i can contribute to the natural appearance of a moving fire as well as the shape deformations above. It may be possible that an animation of the moving fire is produced by making the deformations and the color pattern change proceed continuously.

Figure 15 is an example that consists of four composite SP-IST-Bézier surfaces, of which the irregular control meshes and base surfaces are (a), (b), (c), and (d). (e) is the composite image of these surfaces given the displacement of height values without applying IST's. (f) is the image obtained by rendering (e) using color and transparency values given to the control points. The surfaces in (g) and (h) are composite SP-IST-Bézier surfaces obtained by applying quasi-fully-connected-IST's to those in (f). The conditions of the four surfaces in (g) are $\delta(\hat{\mathbf{u}}) = 0.5, 0.5, 0.75, 0.5$ and $K_{max} = 2, 3, 3, 2$ in the order of the models (a), (b), (c), (d), respectively. The color values of the "mountain" model (b) and the transparency values of the "wave" model (c) are given the application of the IST's for only $K = K_{max}$. The conditions of (h) are the same as (g) except for $K_{max} = 3, 4, 4, 3$. Comparing (g) and (h), we find that the levels of detail of the models are controlled by the difference of K_{max}, which can be used for LOD display. Moreover, comparing the four models, we also find that WR surfaces can be given different impressions by appropriately setting $\mathbf{p}_i$, h_i, $\mathbf{n}_i$, $\mathbf{c}_i$, t_i, $\delta(\hat{\mathbf{u}})$, and other attributes if possible.

Concerning the methods to create "fire," "wave," or "water" model in general, a lot of studies have been reported such as [4], [5], [16]–[18], [20]. Particularly, the approach using "particle system" is one of the most familiar methods. This approach basically represents the objective model as a set of particles and calculates their proper motions automatically based on some given conditions. Then, final images are usually rendered using shading techniques, such as three-dimensional textures, for reality. On the other hand, the method proposed in this paper provides a modeling tool to create the shape of the model freely by trial and error. As shown above, a WR surface is intuitive and easy to control for obtaining an objective fractal-like shape by changing the data of control points. Compared with automatic generation methods such as particle system above, this method is useful when a certain shape is first desired. However, this method provides only surfaces, in other words, two-dimensional manifolds, and has the ability to produce only the exterior shape of an object. When we consider a wide use for representing various objects, this method has one serious drawback that it can not represent "three-dimensional solid"-like impressions, such as the spray of wave, well. For high reality, this method should be incorporated with other techniques such as two- or three-dimensional textures, shading models, or volume rendering techniques[†]. It may be possible to create a desired exterior shape using WR surfaces and add a detailed shape using particle system.

The images in Figs. 13, 14, and 15 were produced

[†] Actually, the qualities of the images in Figs. 13, 14, and 15 may look more or less poor. We did not give much elaboration to rendering, since the aim of this paper is to propose a method for shape modeling.

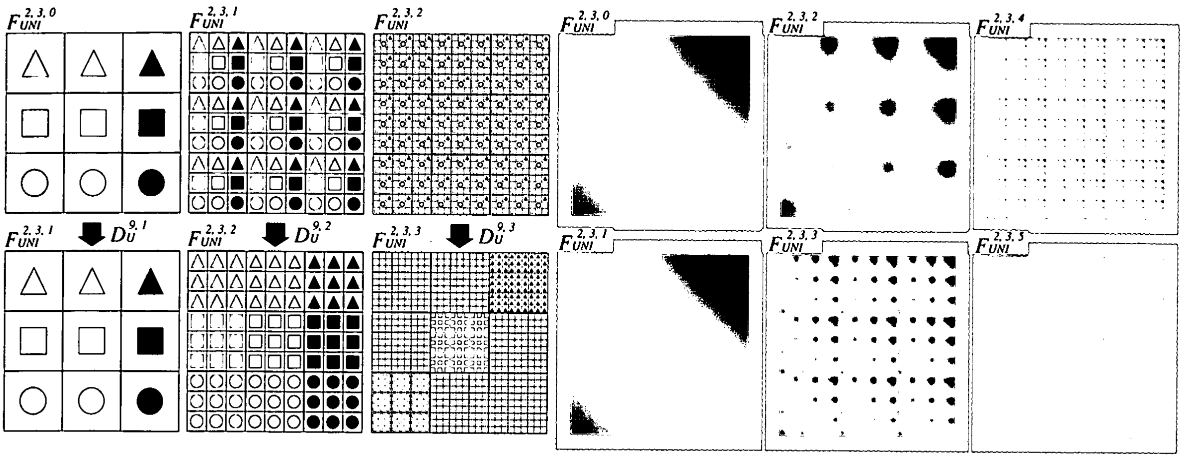


Fig. 4 Illustrations of a two-dimensional unit-IST on a unit cube.

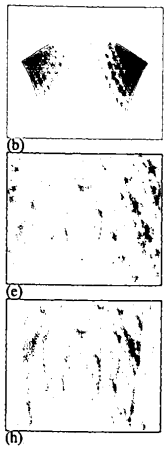


Fig. 10 (b), (e), (h)

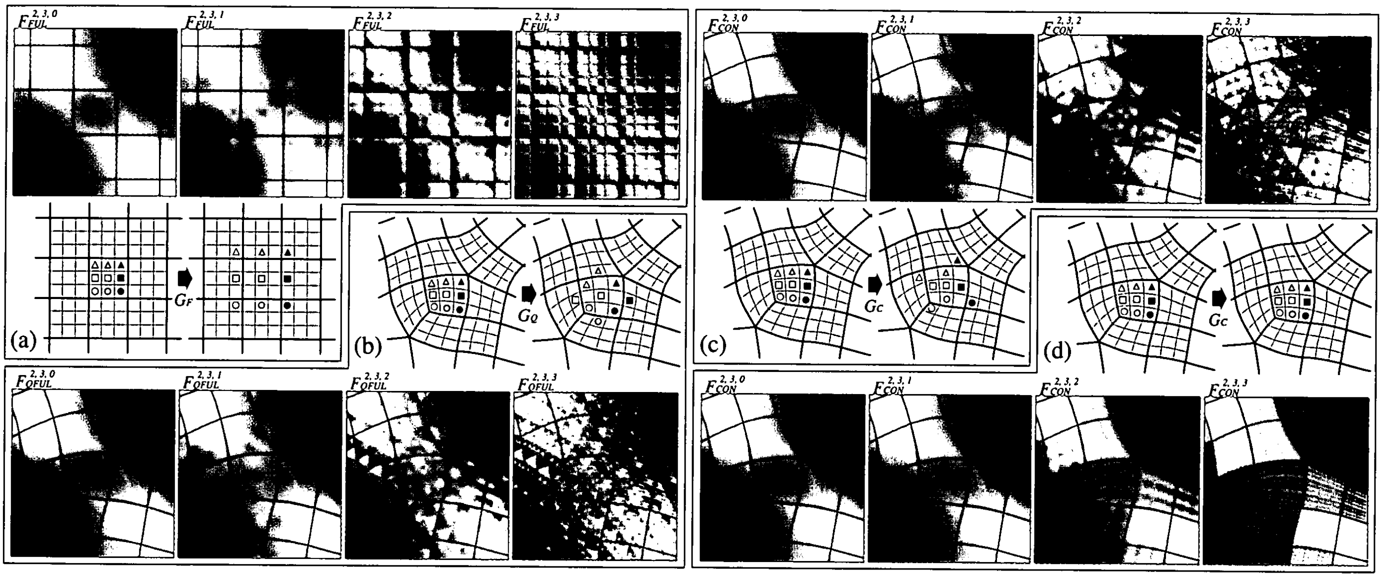


Fig. 6 Illustrations of two-dimensional connected-IST's on constructive spaces.

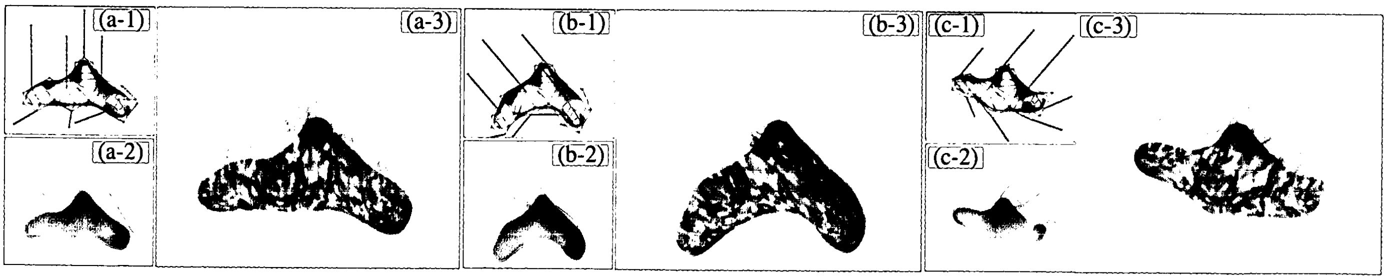


Fig. 14 "A moving object with fire."

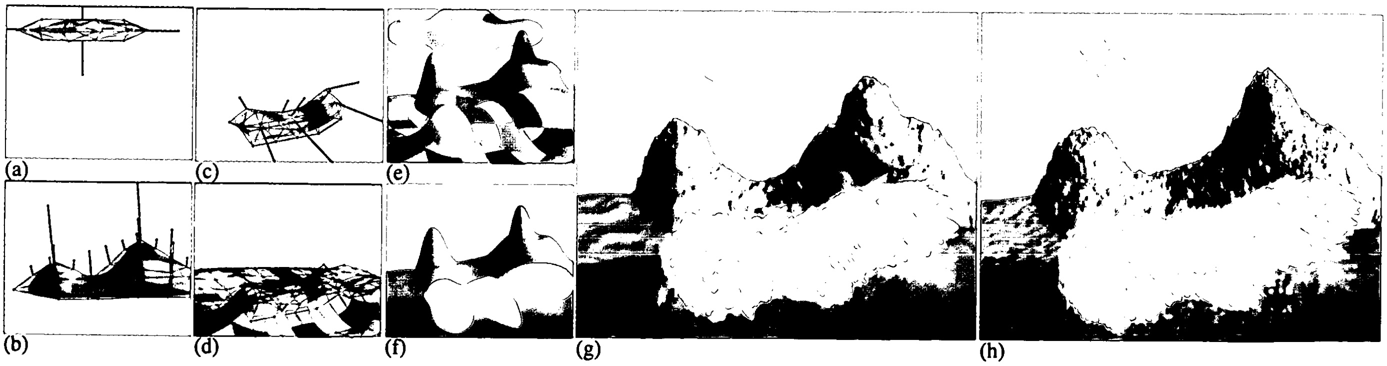


Fig. 15 "A landscape of cloud, mountain, wave, and sea."

Table 1 Execution time of Fig. 13 (second).

	(x-2)	(a-1)	(a-2)	(a-3)	(b-1)	(b-2)	(b-3)
sam.	3.09	9.33	9.37	9.34	9.40	9.28	9.34
pol.	0.42	0.43	0.39	0.40	0.44	0.41	0.41
ren.	11.22	18.20	23.46	23.44	15.28	20.91	20.92
total	14.73	27.96	33.22	33.18	25.12	30.60	30.67

Table 2 Execution time of Fig. 14 (second).

	(a-2)	(a-3)	(b-2)	(b-3)	(c-2)	(c-3)
sam.	2.79	7.52	2.73	7.44	2.73	7.47
pol.	0.34	0.30	0.32	0.32	0.31	0.29
ren.	12.01	15.49	11.38	15.95	9.80	15.40
total	15.14	23.31	14.43	23.71	12.84	23.16

Table 3 Execution time of Fig. 15 (second).

(f-1)

	(a)	(b)	(c)	(d)
sam.	0.25	3.39	2.81	2.03
pol.	0.03	0.37	0.62	0.17
ren.	2.91	6.23	6.30	5.10
total	3.19	9.99	9.73	7.30

(g)

	(a)	(b)	(c)	(d)
sam.	0.44	11.22	9.95	5.01
pol.	0.02	0.37	0.55	0.25
ren.	3.02	6.73	10.53	5.13
total	3.48	18.32	21.03	10.39

(f-2)

	(a)	(b)	(c)	(d)
sam.	1.69	28.69	23.40	14.22
pol.	0.24	3.38	5.51	1.82
ren.	4.05	14.52	14.46	9.26
total	5.98	46.59	43.37	25.30

(h)

	(a)	(b)	(c)	(d)
sam.	3.67	103.51	90.98	40.95
pol.	0.24	3.43	5.54	1.86
ren.	4.71	17.20	37.11	9.67
total	8.62	124.14	133.63	52.48

using Silicon Graphics 230 Visual Workstation 800 (Pentium III 800 MHz, 768 MByte). The execution time is shown in Tables 1, 2, and 3, where "sam.," "pol.," and "ren." denote the execution time for sampling, making polygons, and rendering, respectively. In Table 3, the columns from (a) to (d) correspond to the four models in Fig. 15. (f-1) and (f-2) show the time for obtaining (f) in the cases of using the same K_{max} as (g) and (h), respectively[†]. The resolution of the images of Fig. 13 is 900 × 1200, that of Fig. 14 is 1200 × 1000, and that of Fig. 15 is 1000 × 800. As shown in the tables, at present, real-time interactive operations are not realized. Achieving real-time performance is indispensable for an interactive modeling tool. First of all, most of the execution time is spent for the rendering process because of the inefficiency of a rendering program of our own making. Using a high performance rendering library, such as OpenGL, can improve the rendering speed dramatically. Besides, when K_{max} is large, the sampling process of WR surfaces becomes dominant, as shown in (b) and (c) of Table 3 (h). The development of an efficient sampling method is necessary for interactive performance.

5. Conclusion

In this paper, we have generalized the IST on code spaces, and then proposed multi-dimensional IST on geometric spaces. Based on them, we have proposed the construction scheme of composite (SP-)IST-Bézier surfaces on irregular meshes by connecting tensor product (SP-)IST-Bézier patches arbitrarily using two-dimensional connected-IST's. Through various examples, the WR surface has been shown to have the property "local resemblance" as a unified model of the interpolation surface and the fractal. And also, it has been demonstrated that we can easily control the appearance of a WR surface by changing control point data intuitively. As a result, we can say that the WR surface can be a useful modeling tool, especially for creating natural objects.

We finally have to say that the current WR surface model has limitations as a CG modeling tool because the shape that is possible to create is limited to a definite class based on tensor product patterns. The following topics are current research goals that we are pursuing.

- The realization of real-time interactive performance and the development of a practical modeling system based on the WR surface.
- A direct rendering technique for the WR surface.
- The generalization of unit-IST's to work on arbitrary two-dimensional n-gonal ($n \geq 3$) unit spaces, and the proposition of connected-IST's using the generalized unit-IST's.
- The construction scheme of the WR surface on arbitrary irregular meshes based on the above IST's.
- The expansion of the WR surface to three-dimensional volume model.
- More advanced model based on the IST for representing various kinds of shapes.

Acknowledgements

This work was supported partly by "A Support System for Region-specific R&D Activities" of Telecommunications Advancement Organization of Japan.

References

[1] M.F. Barnsley, "Fractal Functions and Interpolation," Constructive Approximation, vol.2, pp.303–329, 1986.

[2] M.F. Barnsley, A. Jacquin, F. Malassenet, L. Reuter, and

[†] In both cases, the shape of the continuous surface does not change. Therefore, we put only one image (f), which is the case corresponding to (h), in Fig. 15.

A.D. Sloan, "Harnessing Chaos for Image Synthesis," Computer Graphics (SIGGRAPH '88), vol.22, no.4, pp.131–140, 1988.

[3] M.F. Barnsley, Fractals Everywhere, 2nd ed., Academic Press, Boston, 1993.

[4] N. Chiba, K. Muraoka, H. Takahashi, and M. Miura, "Two-dimensional visual simulation of flames, smoke and the spread of fire," J. Visualization and Computer Animation, vol.5, pp.37–53, 1994.

[5] N. Chiba, S. Sanakanishi, K. Yokoyama, I. Ootawara, K. Muraoka, and N. Saito, "Visual simulation of water currents using a particle-based behavioural model," J. Visualization and Computer Animation, vol.6, pp.155–171, 1995.

[6] S. Demko, "Construction of fractal objects with iterated function systems," Computer Graphics (SIGGRAPH '85), vol.19, no.3, pp.271–278, 1985.

[7] T. DeRose, M. Kass, and T. Truong, "Subdivision surfaces in character animation," Computer Graphics (SIGGRAPH '98), pp.85–94, 1998.

[8] G.E. Farin, Curves and Surfaces for Computer Aided Geometric Design: A Practical Guide, 2nd ed., Academic Press, Boston, 1990.

[9] T. Fujimoto and Y. Ohno, "Wrinkly surfaces: Generating parametric surfaces by procedural interpolation," Trans. IPS. Japan, vol.39, no.7, pp.2168–2179, 1998.

[10] T. Fujimoto and Y. Ohno, "Generating wrinkly surfaces on constrained irregular mesh," Trans. IPS. Japan, vol.40, no.10, pp.3672–3684, 1999.

[11] T. Fujimoto and Y. Ohno, "Formalization and Superposed Construction of Wrinkly Surface," Trans. IPS. Japan, vol.41, no.9, pp.2518–2535, 2000.

[12] J.S. Geronimo and D. Hardin, "Fractal interpolation surfaces and a related 2-D multiresolution analysis," J. Mathematical Analysis and Applications, vol.176, pp.561–586, 1993.

[13] B.B. Mandelbrot, The fractal geometry of nature, W.H. Freeman and Company, New York, 1983.

[14] P.R. Massopust, "Fractal surfaces," J. Mathematical Analysis and Applications, vol.151, pp.275–290, 1990.

[15] H.O. Peitgen and D. Saupe, (Eds.), The Science of Fractal Images, Springer-Verlag, New York, 1988.

[16] W.T. Reeves, "Particle systems—A technique for modeling a class of fuzzy objects," Computer Graphics (SIGGRAPH '83), vol.17, no.3, pp.359–376, 1983.

[17] K. Sims, "Particle animation and rendering using data parallel computation," Computer Graphics (SIGGRAPH '90), vol.24, no.4, pp.405–413, 1990.

[18] J. Stam and E. Fiume, "Depicting fire and other gaseous phenomena using diffusion processes," Computer Graphics (SIGGRAPH '95), pp.129–136, 1995.

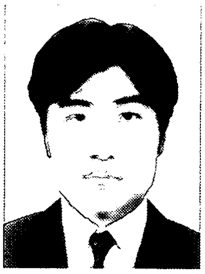
[19] R. Szeliski and D. Terzopoulos, "From splines to fractals," Computer Graphics (SIGGRAPH '89), vol.23, no.3, pp.51–60, 1989.

[20] J. Takahashi, H. Takahashi, and N. Chiba, "Image synthesis of flickering scenes including simulated flames," IEICE Trans. Inf. & Syst., vol.E80-D, no.11, pp.1102–1108, Nov. 1997.

[21] C.E. Zair and E. Tosan, "Fractal modeling using free form techniques," EUROGRAPHICS'96, vol.15, no.3, pp.269–278, 1996.

[22] C.E. Zair and E. Tosan, "Unified IFS-based model to generate smooth or fractal forms," Surface Fitting and Multiresolution Methods, pp.345–354, Vanderbilt University Press, 1997.

[23] N. Zhao, "Construction and application of fractal interpolation surfaces," The Visual Computer, vol.12, pp.132–146, 1996.



Tadahiro Fujimoto is currently a Research Associate in the Department of Computer Science at Iwate University, Japan. His research interests include computer graphics, geometric model, and mathematics for shape description in general. He received a BE in electrical engineering, and an ME and Ph.D. in computer science from Keio University, Japan, in 1990, 1992, and 2000, respectively. He worked at Mitsubishi Research Institute, Japan, from 1992 to 1995. Since 1999, he has been in the present post. He is a member of IPS of Japan, IEEE and ACM.

Yoshio Ohno received a BS and an MS in administration engineering from Keio University in 1968 and 1970, respectively. Following receipt of his MS, he did research at the Institute of Information Science at Keio University from 1970 to 1988. He received a Ph.D. in administration engineering from Keio University in 1986. He was an assistant professor from 1987 to 1994 and has been a professor at the Faculty of Science and Technology of Keio University since 1995. His research interests include computer graphics, CAGD, VR and DTP.

Kazunobu Muraoka is currently an Associate Professor in the Department of Computer Science at Iwate University. His research interests include computer graphics and computer animation. He received a BE in electronics engineering from Tohoku Institute of Technology in 1976 and a DE in electronics and information science from Iwate University in 1999. He was a research student at Tohoku Institute of Technology from 1976 to 1977, a Lecturer at Morioka Junior College from 1989 to 1991, and an Associate Professor from 1991 to 1999. He is a member of IPS of Japan, IEEE and ACM.

Norishige Chiba is currently a Professor in the Department of Computer Science at Iwate University. His research interests include computer graphics, algorithm theory and science on form. He received a BE in electrical engineering from Iwate University and an ME and DE in information engineering from Tohoku University in 1975, 1981 and 1984, respectively. He worked at Nippon Business Consultant Co., Ltd from 1975 to 1978. He was a Research Associate in the Department of Communication Engineering at Tohoku University from 1984 to 1986, an Associate Professor of Computer Science at Sendai National College of Technology from 1986 to 1987 and an Associate Professor of Computer Science at Iwate University from 1987 to 1991. He is a member of IPS of Japan, IEEE and ACM.