



HAL
open science

Set manipulations of fractal objects using matrices of IFS

Joëlle Thollot

► **To cite this version:**

Joëlle Thollot. Set manipulations of fractal objects using matrices of IFS. Discrete Geometry for Computer Imagery (DGCI'97), Dec 1997, Montpellier, France. inria-00510104

HAL Id: inria-00510104

<https://inria.hal.science/inria-00510104v1>

Submitted on 10 Nov 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.

Set Manipulations of Fractal Objects Using Matrices of IFS

Joëlle Thollot

LIGIM - Bât 710 - Université Claude Bernard

43, bd du 11 Novembre 1918

69622 VILLEURBANNE Cedex

Tel: 04.72.44.80.00 ext. 42.74 Fax: 04.72.43.13.12

E-mail: jthollot@ligim.univ-lyon1.fr

Laboratoire d'Informatique Graphique Image et Modélisation

ABSTRACT. 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 an extension of the IFS model based on the definition of matrices of IFS that provides a constructive approach of fractal shapes.

KEY WORDS. *Fractals, geometric modeling, IFS, matrices of IFS.*

1 Introduction

The aim of geometric modeling is to model forms by manipulating or combining well-known shapes such as circles and boxes. In order to be convenient such a model should allow easy control and manipulation of the final shape. When dealing 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 of a set of basic shapes, a set of unary operations (shape modifications) and a set of binary operations (shape combinations).

Our work focuses on set operations based on the IFS model. This approach has been inspired by constructive solid geometry (CSG). This technique is classical in geometric modeling. It permits to build complex objects using set operations (union, intersection, ...). In fractal modeling such operations are possible using IFS matrices which are a way to generate a wide class of fractal shapes, not necessarily self-similar. These matrices can be combined using certain operations to yield complex fractal shapes in a constructive way.

2 IFS modeling

Fractals are characterized by their property of self-similarity. BARNSELY's idea [Bar88] consists in using this property in order to encode the different parts of the fractal by contractive operators. An iterative scheme based on these operators is applied. This scheme transforms any initial shape into a union of its reduced copies. 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.

2.1 Definitions

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

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

The space $\mathcal{H}(\mathcal{X})$ denoting the set of non-empty compacts of \mathcal{X} 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} :

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

The associated HUTCHINSON operator is:

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

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

$$\mathcal{A}(\mathcal{T}) = \lim_{k \rightarrow \infty} \mathcal{T}^k \circ K_0,$$

for any non empty compact set K_0 .

IFS attractors constitute a particular family of compacts. They are *invariant sets* for \mathcal{T} :

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

Example 1. We present on Figure 1 an IFS given in [PH91]. By convention,

- $T(a, b, c)$ denotes the translation by the vector (a, b, c) .
- $Rx(a)$ denotes the rotation of angle a around the Ox axis.
- $H(a)$ denotes the scaling of ratio a with respect to the origin of the coordinate system.

$$\mathcal{T}_{tree} = \{T_1, T_2, T_3, T_4\}$$

$$T_1 = H(0.5);$$

$$T_2 = T(0, 0.5, 0) \circ H(0.5);$$

$$T_3 = T(0, 1, 0) \circ Rx(\pi/4) \circ H(0.5);$$

$$T_4 = T(0, 1, 0) \circ Rx(-\pi/4) \circ H(0.5).$$

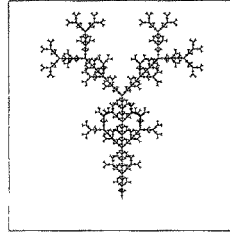


Fig. 1: Example of IFS.

IFS techniques are 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. This paper focuses on affine operators.

However, 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. Such a deficiency is an important limit of fractal geometric modeling.

2.2 Generalizations of IFS techniques

Many authors have proposed generalizations of the IFS model. These generally consist in constructing subsets of an IFS attractor:

$$K_\omega \subseteq \mathcal{A}(\mathcal{T}) \text{ with } K_\omega = \lim_{n \rightarrow \infty} K_n \text{ and } K_n \subseteq \mathcal{T}^n \circ K_0.$$

Two different approaches can be distinguished. Firstly, a “discrete” approach based on quadrees or octrees [BM89] [CD93]: $\mathcal{A}(\mathcal{T}) = [0, 1]^2$ or $[0, 1]^3$. Secondly, a “continuous” approach in which the IFS is arbitrary [Gen92] [PH92] [TT93] [DTG95]. The discrete approach is often used for image compression.

Moreover, in each case there are different techniques to define the restriction of the IFS, such as formal languages, equations systems, trees, graphs, *etc.* These approaches are not totally equivalent. That’s why we have looked for a model that could integrate a wide part of these models. Moreover we needed a model on which operations could easily be defined.

A generalization based on a matricial formalism is proposed in this paper. This approach integrates not only the existing models based on equations systems or graphs but also those based on regular languages (finite automaton, grammars). Moreover, a set of operations is given for a constructive approach of fractal attractors using basic operations such as union and intersection.

3 IFS matrices

Our approach is based on the HUTCHINSON operator, generalized by PEITGEN, JÜRGENS and SAUPE [PJS92].

3.1 Space of matrices of compacts

We define matrices which components are compact sets. As the HAUSDORFF distance is defined for non empty compacts, we should generalize the HAUSDORFF distance in a particular way in order to allow empty components.

We denote $M_R(\mathcal{H}(\mathcal{X}))$ the set of matrices in $\mathcal{H}(\mathcal{X})$ where the non empty elements are given by $R \subseteq \{1, \dots, n\} \times \{1, \dots, n\}$. That is to say for all matrix $\tilde{K} \in M_R(\mathcal{H}(\mathcal{X}))$,

$$\begin{aligned} \tilde{K}_{ij} &\in \mathcal{H}(\mathcal{X}) \cup \emptyset, \\ \tilde{K}_{ij} \neq \emptyset &\Leftrightarrow (i, j) \in R. \end{aligned}$$

We can now define a distance on $M_R(\mathcal{H}(\mathcal{X}))$: let \tilde{K} and \tilde{K}' be in $M_R(\mathcal{H}(\mathcal{X}))$,

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

As R is a finite set and $(\mathcal{H}(\mathcal{X}), d_H)$ a metric space, $(M_R(\mathcal{H}(\mathcal{X})), d_\infty)$ is a complete metric space.

3.2 IFS Matrix

We now define a matrix which components are IFS. Let \mathcal{S} be an affine semigroup. We define a matrix of IFS as an $n \times n$ matrix which components are finite, possibly empty, subsets of \mathcal{S} . We denote $\mathcal{P}_+(\mathcal{S})$ the set of finite subsets of \mathcal{S} and $M_R(\mathcal{P}_+(\mathcal{S}))$ the set of matrices of IFS.

Let $\tilde{H} = (\tilde{H}_{ij}) \in M_R(\mathcal{P}_+(\mathcal{S}))$

$$\tilde{H}_{ij} = \begin{cases} \bigcup_{k \in I_{ij}} T_k & \text{if } (i, j) \in R \\ \emptyset & \text{otherwise} \end{cases},$$

with $T_k \in \mathcal{S}$ and $I_{ij} \subseteq [1 \dots N]$ where $N = \text{Card}(\bigcup_{i,j} \tilde{H}_{ij})$.

Example 2. Let \mathcal{T}_{tree} be the IFS previously defined. We can choose the following matrix of IFS:

$$\tilde{H}_{tree} = \begin{pmatrix} \{T_3, T_4\} & \{T_1, T_2\} \\ \emptyset & \{T_1, T_2\} \end{pmatrix}.$$

$\tilde{H}_{tree} \in M_R(\mathcal{P}_+(\mathcal{S}))$ with $R = \{(1, 1), (1, 2), (2, 2)\}$.

We can now define the composition of these matrices.

Let $\tilde{H} \in M_R(\mathcal{P}_+(\mathcal{S}))$, $\tilde{K} \in M_S(\mathcal{H}(\mathcal{X}))$ and $\tilde{H}' \in M_S(\mathcal{P}_+(\mathcal{S}))$:

$$(\tilde{H} \circ \tilde{K})_{ij} = \bigcup_{k=1..n, (i,k) \in R, (k,j) \in S} \tilde{H}_{ik} \circ \tilde{K}_{kj} \in M_{R \circ S}(\mathcal{H}(\mathcal{X})).$$

$$(\tilde{H} \circ \tilde{H}')_{ij} = \bigcup_{k=1..n, (i,k) \in R, (k,j) \in S} \tilde{H}_{ik} \circ \tilde{H}'_{kj} \in M_{R \circ S}(\mathcal{P}_+(\mathcal{S})).$$

with $R \circ S = \bigcup_{(i,k) \in R, (k,j) \in S} \{(i, j)\}$.

Moreover if the relation R is a transitive one, then $M_R(\mathcal{P}_+(S))$ is stable by composition, that is:

$$\tilde{H} \in M_R(\mathcal{P}_+(S)), \tilde{K} \in M_R(\mathcal{H}(\mathcal{X})) \Rightarrow \tilde{H} \circ \tilde{K} \in M_R(\mathcal{H}(\mathcal{X}))$$

and \tilde{H} is a contractive operator on $(M_R(\mathcal{H}(\mathcal{X})), d_\infty)$.

3.3 Attractor matrix

We now define an attractor matrix as the limit of an iterated matrix of IFS. Such a limit exists when the matrix of IFS is aperiodic. Intuitively, this means that empty components of the matrix will be stable during the iteration. This hypothesis is necessary because of the HAUSDORFF distance.

More precisely, we define $\tilde{H} \in M_R(\mathcal{P}_+(S))$ as an aperiodic matrix when it exists $m \in \mathbb{N}$ such that

$$\forall k > m, R^m = R^k.$$

Let $S = R^m$, then $M_S(\mathcal{P}_+(S))$ is stable by \circ and

$$\forall k > m, \tilde{H}^k \in M_S(\mathcal{P}_+(S)).$$

Moreover, if \tilde{K} is a diagonal matrix, we have

$$\forall k > m, \tilde{H}^k \circ \tilde{K} \in M_S(\mathcal{H}(\mathcal{X})).$$

We can now define an attractor matrix in the same way as the attractor of an IFS. Indeed, let \tilde{H} be an aperiodic matrix, as \tilde{H} is a contractive operator, we can prove that the sequence $(\tilde{H}^k \circ \tilde{K})_{k > m}$ is a CAUCHY sequence in the complete metric space $(M_S(\mathcal{H}(\mathcal{X})), d_\infty)$.

The limit of such a sequence is thus defined, and we call it the attractor matrix:

$$\tilde{\mathcal{A}}(\tilde{H}) = \lim_{n \rightarrow \infty} \tilde{H}^n \circ \tilde{K}.$$

Denoting $\mathcal{T} = \bigcup_{(i,j) \in R} \tilde{H}_{ij}$, we have

$$\tilde{\mathcal{A}}(\tilde{H})_{ij} \subseteq \mathcal{A}(\mathcal{T}).$$

Example 3. In the case of the IFS $\mathcal{T}_{tree} = \{T_1, T_2, T_3, T_4\}$, with the matrix of IFS:

$$\tilde{H}_{tree} = \begin{pmatrix} \{T_3, T_4\} & \{T_1, T_2\} \\ \emptyset & \{T_1, T_2\} \end{pmatrix},$$

we have

$$\tilde{H}_{tree}^k = \begin{pmatrix} \{T_3, T_4\}^k & \bigcup_{i+j=k-1} \{T_3, T_4\}^i \{T_1, T_2\}^j \\ \emptyset & \{T_1, T_2\}^k \end{pmatrix},$$

thus

$$\tilde{\mathcal{A}}(\tilde{H}_{tree}) = \lim_{n \rightarrow \infty} \tilde{H}_{tree}^n \circ \tilde{K} = \begin{pmatrix} \mathcal{A}(T_3, T_4) & \mathcal{A}(T_3, T_4) \cup \{T_3, T_4\}^* \circ \mathcal{A}(T_1, T_2) \\ \emptyset & \mathcal{A}(T_1, T_2) \end{pmatrix}.$$

$\tilde{\mathcal{A}}(\tilde{H}_{tree})$ is shown Figure 2.

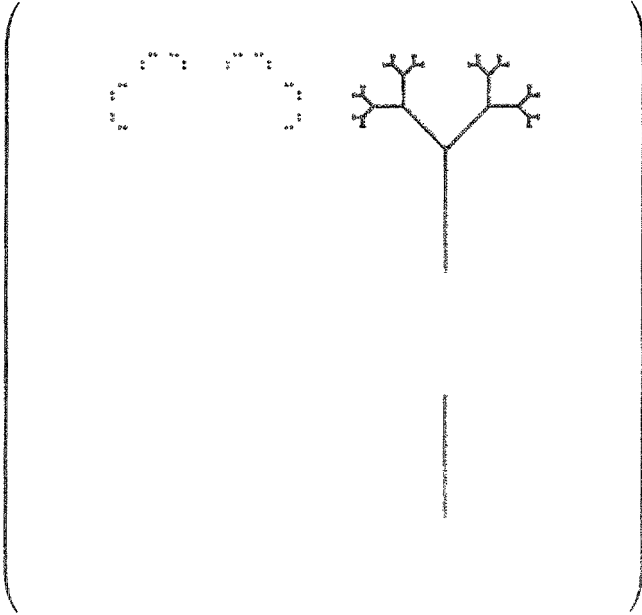


Fig. 2: Attractor matrix $\tilde{A}(\tilde{H}_{tree})$.

We have proved the equivalence of our model with the existing techniques, in particular we have worked on the relation between our model and formal languages ones [Tho96].

In the case of a model based on finite automata, the idea of this relation is to select some of the components of the matrix. The choice is made according to the initial and final states of the automaton. The attractor associated to the automata is then the union of the chosen components.

4 Set manipulations

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 [Tho96]; and some operations associated with formal language theory, such as tensor sum, block matrix [TT95].

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

Let \tilde{H} and \tilde{H}' be two matrices of IFS of size respectively n and m .

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:

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

Tensor product is an external operation:

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

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 “ \diamond_{\cup} ”. We define it by

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

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

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

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

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.

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 “ \diamond ”.

4.3 Constructive approach

We show Figure 3 some attractor matrices made for a project “art and science”. These pictures have been made by Martine Rondet-Mignotte. She is an artist and uses our work to produce artistic images. This figure shows how some complex shapes can be constructed, using the “step by step” approach, with the use of the matrix operations.

Once the resulting matrix has been constructed, one can choose some of the components. This allows to construct an object which is the union of some components. Figure 4 shows an example of objects extracted from the resulting matrix of Figure 3

Figure 5 shows some other results of operations.

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:

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

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 [Tho96], where \downarrow denotes inclusion.

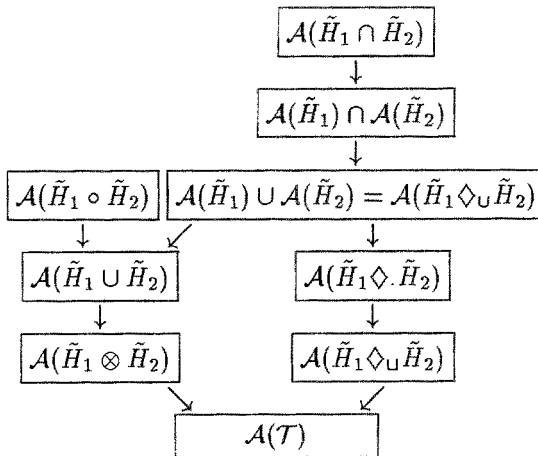


Figure 6 shows an example of inclusion.

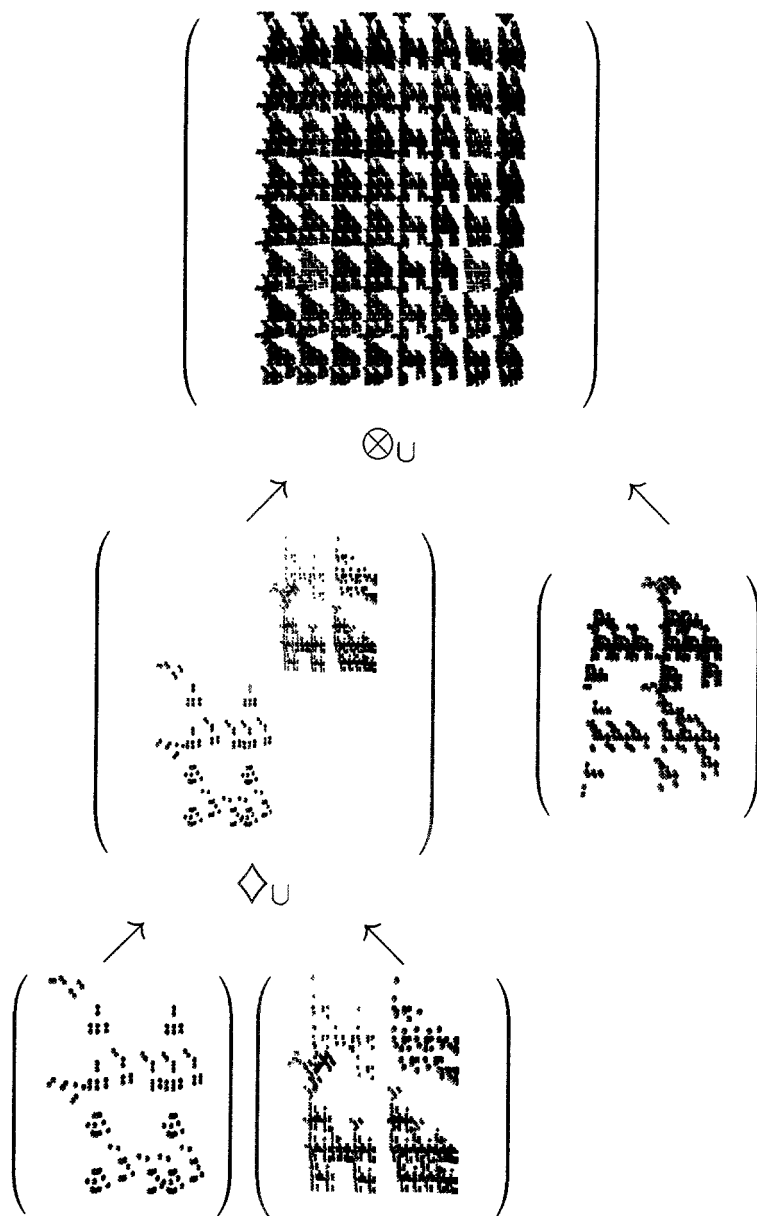


Fig. 3: Constructive tree.

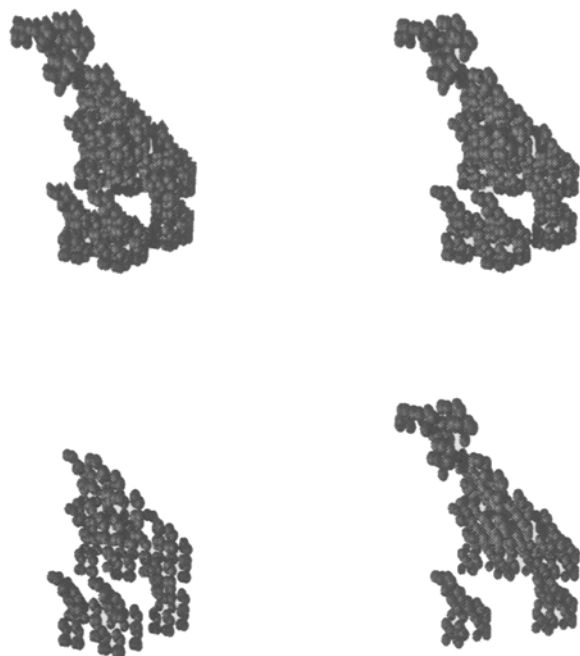


Fig. 4: Four different selections of some components of the precedent attractor matrix

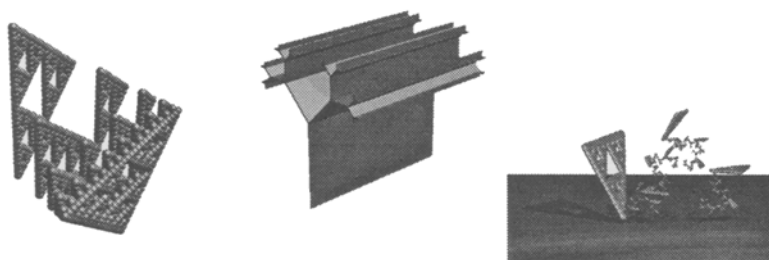


Fig. 5: Examples of 3D attractors

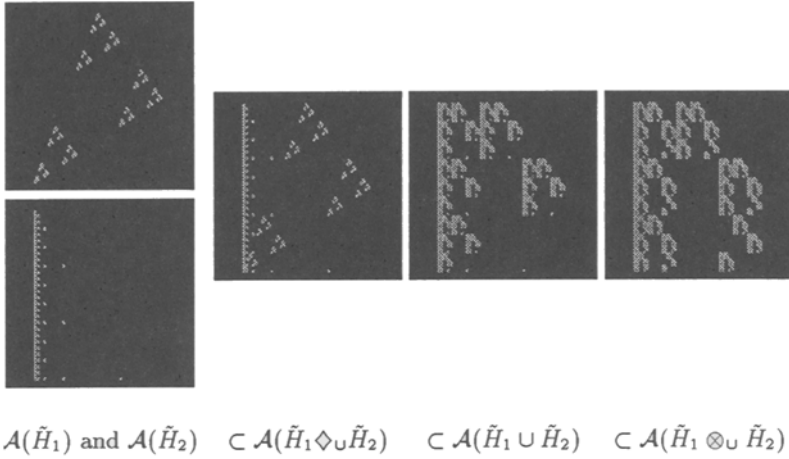


Fig. 6: Example of inclusion.

5 Conclusion

Using a generalization of the classical IFS model, we construct a fractal model that enables a constructive approach.

This approach is a part of a project of fractal modeler. In this project, we have also worked on free forms techniques applied to fractals [ZT96]. We should now combine both approaches to obtain a complete system of fractal modeling.

References

- [Bar88] M.F. Barnsley. *Fractals Everywhere*. Academic press, INC, 1988.
- [BM89] J. Berstel and M. Morcrette. Compact representation of patterns by finite automata. In *Proceedings of Pixim*, pages 387–395, 1989.
- [CD93] K. Culik II and S. Dube. Rational and affine expressions for image synthesis. *Discrete Appl. Math.*, 41:85–120, 1993.
- [DTG95] S. Duval, M. Tagine, and D. Ghazanfarpour. Modélisation de fractals par les arbres étiquetés. In *3emes journées AFIG, Marseille*, pages 125–134, novembre 1995.
- [Gen92] C. Gentil. *Les Fractales en Synthèse d’Images : le Modèle IFS*. PhD thesis, Université Claude Bernard LYON 1, France, 1992.
- [PH91] P. Prusinkiewicz and M.S. Hammel. Automata, languages and iterated function systems. In *lecture notes for the SIGGRAPH’91 course : “Fractal modeling in 3D computer graphics and imagery”*, 1991.
- [PH92] P. Prusinkiewicz and M.S. Hammel. Escape-time visualization method for language-restricted iterated function systems. In *Proceedings of Graphics Interface’92*, May 1992.
- [PJS92] H.O. Peitgen, H. Jürgens, and D. Saupe. Encoding images by simple transformations. In *Fractals For The Classroom*, New York, 1992.

- [Tho96] J. Thollot. *Extension du Modèle IFS pour une Géométrie Fractale Constructive*. PhD thesis, Université Claude Bernard LYON 1, France, sept 1996.
- [TT93] J. Thollot and E. Tosan. Construction of fractales using formal languages and matrices of attractors. In Harold P. Santos, editor, *Proceedings of Compugraphics'93*, pages 74–81, Technical University of Lisbon, december 1993.
- [TT95] J. Thollot and E. Tosan. Constructive fractal geometry : constructive approach to fractal modeling using languages operations. In *Proceedings of Graphics Interface'95, Quebec, Canada*, pages 196–203, may 1995.
- [ZT96] C. Zair and E. Tosan. Fractal modeling using free forms techniques. In *Proceedings of Eurographics*, 1996.