Modeling fractal shapes using generalizations of IFS techniques

Joëlle Thollot, Chems Eddine Zair, Eric Tosan, Denis Vandorpe

▶ **To cite this version:**

Joëlle Thollot, Chems Eddine Zair, Eric Tosan, Denis Vandorpe. Modeling fractal shapes using generalizations of IFS techniques. Fractals in Engineering, May 1997, Arcachon, France. inria-00510101

HAL Id: inria-00510101

https://inria.hal.science/inria-00510101

Submitted on 1 Dec 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Modeling fractal shapes using generalisations of IFS techniques

J. Thollot, C.E. Zair, E. Tosan, and D. Vandorpe

LIGIM - Université Claude Bernard
Bât 710 - 43, bd du 11 Novembre 1918
69622 Villeurbanne Cedex, FRANCE
Tel: 04.72.44.80.00 p 40.07 Fax: 04.72.43.13.12
{jthollot,czair}@ligim.univ-lyon1.fr

Summary. One of the major problems in geometric modeling is the control of shape construction. Indeed, one should be able to construct geometrical forms by combining or manipulating simple entities. This problem is even more important when we deal with fractal geometry. In this paper, we propose some methods for increasing the modeling capabilities of fractal shape constructions. We propose two extensions of the IFS model. The first is based on the integration of free form techniques in an IFS formalism, while the second deals with the definition of matrix of attractors to give a constructive approach of fractal shapes.

1. Introduction

The aim of geometric modeling is to model forms by manipulating or combining well-known shapes such as circles or boxes. The resulting model becomes more convenient if it allows us to control and to manipulate the final shape easily. When we deal with fractal geometric modeling, the control on fractal figures is not so easily achieved as with classical smooth ones. The difficulty comes from the fact that fractal shapes are usually generated with iterative or recursive procedures.

In order to have a better control on the shape, we wish to develop a *fractal modeler*. Such a modeler should be constituted by a set of basic shapes, a set of unary operations (shape modifications) and a set of binary operations (shape combinations). We distinguish two kinds of operations: set operations (union, intersection, restrictions) and geometric operations (transformations, deformations, interpolations, extrusions).

Our work focuses on extensions of the IFS model. We have been inspired by existing techniques in classical geometric modeling. One of these techniques is constructive geometry which allows to build complex attractors using set operations (union, intersection, ...). Such a construction is possible using IFS matrices which are able to generate a wide class of fractal shapes, not necessarily self-similar. These matrices can be combined with particular operations to yield complex fractal shapes in a constructive way. Another classical technique is the free form one. We demonstrate, by considering a particular complete metric space and particular IFS, that we can generate IFS attractors that are either smooth or fractal figures. These attractors combine the advantages of free forms (control via a collection of control points) and those of fractals (self-similarity, complex shapes). Thus, fractal shapes can be manipulated with the same flexibility as classical free forms.

2. IFS modeling

Fractals are characterized by their property of self-similarity. Barnsley's idea consists of using this property to encode the different parts of the fractal by contractive operators. These operators use an iterative scheme to transform any initial shape to a union of its reduced parts. The contraction property of the operators is used to prove the existence of a unique invariant set named the attractor. Below, we summarize the basic notions of IFS theory and introduce the notation used in the remainder of the paper.

An IFS-based modeling system is defined by a triple $(E, d, \mathcal{S})$ where:

- (E, d) is a complete metric space. We shall call it the *iterative space.*
- $\mathcal{S}$ is a semigroup acting on points of E such that : $p \mapsto T\ p$, where $T \in \mathcal{S}$ is a contractive operator. $\mathcal{S}$ is called the *iterative semigroup.*

The space $\mathcal{H}(E)$ denoting the set of non-empty compacts of E is a complete metric space, endowed with the Hausdorff distance :

$$d_H(K, K') = \max\{\max_{p \in K} \min_{q \in K'} d(p, q), \max_{q \in K'} \min_{p \in K} d(p, q)\}.$$

An IFS is a finite subset of the semigroup $\mathcal{S}$:

$$\mathcal{T} = \{T_1, ..., T_N\}.$$

The associated Hutchinson operator is:

$$K \in \mathcal{H}(E) \mapsto \mathcal{T}K = T_1\ K \cup ... \cup T_N\ K.$$

This operator is contractive in the metric space $(\mathcal{H}(E), d_H)$ and admits a fixed point $\mathcal{A}(\mathcal{T})$ given by :

$$\mathcal{A}(\mathcal{T}) = \lim_{k \to \infty} \mathcal{T}^k K_0.$$

IFS attractors constitute a particular family of compacts, they are *invariant sets* for the semigroup of IFSs :

$$\mathcal{A} = T_1\ \mathcal{A} \cup ... \cup T_N\ \mathcal{A}, \text{ with } T_i \in \mathcal{S}.$$

IFS techniques present an interesting approach for modeling fractals. In addition to the simplicity of their mathematical formalism, they provide an efficient tool to encode complex shapes with a small set of contractive operators. The structure of the attractor depends strongly on the semigroup of the operators. This dependency suggests a classification of fractal shapes into families. For instance, if the semigroup is taken to be of the following types:

- Affine operators: in this case, the attractors are self-affine.
- Homographies: here, the corresponding attractors are self-homographic (see Lévy Véhel [13]).
- Similitudes: the corresponding attractors are self-similar. A wide variety of fractal curves belongs to this family of attractors. (see Dekking [5], Dubuc [7], Tricot [27], Prusinkiewicz [24] and Hwang [12]).
- Dilatations-shearing : this semigroup permits graphs of functions to be defined as IFS attractors. The *fractal interpolation* of curves introduced by Barnsley [2] and extended to surfaces by Massopust [14] [15] gives examples of such functions.
- Dilatations: this semigroup was used by Dubuc [7] to construct *hyperbolic curves.*

This paper focuses on affine operators.

IFS techniques suffer from important drawbacks when used for geometric modeling purposes. These consist of two essential restrictions. Firstly, the attractors are strictly self-affine. Secondly, there is no convenient way to manipulate these attractors. By analogy with free form techniques, manipulation means the possibility of controlling the global shape of the attractor. Such a deficiency is an important limit of fractal geometric modeling.

We propose two generalisations of IFS techniques to overcome these drawbacks. First, we define the projected IFS attractor

$$K'_\omega = P\mathcal{A}(\mathcal{T}) \text{ with } K'_k = P(\mathcal{T}^k K_0),$$

where P denotes projection.
This generalisation enables the construction of attractors with control points which can be handled as efficiently as classical free form shapes (Bézier, spline,...).

Second, we define the restriction of an IFS attractor

$$K''_\omega \subseteq \mathcal{A}(\mathcal{T}) \text{ with } K''_k \subseteq \mathcal{T}^k K_0.$$

This generalisation extends the family of modeling shapes to non-restricted *self-affine* attractors. A set of operations will be given for a constructive approach of fractal attractors using basic operations, such as union and intersection.

3. IFS generalisations

3.1 Free form shapes

In this section, we present a generalisation of the IFS model which integrates free form techniques. This generalisation distinguishes between the iterative space, in which the attractor is defined, and the modeling space, in which this attractor will be projected using a set of control points. In this way, we can construct smooth or fractal forms characterized by a set of control points.

Let us take as iterative space the I-simplex defined by:

$$\mathcal{B}^I_+ = \{(\lambda_j)_{j\in I} |\ \lambda_j \geq 0\ ,\ \sum_{j\in I} \lambda_j = 1\ ,\ I = [0,...,m]\},$$

and Markov matrices as the iterative semigroup:

$$\mathcal{S}(I) = \{T |\ \sum_{j\in I} T_{ij} = 1 \text{ and } T_{ij} \geq 0\}.$$

We build IFS attractors which belong to the I-simplex: $\mathcal{A}(\mathcal{T}) \subseteq \mathcal{B}^I_+$. As we distinguish between the iterative space and the modeling space, we can project this attractor in the modeling space (R^2 or R^3) using a set of control points as follows:

$$P\mathcal{A}(\mathcal{T}) \quad = \quad \{P\lambda |\ \lambda \in \mathcal{A}(\mathcal{T})\} \tag{3.1}$$

where:

$$P\lambda = \sum_{j\in I} \lambda_j p_j \text{ with } P = (p_j)_{j\in I}. \tag{3.2}$$

As the attractor belongs to the I-simplex, we have the convex-hull property of the attractor:

$$P\mathcal{A}(\mathcal{T}) \subseteq \text{conv}(\{p_j | \ j \in I\}),$$

where $\text{conv}(\{p_j | \ j \in I\})$ is the bounding box of the p_i's.

In this way, we generate free form fractal shapes which are controlled with the same flexibility as classical free form shapes [33].

3.2 Matrix of IFS

Another way to generalize the IFS model is to construct subsets of an IFS attractor. This can be done using matrices of IFS. Such matrices are a generalisation of the Hutchinson operator. This approach has been developed by Peitgen, Jűrgen and Saupe in [23]. They consider an IFS matrix of size $n \times n$ to be an operator on the space of vectors of size n with compact components. This matrix has a fixed point vector, which is the attractor of the IFS matrix.

Here we propose a new type of attractor. By considering the space of matrices of size $n \times n$ with compact components as iterative space, we construct a fixed point matrix. This definition of an attractor allows us to simulate a large family of existing models. For more details see [28] [25].

More precisely, we write $M(\mathcal{H}(E))$ for the set of matrices of compacts: for $\tilde{K} \in M(\mathcal{H}(E))$,

$$\tilde{K}_{ij} \in \mathcal{H}(E) \cup \emptyset.$$

In the same way, we write $M(\mathcal{P}(\mathcal{T}))$ for the set of matrices of IFS: for $\tilde{H} \in M(\mathcal{P}(\mathcal{T}))$,

$$\tilde{H}_{ij} = \bigcup_{k \in I_{ij}} T_k \text{ or } \emptyset,$$

with $I_{ij} \subseteq \{1, \ldots, N\}$ and $T_k \in \mathcal{T}$.

Using the union and composition operators, as defined in the IFS model, we define the composition of an IFS matrix with a matrix of compacts:

$$(\tilde{H}\tilde{K})_{ij} = \bigcup_{k=1}^{n} \tilde{H}_{ik}\tilde{K}_{kj}.$$

In the same way, we define the composition of two IFS matrices:

$$(\tilde{H}\tilde{H}')_{ij} = \bigcup_{k=1}^{n} \tilde{H}_{ik}\tilde{H}'_{kj}.$$

It is now possible to define an attractor matrix, in the same way as for IFS. Indeed, given $\tilde{K}$ and $\tilde{K}' \in M(\mathcal{H}(E))$, the distance defined by

$$d_\infty(\tilde{K}, \tilde{K}') = \max_{(i,j) \in R} d_H(\tilde{K}_{ij}, \tilde{K}'_{ij})$$

is a distance on $M(\mathcal{H}(E))$. Moreover, given $\tilde{K} \in M(\mathcal{H}(E))$ and $\tilde{H} \in M(\mathcal{P}(\mathcal{T}))$, one can prove that, under certain hypotheses, the sequence $(\tilde{H}^k\tilde{K})_k$ is a Cauchy sequence on $(M(\mathcal{P}(\mathcal{T})), d_\infty)$.

We thus define the attractor matrix associated with an IFS matrix as the limit of the sequence:

$$\bar{A}(\bar{H}) = \lim_{k\to\infty} \bar{H}^k \bar{K}_0.$$

Moreover, instead of constructing an attractor matrix, one can consider the union of certain of its components:

$$\mathcal{A}(\bar{H}, Q_I, Q_F) = \bigcup_{i\in Q_I, j\in Q_J} \bar{A}(\bar{H})_{ij} \text{ with } Q_I, Q_F \subseteq \{1,\ldots,n\}.$$

Thus, a single matrix leads to many different shapes from which a choice can be made.

This type of union has a relation with regular languages.
Writing $\Sigma = \{1,\ldots N\}$ the alphabet associated with an IFS $\mathcal{T} = \{T_1,\ldots,T_N\}$ and $L \subseteq \Sigma^*$ a language over Σ, we have:

$$\bigcup_{i\in Q_I, j\in Q_J} \bar{A}(\bar{H})_{ij} = \mathcal{A}(L) = \lim_{k\to\infty} \{T_{\sigma_1}\ldots T_{\sigma_k}\}K_0.$$

Many fractal models have been developed using formal languages [6] [3] [22] [4] [19] as well as some IFS generalisations based on graphs [20] [1] [10]. Our approach enables us to create the same shapes as these approaches [21] [16] [25].

4. Set operations

The matrix model permits combination of two IFS matrices to obtain a new attractor matrix. We have studied two types of operations, some general operations on matrices, such as union, composition, intersection, tensor product [25]; and some operations associated with formal language theory, such as tensor sum, block matrix [29].

These operations provide us with a constructive approach. Indeed, it is possible to combine two IFS matrices and to obtain a resultant matrix that we combine with another one, and so on. Before illustrating this approach with an example, let us give a brief definition of each operation.

4.1 Matrices operations

The union and intersection operations are defined in the following way,

$$(\tilde{H} \cup \tilde{H}')_{ij} = \tilde{H}_{ij} \cup \tilde{H}'_{ij},$$

$$(\tilde{H} \cap \tilde{H}')_{ij} = \tilde{H}_{ij} \cap \tilde{H}'_{ij}.$$

Composition has already been defined:

$$(\tilde{H}\tilde{H}')_{ij} = \bigcup_k \tilde{H}_{ik}\tilde{H}'_{kj}.$$

Tensor product is a more interesting operation because it increases the size of the matrix:

$$\tilde{H}'' = \tilde{H} \otimes_\cup \tilde{H}' = \begin{pmatrix} \tilde{H}_{11} \cup \tilde{H}' & \ldots & \tilde{H}_{1n} \cup \tilde{H}' \\ \vdots & & \vdots \\ \tilde{H}_{n1} \cup \tilde{H}' & \ldots & \tilde{H}_{nn} \cup \tilde{H}' \end{pmatrix},$$

with $\tilde{H}''$ a square matrix of size $n.m$, and

$$\tilde{H}_{ij} \cup \tilde{H}' = \begin{pmatrix} \tilde{H}_{ij} \cup \tilde{H}'_{11} & \dots & \tilde{H}_{ij} \cup \tilde{H}'_{1m} \\ \vdots & & \vdots \\ \tilde{H}_{ij} \cup \tilde{H}'_{m1} & \dots & \tilde{H}_{ij} \cup \tilde{H}'_{mm} \end{pmatrix}.$$

4.2 Language operations

Tensor summation is an operation associated with language theory. This operation corresponds to the shuffle (denoted by $\sqcup$) of two regular languages. For this reason, we denote it "$\diamondsuit_{\sqcup}$". We define it by

$$\tilde{H}'' = \tilde{H} \diamondsuit_{\sqcup} \tilde{H}' = \begin{pmatrix} \widetilde{\mathrm{Id}}.\tilde{H}_{11} \cup \tilde{H}' & \widetilde{\mathrm{Id}}.\tilde{H}_{12} & \dots & \widetilde{\mathrm{Id}}.\tilde{H}_{1n} \\ \widetilde{\mathrm{Id}}.\tilde{H}_{21} & \widetilde{\mathrm{Id}}.\tilde{H}_{22} \cup \tilde{H}' & \dots & \widetilde{\mathrm{Id}}.\tilde{H}_{2n} \\ \vdots & & & \vdots \\ \widetilde{\mathrm{Id}}.\tilde{H}_{n1} & \widetilde{\mathrm{Id}}.\tilde{H}_{n2} & \dots & \widetilde{\mathrm{Id}}.\tilde{H}_{nn} \cup \tilde{H}' \end{pmatrix},$$

with $\tilde{H}''$ a square matrix of size $n.m$ and $\widetilde{\mathrm{Id}}$ the identity matrix of size m.

The block matrix operation is a simple combination of two matrices:

$$\tilde{H}'' = \tilde{H} \diamondsuit_{\cup} \tilde{H}' = \begin{pmatrix} \tilde{H} & \emptyset \\ \emptyset & \tilde{H}' \end{pmatrix}.$$

As for the tensor summation, this operation has been inspired by language theory. It corresponds to the union of two regular languages. Using this operation we can define the union of two fractal shapes:

$$\mathcal{A}(\tilde{H} \diamondsuit_{\cup} \tilde{H}', Q_I \cup Q'_I, Q_F \cup Q'' F) = \mathcal{A}(\tilde{H}, Q_I, Q_F) \cup \mathcal{A}(\tilde{H}', Q'_I, Q'_F)$$

Moreover, if we replace some $\emptyset$ with the identity in the resulting matrix, we obtain the concatenation of two regular languages. We will denote this operation by "$\diamondsuit.$".

Figure 4.1 shows an example of concatenation on the left and an example of shuffle on the right.

4.3 Constructive approach

Figure 4.2 shows how some complex shapes can be constructed, using the "step by step" approach, with the use of the matrix operations. The construction tree is based on four simple IFS attractors.

The color plate C.2 shows a picture obtained by this approach.

4.4 Classification by inclusion

To facilitate the choice of operations, we have ordered (in the sense of set inclusion) the results of these operations. We consider the compact union of all components of the attractor matrix:

$$\bigcup_{i,j \in \{1,\dots n\}} \tilde{A}(\tilde{H})_{ij}.$$

Denoting by $\mathcal{T}$ the IFS composed of all the transformations used in both IFS matrices $\tilde{H}_1$ and $\tilde{H}_2$, we can prove the following inclusions [25], where $\downarrow$ denotes inclusion.

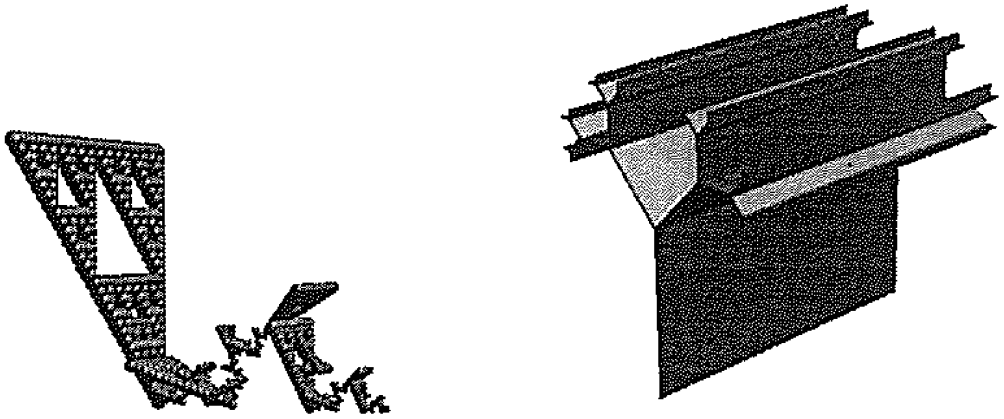


Fig. 4.1. Language operations

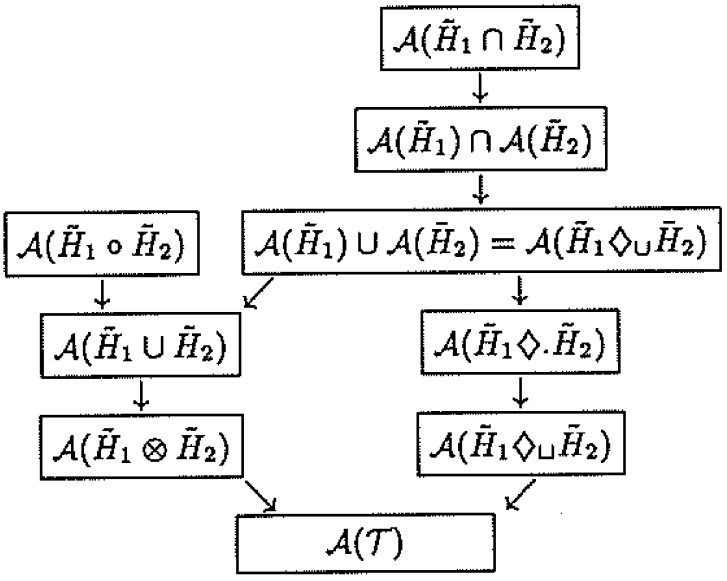


Figure 4.3 shows an example of inclusion.

5. Geometrical operations

5.1 Shape operations

In classical geometric modeling, basic operations are defined on shapes to provide sufficient capability of modeling. Such basic operations can for instance be a deformation or a combination of two elementary shapes, to generate a complicated one.

Two types of operations may be distinguished: unary operations (transformation or deformation of a shape):

$$F(A) = \{F(p) | \, p \in A\},$$

and binary operations (combination of shapes):

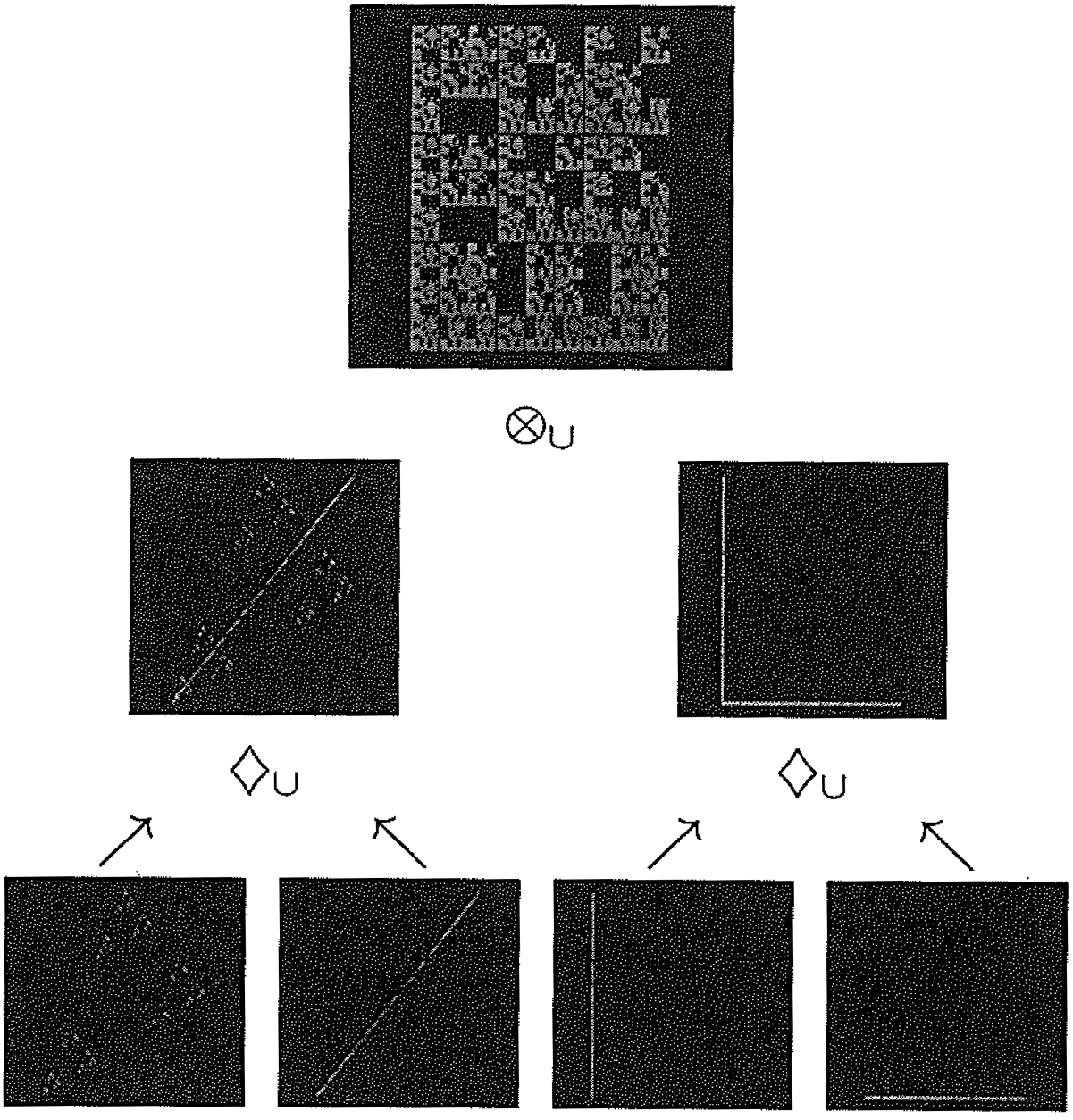
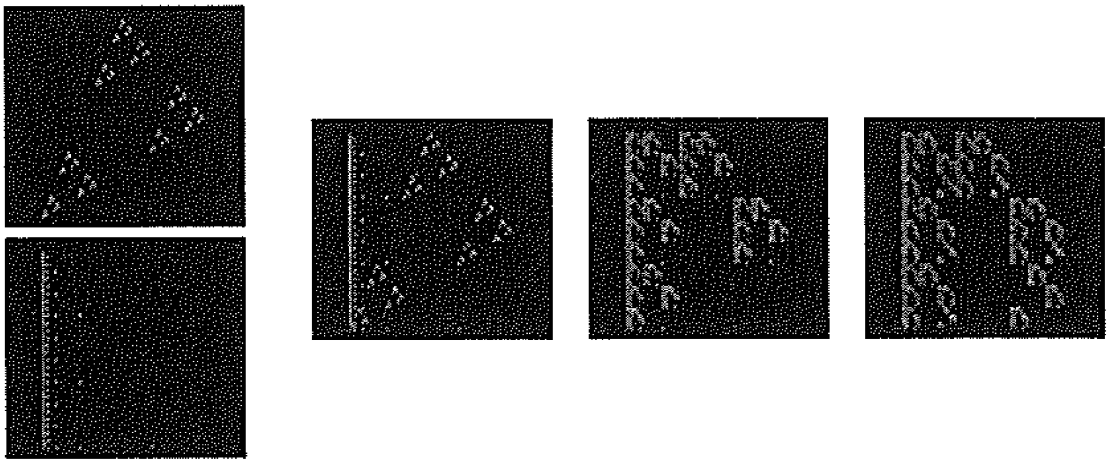


Fig. 4.2. Construction tree



$\mathcal{A}(\tilde{H}_1)$ and $\mathcal{A}(\tilde{H}_2)$ $\subset \mathcal{A}(\tilde{H}_1 \diamondsuit_\cup \tilde{H}_2)$ $\subset \mathcal{A}(\tilde{H}_1 \cup \tilde{H}_2)$ $\subset \mathcal{A}(\tilde{H}_1 \otimes_\cup \tilde{H}_2)$

Fig. 4.3. Example of inclusion.

$$A \# A' = \{p \# p' |\ p \in A \text{ and } p' \in A'\}.$$

In classical geometric modeling, curves and surfaces can be obtained by using the two sorts of geometric operations. Parametric curves can be formulated as deformation (or embedding) of the standard interval $[0,1]$ in the modeling space E:

$$A = F([0,1]) = \{F(s) |\ s \in [0,1]\}.$$

Parametric surfaces can be formulated as deformation (or embedding) of the standard set $[0,1]^2$. A classical procedure for patchwork construction is extrusion, i.e. combination of parametric curves :

$$F''(s,t) = F(s) \# F'(t).$$

Such operations cannot be achieved on IFS attractors, since a combination or a deformation of an attractor is not necessarily an IFS attractor. To solve this problem, we introduce the notion of morphism [26]. This allows operations on attractors to be considered as operations on IFS operators [8].

5.2 Operations on attractors

Let us consider three IFS modeling systems $(E,d,\mathcal{S}),(E',d',\mathcal{S}'),(E'',d'',\mathcal{S}'')$. A *unary morphism* is a pair of functions $F : E \to E'$ and $\hat{F} : \mathcal{S} \to \mathcal{S}'$ such that :

$$F(Tp) = \hat{F}(T)F(p).$$

A *binary morphism* is a pair of binary operations $\# : E \times E' \to E''$ and $\hat{\#} : \mathcal{S} \times \mathcal{S}' \to \mathcal{S}''$ such that:

$$(Tp)\#(T'p') = (T\hat{\#}T')(p\#p').$$

Using these morphisms, operations on attractors correspond to operations on the IFS operators. Indeed, if we take an IFS $\mathcal{T} = \{T_1, ..., T_N\}$ and its corresponding attractor $\mathcal{A}$ then:

$$F(\mathcal{A}) = F(\cup_{i=1}^{N} T_i\,\mathcal{A}) = \cup_{i=1}^{N} \hat{F}(T_i)\,F(\mathcal{A}). \tag{5.1}$$

Hence, the deformation of the attractor $\mathcal{A}$ using F is also an IFS attractor. Its corresponding IFS is $\hat{F}(\mathcal{T}) = \{\hat{F}(T_i)\,,\ i = 1, ..., N\}$.

These morphisms can be viewed as functional equations, where, for a function F and a couple of IFSs $\{T_i\}$ and $\{T_i^{'}\}$, we have

$$\forall i = 1, .., N\,,\ \forall p \in E\ \ F(T_i\,p) = T_i^{'} F(p). \tag{5.2}$$

It is interesting to notice that there are two ways to solve this functional equation:

1. Given a function F, is there a pair of IFSs such that the functional equation holds ? This leads to polynomial deformation of attractors.
2. Given a pair of IFSs, is there a function F which is a solution of the functional equation ? We demonstrate that, with certain conditions on the IFS, such a function F exists and allows a fractal deformation of IFS attractors.

5.3 Curve deformations

Let us investigate more precisely the functional equation (5.2). Our purpose is to define, using this functional equation, parametric curves which are IFS attractors. Suppose that $E = [0, 1]$, on which we define contractive affine operators

$$\mathcal{S} = \{T = [a, u]\ |\ \forall t \in [0, 1]\,,\ T\,t = at + u\}.$$

If T_0 and T_1 are two operators such that:

$$T_0\,t = rt;\ \ T_1\,t = (1 - r)t + r,\ \ 0 < r < 1, \tag{5.3}$$

then, obviously, $[0, 1]$ is the attractor of this IFS since we can write :

$$[0, 1] = T_0\,[0, 1] \cup T_1\,[0, 1].$$

Thus, if a function $\hat{F}$ exists, then $F([0, 1])$ is the attractor of the IFS $\hat{F}(\mathcal{T}) = \{\hat{F}(T_0), \hat{F}(T_1)\}$ which is a self-affine curve defined in the space $E^{'}$.

This development represents a general framework for modeling curves using IFS [31]. Next we give some convenient hypotheses for generating classical (Bézier, spline) and fractal free form shapes (see Figure 5.1).

5.3.1 Polynomial deformation. Let $E^{'} = B_+^i$ and $\mathcal{S}^{'} = \mathcal{S}(I)$. Given a function $F(t) = [F_0(t), ..., F_n(t)]^T \in E^{'}$, we are interested in the explicit formulation of the function $\hat{F}$. This establishes a correspondence between attractors in the space E and attractors in $E^{'}$. If such a function exists, then the curve $F([0, 1])$ is the attractor of the IFS $\hat{F}(\mathcal{T}) = \{\hat{F}(T_0), \hat{F}(T_1)\}$. Its projection $P\,F([0, 1])$ yields a parametric free form curve [32].
For example, if we take $F(t) = B(t)$, the set of Bernstein functions, then we can explicitly define the function $\hat{F}$ as:

$$\begin{array}{rccl} \hat{F}: & \mathcal{S} & \longrightarrow & \mathcal{S}^{'} \\ & T = [u, a] & \longmapsto & \hat{F}(T) = \hat{F}(u, a) = T_1^{'}(u)\,T_0^{'}(a/(1 - u)) \end{array}$$

where

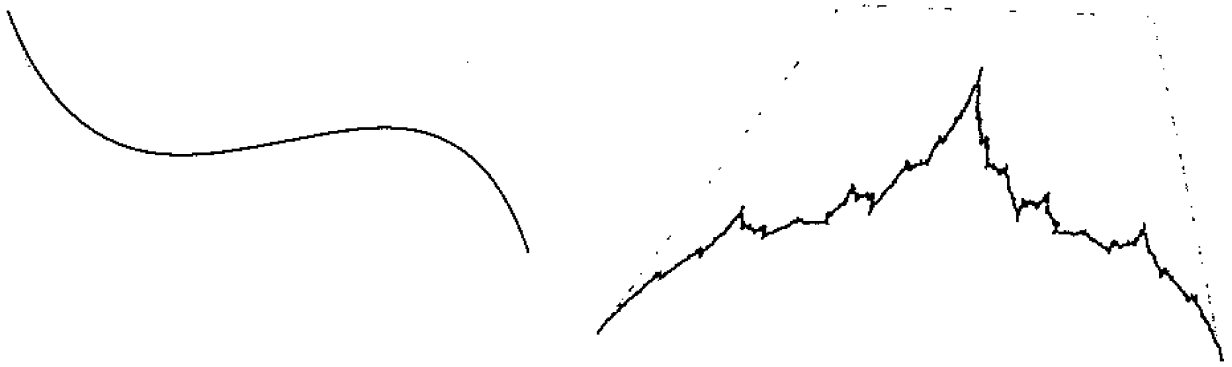


Fig. 5.1. Bézier and fractal curves obtained as IFS attractors. The fractal curve is controlled in the same way as the Bézier curve.

$$T_0' = \begin{pmatrix} 1 & 1-r & (1-r)^2 & (1-r)^3 \\ 0 & r & 2(1-r) & 3r(1-r)^2 \\ 0 & 0 & r^2 & 3(1-r)r^2 \\ 0 & 0 & 0 & r^3 \end{pmatrix},$$

$$T_1' = \begin{pmatrix} (1-r)^3 & 0 & 0 & 0 \\ 3(1-r)^2 r & (1-r)^2 & 0 & 0 \\ 3r^2(1-r) & 2(1-r) & 1-r & 0 \\ r^3 & r^2 & r & 1 \end{pmatrix}.$$

In this way, polynomial deformation of $[0,1]$ yields a classical Bézier curve as the projection of an IFS attractor (see Figure 5.1).

5.3.2 Fractal deformation. It is also interesting to study the inverse procedure to that discussed in (5.2). Suppose that the function F is unknown. Then, for two given IFSs $\mathcal{T} = \{T_0, T_1\}$ (defined as previously) and $\mathcal{T}' = \{T_0', T_1'\} \in \mathcal{S}'$, there is a necessary and sufficient condition for the existence of F. Indeed, if we choose the Markov matrices T_0', T_1' such that $T_0' c_1 = T_1' c_0$ (c_0, c_1 are respectively the fixed points of T_0' and T_1') then a unique function F exists, which is the solution of the functional equation (5.2) [18] [11] [30] [33].

Experiments show that, with general Markov matrices satisfying the condition above, the curves we obtain are fractals, due to the fact that they are nowhere differentiable with a self-similar structure (from the IFS model itself). Thus we see the intimate relationship between the Markov matrices and the smoothness of the attractor (see Figure 5.1).

5.4 Combinations of attractors

Binary geometric operations can be defined on two elementary IFS attractors to create a more complicated one. We can distinguish two types of operations:

- linear operations, such as affine combination or vectorial summation;
- bilinear operations, such as the tensor product.

Both types are discussed below.

5.4.1 Vectorial summations. Under certain conditions, a "special summation" [9] [8] [26], can be associated with vectorial summation of points such that:

$$(Tp) + (T'p') = (T\hat{+}T')(p + p').$$

Summation of attractors is defined by:

$$\mathcal{A}(T) + \mathcal{A}(T') = \mathcal{A}(T\hat{+}T').$$

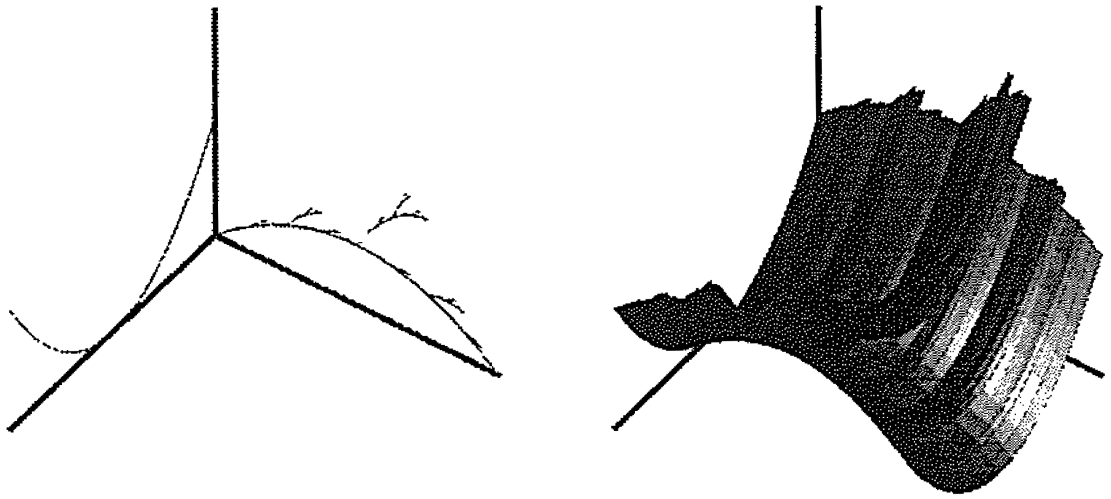


Fig. 5.2. Summation of a tree and a parabola.

Figure 5.2 displays the sum of two attractors, a tree and a parabola. The color plate C.3 displays an other sum of two bidimensional attractors.

5.4.2 Tensor products. A classical method of defining free surface patches is to use tensor products of curves. Given two functions $(F(s), F'(t))$, the free form surface obtained by the tensor product of these two functions is:

$$Q(s,t) = P(F(s) \otimes F'(t)) = \sum_{i=0}^{m} \sum_{j=0}^{m'} F_i(s) F'_j(t) p_{ij} \text{ with } P = (p_{ij}). \tag{5.4}$$

This operation has the following advantage: from two IFS corresponding to two parametric curves, we can find the IFS whose attractor is the surface obtained by the tensor product of these two curves. Indeed, the following property holds [17] [32] [33]:

$$P(\mathcal{A}(T) \otimes \mathcal{A}(T')) = P\mathcal{A}(T \otimes T'),$$

where $T \otimes T'$ is defined as the tensor product of Markov matrices.

There are also two cases of curve deformation: polynomial deformation and fractal deformation. In general, $\hat{F}$ can be defined explicitly in the case of Bézier deformation by:

$$\begin{array}{rccc} \hat{F}: & S \simeq S \otimes S & \longrightarrow & S' \otimes S' \\ & T = \left[\begin{array}{c|cc} u & a & 0 \\ v & 0 & d \end{array}\right] & \longmapsto & \hat{F}(T) = \hat{F}(u,a) \otimes \hat{F}(v,d) \end{array}.$$

We can use this function for fractal deformation by replacing the matrices T_0', T_1' by general Markov matrices (see Figure 5.3).

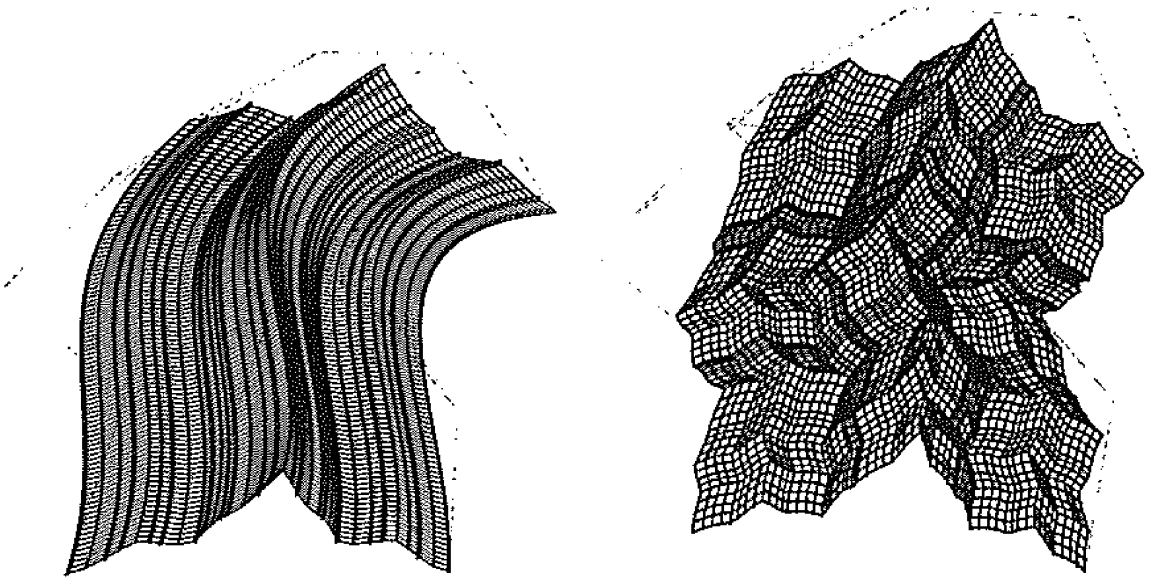


Fig. 5.3. The left figure illustrates a surface obtained by the tensor product of a Bézier curve with a fractal one. The right figure shows the tensor product of two fractal curves.

5.5 Deformations of attractors

The procedure of constructing free form fractals is based essentially on the property which associates another subdivision in E' to each subdivision in the space E through a morphism. In this way, all IFS attractors defined in E can be deformed in E' using the tensor product property. This leads to a straightforward way to construct IFS attractors with control points.

For example, to generate a Sierpinski gasket with control points, we take the Sierpinski IFSs in the spaces E and E':

$$\mathcal{T}_S = \{T_{00}, T_{01}, T_{10}\},\ \mathcal{T}_S' = \{T_{00}', T_{01}', T_{10}'\},$$

where $T_{ij}' = T_i' \otimes T_j'$.
The deformed attractor is a subset of the surface defined above:

$$\mathcal{A}(\mathcal{T}_S) \subset [0,1]^2 \ \Rightarrow\ \mathcal{A}(\hat{F}(\mathcal{T}_S)) = F(\mathcal{A}(\mathcal{T}_S)) \subset F([0,1]^2).$$

This consists of subdividing the surface into four patches and then omitting one of them (see figure 5.4). However, the generalisation to other IFS attractors can be done by giving the IFS in the space E and then computing its image under the function $\hat{F}$ to yield the IFS in the space E' [33].

Remark:
Previously, we dealt only with polynomial deformation. Such a deformation allows

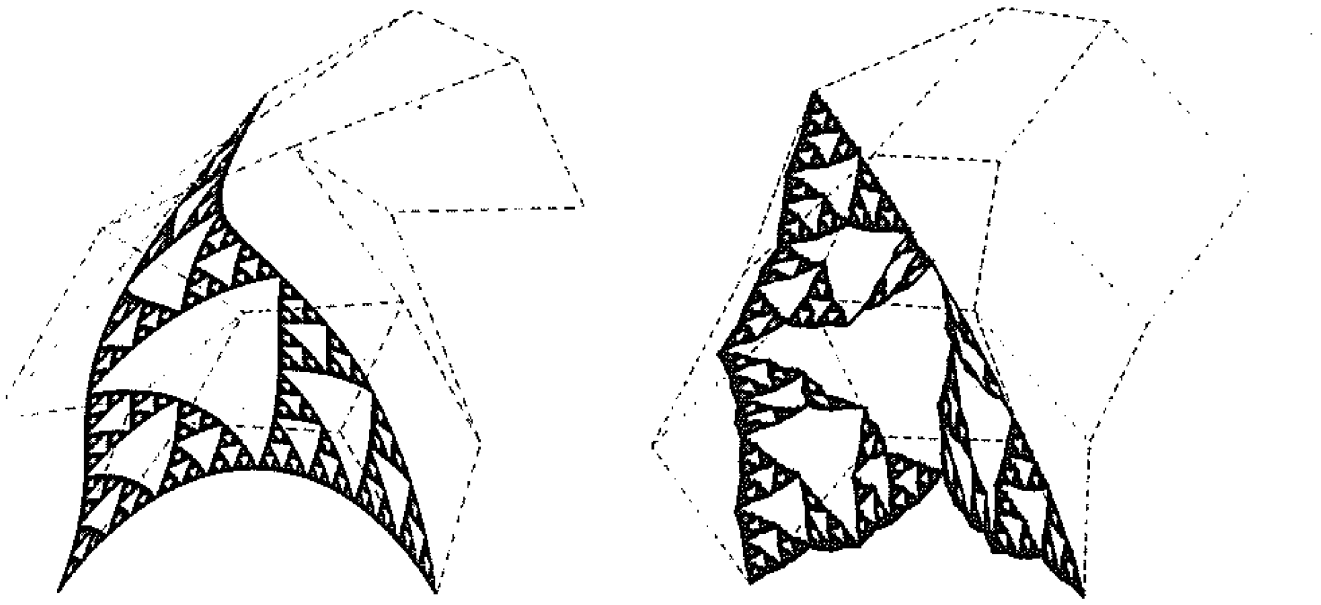


Fig. 5.4. These figures illustrate smooth and fractal deformation of the Sierpinski gasket.

us to construct attractors with control points. However, other types of deformations can also be used. Figure 5.5 illustrates an example of deformation using quadratic functions (elliptic paraboloid and hyperbolic paraboloid) [33].

6. Conclusion

Using generalisations of the classical IFS model, we construct a fractal modeler which manipulates IFS attractors with the same flexibility as classical smooth shapes. However, this modeler manipulates matrix of attractors and free form fractals in a disjoint way. Free form fractals are strictly classical IFS attractors. As an immediate extension of this work, we suggest a further generalisation to construct free form matrices of attractors.

References

1. Bandt, C. (1989): Topological Markov Chaïne and Mixed Self-Similar Sets. *Math. Nachr.*, (142), 107–123.
2. Barnsley, M.F. (1988): *Fractal Everywhere.* Academic press, INC.
3. Berstel, J., Morcrette, M. (1989): Compact Representation of Patterns by Finite Automata. In *Proceedings of Pixim*, 387–395.
4. Culik II, K., Dube, S. (1993): Rationnal and affine expressions for image synthesis. *Discrete Appl. Math.*, 41, 85–120, 1993.

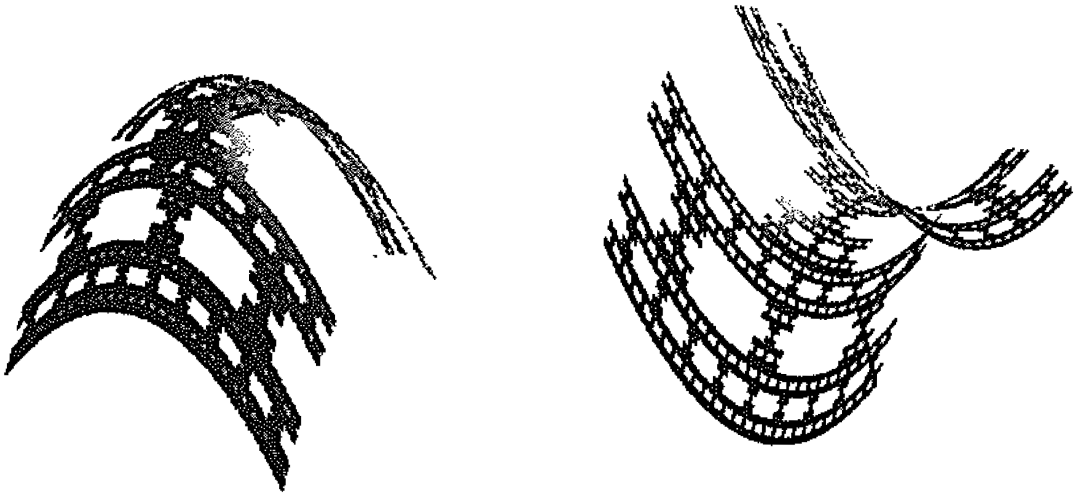


Fig. 5.5. Two embeddings of planar attractor in elliptic paraboloid and in hyperbolic paraboloid.

5. Dekking, F.M. (1980): Variations on Peano. *Nieuw Archief Voor Wiskunde*, (XXVIII), 275–281.
6. Dekking, F.M. (April 1982): Recurrent sets. *Advances in Mathematics*, 44(1), 78–104.
7. Dubuc, S. (1987): Modèles de courbes irrégulières. In *Dimensions Non Entières et Applications*. Masson.
8. Gentil, C. (March 1992): *Les fractales en synthèse d'images: le modèle IFS*. Thèse de doctorat, Université LYON I.
9. Gentil, C., Tosan, E. (December 1992): Description of fractals with IFS. In *COMPUGRAPHICS'92*, 214–221, Lisbon, Portugal. Portuguese ACM Chapter.
10. Hart, J. (May 1992): The object instancing paradigm for linear fractal modeling. In *Proceedings of Graphics Interface'92*.
11. Hata, M. (1991): Topological aspects of self-similar sets and singular functions. In J. Belair and S. Dubuc, editors, *Fractal Geometry and Analysis*, 255–276. Kluwer Academic Publishers.
12. Hwang, S.C., Yang, H.S. (1993): Discrete Approximation of the Koch Curve. *Computer & Graphics*, 17(1), 95–102.
13. Lévy Véhel, J. (1988): Analyse et synthèse d'objets bi-dimensionnels par des méthodes stochastiques. Thèse de doctorat, Université de Paris-sud centre d'Orsay, INRIA.
14. Massopust, P. (April 1990): Fractal surfaces. *Journal of Mathematical Analysis and Applications*, (151), 275–290.
15. Massopust, P. (1994) : *Fractal function and wavelets*. Academic Press.
16. Morcrette, M. (1996): Sur l'équivalence de descriptions de figures itérées. À pararaître dans Theoritical Computer Science.
17. Micchelli, C.A., Prautzsch, H. (1987): Computing surfaces invariant under subdivision. *Computer Aided Geometric Design*, (4), 321–328.
18. Micchelli, C.A., Prautzsch, H. (1989): Union Refinement of Curves. *Linear Algebra and its Applications*, (114-115), 841–870.
19. Merzenich, W., Staiger, L. (1994): Fractals,dimension, and formal languages. *Theoritical Informatics and Application*, 28(3-4), 361–386.

20. Mauldin, R.D., Williams, L.C. (1988): Hausdorff dimension in graph directed constructions. *Trans. Amer. Math. Soc.*, 309, 811–829.
21. Prusinkiewicz, P., Hammel, M.S. (1991): Automata, languages and iterated function systems. In *lecture notes for the SIGGRAPH'91 course : "Fractal modeling in 3D computer graphics and imagery"*.
22. Prusinkiewicz, P., Hammel, M.S. (May 1992): Escape-time Visualization Method for Language-restricted Iterated Function Systems. In *Proceedings of Graphics Interface'92*.
23. Peitgen, H.O., Jurgens, H., Saupe, D. (1992): Encoding images by simple transformations. In *Fractals For The Classroom, New York*.
24. Prusinkiewicz, P., Sandness, G. (Novembre 1988): Koch curves as attractors and repellers. *IEEE Computer Graphics & Applications*, 27–40.
25. Thollot, J. (September 1996): *Extension du modèle IFS pour une géométrie fractale constructive.* Thèse de doctorat, Université Claude Bernard Lyon 1.
26. Tosan, E. (September 1996): *Une Algèbre de Formes fractales utilisant les IFS.* Thèse de doctorat, Université Claude Bernard Lyon 1.
27. Tricot, C. (1993): *Courbes et dimension fractale.* Springer Verlag.
28. Thollot, J., Tosan, E. (December 1993): Construction of fractales using formal languages and matrices of attractors. In Harold P. Santos, editor, *COMPUGRAPHICS'93*, 74–81, Technical University of Lisbon.
29. Thollot, J., Tosan, E. (May 1995): Constructive Fractal Geometry : constructive approach to fractal modeling using language operations. In *GRAPHICS INTERFACE'95, Quebec, Canada*, 196–203.
30. van Overveld, C.W.A.M. (1990): Family of recursively defined curves ,related to the cubic Bézier curve. *Computer-Aided Design*, 22(9), 591–597.
31. Zair, C.E., Tosan, E.: Computer Aided Geometric Design with IFS techniques. In *FRACTALS'97, Denver, USA*, April. to appear.
32. Zair, C.E., Tosan, E. (November 1994): Définition de courbes et surfaces lisses ou fractales à l'aide d'IFS. In *AFIG 94, Toulouse*, 189–200.
33. Zair, C.E., Tosan, E. (August 1996): Fractal modeling using free form techniques. *Computer Graphics Forum*, 15(3), 269–278. EUROGRAPHICS'96 Conference issue.