Fractals, Vol. 9, No. 1 (2001) 95–103

A FRACTAL APPROXIMATION OF CURVES

ERIC GUÉRIN, ERIC TOSAN and ATILLA BASKURT
LIGIM EA1899 - Laboratoire d'Informatique Graphique, Image et Modélisation,
Université Claude Bernard Lyon 1,
43, bd du 11 novembre 1918, 69622 Villeurbanne Cedex

Received June 28, 2000; Accepted November 13, 2000

Abstract

This paper deals with the approximation of rough curves using a fractal model. The approximation criterion is based on a curve parametrization and the fractal model is a projected iterated function system (IFS) model. This model unifies the IFS model and a classical model used in computer graphics (free form representation with control points). To formulate the approximation problem, we introduce a family of curves based on our model. Then, the approximation problem has a nonlinear fitting formulation. We have tested this fractal approximation method on both smooth and rough synthetic curves. A shape extracted from a natural scene has also been approximated. The obtained results are very satisfying from both quantitative and visual point of view.

1. INTRODUCTION

The approximation of natural complex shapes constitutes an important research area for reconstruction and representation problems in several application domains, such as medical imaging, multimedia data representation and CAGD (computed assisted geometric design).

In fractal theory, the determination of a model for approximating natural data, is called "the inverse problem." The model has to be simple, compact and easy to manipulate in terms of computation and visualization. The paper addresses the problem of approximation of natural rough curves with such a model. Few works have been reported on this subject. Most of them use the fractal interpolation function (FIF) model introduced by Barnsley.[2] This model is an iteration function system (IFS) composed of shearings. Its stability has been studied by Feng and Xie.[3] Among the studies proposed to solve the inverse problem for this FIF model, Berkner[4] uses a method based on the wavelet transform, in order to obtain the exact reconstruction for curves. Cochran *et al.*[5] propose another study with a more general model and a method based on the fractal image compression.[6] The model obtained is composed of a great number of transformations and depends on the way of partitioning the original curve.

In this paper, we propose a new approximation method based on a model that generalizes IFS model and free form curves used in computer graphics. This model has the convenience to be simple to code and to represent a large variety of curves. The resolution is based on the minimization of a distance with a nonlinear fitting algorithm.

2. CURVE APPROXIMATION PROBLEM

The quality of the approximation between two comparable objects is given by an evaluation criterion. Traditionally, the criterion used is a given distance. To compute this distance, one needs a unified representation of the objects to compare. At first, let us present the three representations for curves used in this study:

(i) Parametric curve: A continuous function taking values in $\mathbb{R}^2$

$$t \in [0,1] \mapsto F(t) \in \mathbb{R}^2 .$$

(ii) Set curve: A compact set of points in $\mathbb{R}^2$

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

(iii) Sampled curve: A list of points in $\mathbb{R}^2$

$$(Q_i)_{i=0,\ldots,m} .$$

These points can be generated from a parametric curve:

$$Q_i = F(t_i) .$$

Several distances can be handled as functions of these representations. In fractal modeling, the Hausdorff distance is often used. It is a distance between sets. Let A and A' be two compact sets of a metric space $(\mathcal{X}, d)$, the Hausdorff distance between A and A' is given by:

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

This distance, used for the IFS theory,[2] is indeed rarely considered due to its computational complexity. In the case of sampled (discrete) curves, the simplest way to compute a distance between $\mathbf{Q} = (Q_i)_{i=0,\ldots,m}$ and $\mathbf{Q}' = (Q'_i)_{i=0,\ldots,m}$ is:

$$D(\mathbf{Q}, \mathbf{Q}') = \sum_{i=0}^{m} d(Q_i, Q'_i) .$$

This distance presents also some drawbacks. Its computation is possible only if the two curves are defined using the same sampling function leading to the same number of points. In order to generalize the measure of similarity between curves without any of the constraints described above, we propose a new distance based on the parametrization of the curves to compare.

To obtain a uniform parametrization, we use the arc length $s_i(\mathbf{Q})$ of the ith sample of $\mathbf{Q}$:

$$\text{for } i = 1,\ldots,m : s_i(\mathbf{Q}) = \frac{1}{l(\mathbf{Q})} \sum_{j=0}^{i-1} d(Q_j, Q_{j+1})$$

and

$$s_0(\mathbf{Q}) = 0 .$$

$l(\mathbf{Q})$ is the length of the sampled curve $\mathbf{Q}$ defined as:

$$l(\mathbf{Q}) = \sum_{j=0}^{m-1} d(Q_j, Q_{j+1}) .$$

Consequently, we can introduce a parametrization function $\mathcal{G}_{\mathbf{Q}}$ associated with the sampled curve $\mathbf{Q}$. $\mathcal{G}_{\mathbf{Q}}$ is a piecewise affine function. For all $i = 0,\ldots,m-1$ and $\forall\, t \in [s_i(\mathbf{Q}), s_{i+1}(\mathbf{Q})]$:

$$\mathcal{G}_{\mathbf{Q}}(t) = \left[\frac{t - s_i(\mathbf{Q})}{s_{i+1}(\mathbf{Q}) - s_i(\mathbf{Q})}\right] Q_{i+1} + \left[1 - \frac{t - s_i(\mathbf{Q})}{s_{i+1}(\mathbf{Q}) - s_i(\mathbf{Q})}\right] Q_i . \quad (1)$$

We can now measure a distance χ^2 between functions $\mathcal{G}_{\mathbf{Q}}$ and $\mathcal{G}_{\mathbf{Q}'}$ rather than a distance between sampled curves $\mathbf{Q}$ and $\mathbf{Q}'$. Note that this approach allows us to compare $\mathbf{Q}$ and $\mathbf{Q}'$ which can be obtained by different sampling functions. In Eq. (2), χ^2 is computed following a uniform sampling of the two functions:

$$\chi^2 = \mathcal{D}(\mathcal{G}_{\mathbf{Q}}, \mathcal{G}_{\mathbf{Q}'}) = \sum_{i=0}^{M} \left[d\left(\mathcal{G}_{\mathbf{Q}}\left(\frac{i}{M}\right), \mathcal{G}_{\mathbf{Q}'}\left(\frac{i}{M}\right)\right)\right]^2 . \quad (2)$$

This new criterion will be used to approximate natural rough curves using a fractal model. This model is based on free form curves and IFS: fractal free form curves that will be detailed in the following section.

3. FRACTAL MODEL

First, we summarize some important definitions about IFS. Then, the fractal free form curve model is presented. Finally, the three representations associated to the model are described.

3.1 Iterated Function System

Introduced by Barnsley[2] in 1988, the IFS model generates a geometrical shape with an iterative process. An IFS-based modeling system is defined by a triple $(\mathcal{X}, d, \mathcal{S})$ where:

(i) $(\mathcal{X}, d)$ is a complete metric space, $\mathcal{X}$ is called *iteration space*; and
(ii) $\mathcal{S}$ is a semigroup acting on the points of $\mathcal{X}$ such that: $p \to Tp$ where T is a contractive operator and $\mathcal{S}$ is called *iteration semigroup*.

An *IFS* $\mathcal{I}$ (*iterative function system*) is a finite subset of $\mathcal{S}$: $\mathcal{I} = \{T_0, \ldots, T_{N-1}\}$ with operators $T_i \in \mathcal{S}$. We note $\mathcal{H}(\mathcal{X})$, the set of non-empty compacts of $\mathcal{X}$. The associated Hutchinson operator is:

$$K \in \mathcal{H}(\mathcal{X}) \mapsto \mathcal{I}K = T_0K \cup \cdots \cup T_{N-1}K\,.$$

This operator is contractive in the new complete metric space $(\mathcal{H}(\mathcal{X}), d_H)$ and admits a fixed point, called *attractor*, defined by[2]:

$$\mathcal{A}(\mathcal{I}) = \lim_{n\to\infty} \mathcal{I}^n K \quad \text{with} \quad K \in \mathcal{H}(\mathcal{X})\,.$$

Each IFS $\mathcal{I}$ defines a unique compact $\mathcal{A}(\mathcal{I})$ called *IFS attractor*. $\mathcal{A}(\mathcal{I})$ is a self-similar compact set:

$$\mathcal{A}(\mathcal{I}) = \mathcal{I}\mathcal{A}(\mathcal{I})\,.$$

3.2 Projected IFS Attractors

In fractal compression or fractal interpolation,[2,5] $\mathcal{X} = \mathbb{R}^2$ and $\mathcal{S}$ is the semigroup of contractive affine operators. In our approach, we take another space, the barycentric space $\mathcal{X} = \mathcal{B}^n$ defined by:

$$\mathcal{B}^n = \left\{ (\lambda_j)_{j=0,\ldots,n} \,\middle|\, \sum_{j=0}^{n} \lambda_j = 1 \right\}.$$

The iteration semigroup is constituted by matrices with barycentric columns:

$$\mathcal{S} = \left\{ T \,\middle|\, \sum_{i=0}^{n} T_{ij} = 1 \qquad \forall\, j = 0, \ldots, n \right\}.$$

This choice leads to the generalization of IFS attractors named **projected IFS attractors**:

$$P\mathcal{A}(\mathcal{I}) = \{P\lambda | \lambda \in \mathcal{A}(\mathcal{I})\} \tag{3}$$

where P is a sequence of control points:

$$P = (p_0, \ldots, p_n)$$

and $P\lambda = \sum_{i=0}^{n} \lambda_i p_i$. The interest of this model is to unify the IFS model and the classical free form representation used in computer aided geometric design.[7,8]

3.3 Representations Associated with the Model

In Sec. 2, we defined three representations for curves. In this section, we show that all these representations can be generated from our fractal model. In order to simplify the expressions, we consider the model based on two transformations T_0 and T_1, and a set of four control points P.

3.3.1 *Parametric representation*

With good hypotheses on IFS,[8] the attractor is a curve: $\mathcal{A}(\mathcal{I}) = \{\phi(t) | t \in [0,1]\}$ where ϕ is a continuous function: $[0,1] \to \mathcal{X}$.

To each real value t, a binary development $\alpha(t)$ can be associated:

$$t = \sum_{i=1}^{\infty} \frac{1}{2^i} \alpha_i(t)\,.$$

Then, the value of $\phi(t)$ is computed iteratively[9]:

$$\phi(t) = \lim_{k\to\infty} T_{\alpha_1(t)} \cdots T_{\alpha_k(t)} \lambda, \qquad \forall\, \lambda \in \mathcal{X}\,.$$

In this way, we can construct a parametric fractal curve characterized by a set of control points P[7,8] using the projection:

$$F(t) = P\phi(t) = \sum_{i=0}^{n} \phi_i(t) p_i$$

where $\phi(t)$ is a vector of functions:

$$\phi(t) = (\phi_0(t), \ldots, \phi_n(t))^T\,.$$

3.3.2 *Set representation*

This representation corresponds to the projected IFS attractor of Eq. (3):

$$P\mathcal{A}(\mathcal{I}) = P \lim_{k\to\infty} \{T_0, T_1\}^k K\,.$$

For all compact K, the result will be the same.

3.3.3 *Sampled representation*

With a tabulation process, considering only the values of t which are integral multiples of $\frac{1}{2^p}$ leads to a simplification in the computing. The sampled curve $(Q_i)_{i=0,\ldots,2^p}$ is then defined by:

$$\begin{aligned} Q_i &= F\left(\frac{i}{2^p}\right) \\ &= P \lim_{k\to\infty} T_{\alpha_1(\frac{i}{2^p})} \cdots T_{\alpha_k(\frac{i}{2^p})}\lambda, \quad \forall\, \lambda \in \mathcal{X}\,. \end{aligned}$$

We know the binary development of $\frac{i}{2^p}$:

for $i = 0, \ldots, 2^p - 1$:

$$\alpha\left(\frac{i}{2^p}\right) = \alpha_1\left(\frac{i}{2^p}\right) \cdots \alpha_p\left(\frac{i}{2^p}\right) 0\ 0 \cdots$$

for $i = 2^p$:

$$\alpha\left(\frac{i}{2^p}\right) = \alpha(1) = 1\ 1\ 1 \cdots\,.$$

We use the following invariance properties of T_0 and T_1:

$$T_0\ e_0 = e_0 \qquad T_1\ e_3 = e_3$$

where $e_0 = (1,0,0,0)^T$ and $e_3 = (0,0,0,1)^T$. For all $\lambda \in \mathcal{X}$, applying the fixed point theorem, we have:

$$\lim_{k\to\infty} {T_0}^k \lambda = e_0 \qquad \lim_{k\to\infty} {T_1}^k \lambda = e_3\,.$$

Finally, the sampled curve can be generated computing p iterations without any loss of information:

for $i = 0, \ldots, 2^p - 1$:

$$\begin{aligned} Q_i &= P \lim_{k\to\infty} T_{\alpha_1(\frac{i}{2^p})} \cdots T_{\alpha_p(\frac{i}{2^p})} {T_0}^k \lambda \\ &= P T_{\alpha_1(\frac{i}{2^p})} \cdots T_{\alpha_p(\frac{i}{2^p})} e_0 \end{aligned} \qquad (4)$$

for $i = 2^p$:

$$Q_i = P \lim_{k\to\infty} {T_1}^k \lambda = P\ e_3\,.$$

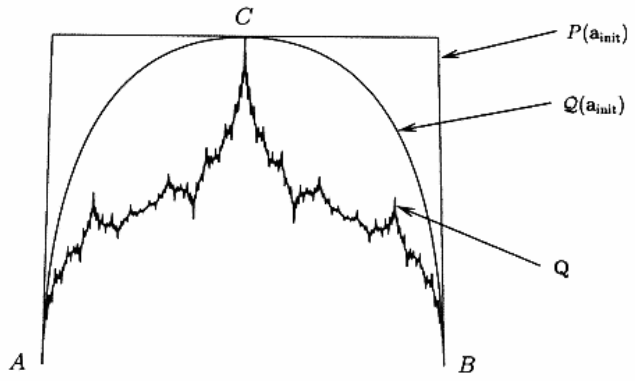

Fig. 1 Extraction of $\mathbf{a}_{\text{init}}$.

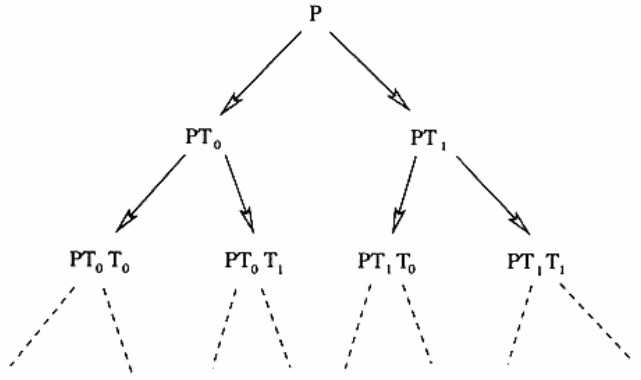

Fig. 2 Construction tree in the two-transformation case.

3.4 Construction

The construction process is directly drawn from the projection Eq. (3):

$$P\mathcal{A}(\mathcal{I}) = P \lim_{k\to\infty} K_k = \lim_{k\to\infty} (P\ K_k) \qquad (5)$$

with $K_k = \mathcal{I}^k\ K$.

Practically, we choose a polygon for the compact K (Fig. 3, step 0) in order to have an easy projection. The following notation permits to introduce an addressing (indexing) of the successive substitutions of the curve: $\alpha = \alpha_1 \cdots \alpha_n$ is a binary word and $|\alpha| = n$ is the length of this word, i.e. $\alpha = 1\,001\,010$ with $|\alpha| = 7$. The number of points is doubled at each iteration.

To visually explain the construction of a curve, let us consider the following example where a given curve (Fig. 3) has been modeled by four control points and two transformations T_0 et T_1:

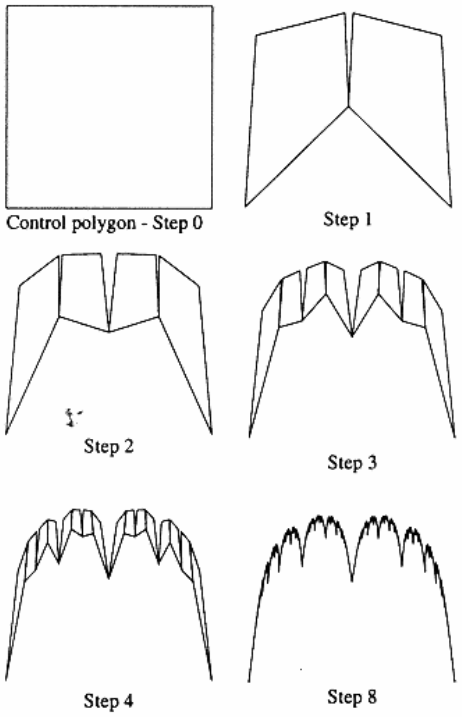

Fig. 3 Example of the construction of a fractal curve.

(i) IFS model defined by 2 contractive operators:

$$T_0 = \begin{pmatrix} 1 & 0.15 & 0.02 & 0.25 \\ 0 & 0.80 & 0.50 & 0.25 \\ 0 & 0.05 & 0.46 & 0.25 \\ 0 & 0.00 & 0.02 & 0.25 \end{pmatrix}$$

$$T_1 = \begin{pmatrix} 0.25 & 0.02 & 0.00 & 0 \\ 0.25 & 0.46 & 0.05 & 0 \\ 0.25 & 0.50 & 0.80 & 0 \\ 0.25 & 0.02 & 0.15 & 1 \end{pmatrix}.$$

(ii) Set of control points: $P = ((-1,0)^T, (-1,2)^T, (1,2)^T, (1,0)^T)$.

The compact K used for visualization is a polygon. The vertices of polygon K are: e_0, e_1, e_2, e_3 with $e_0 = (1,0,0,0)^T$, $e_1 = (0,1,0,0)^T$, $e_2 = (0,0,1,0)^T$ and $e_3 = (0,0,0,1)^T$.

The expression $P\ K_k$ in the projection Eq. (5) can be developed as:

$$\begin{aligned} P\ K_k &= P\mathcal{I}^k\ K \\ &= P\{T_0, T_1\}^k K \\ &= P \bigcup_{|\alpha|=k} T_{\alpha_1} T_{\alpha_2} \cdots T_{\alpha_k} K\,. \end{aligned}$$

This contruction is clearly similar to a binary tree (Fig. 2). Figure 3 represents the evolution of $P\ K_k$, beginning with the control polygon and converging to the fractal curve in a small number of iterations.

4. FRACTAL APPROXIMATION

This section details the construction of a family of curves based on our fractal model. Then, the approximation with the Levenberg-Marquardt nonlinear fitting method is presented. Finally, the implementation process is developed.

4.1 Parametrization of a Family of Curves

We consider a family of curves for which the parameters are the coefficients of matrices T_0 and T_1 and components of control points P. Let us consider a fractal curve modeled with two transformation matrices and four control points. This leads to the following familly of triplets $(T_0(\mathbf{a}), T_1(\mathbf{a}), P(\mathbf{a}))$ parametrized by a vector $\mathbf{a} = (a_1, \ldots, a_{23})$:

$$T_0(\mathbf{a}) = \begin{pmatrix} 1 & a_1 & a_4 & a_7 \\ 0 & a_2 & a_5 & a_8 \\ 0 & a_3 & a_6 & a_9 \\ 0 & 1-(a_1+a_2+a_3) & 1-(a_4+a_5+a_6) & 1-(a_7+a_8+a_9) \end{pmatrix}$$

$$T_1(\mathbf{a}) = \begin{pmatrix} a_7 & a_{10} & a_{13} & 0 \\ a_8 & a_{11} & a_{14} & 0 \\ a_9 & a_{12} & a_{15} & 0 \\ 1-(a_7+a_8+a_9) & 1-(a_{10}+a_{11}+a_{12}) & 1-(a_{13}+a_{14}+a_{15}) & 1 \end{pmatrix} \quad (6)$$

$$P(\mathbf{a}) = \begin{pmatrix} a_{16} & a_{18} & a_{20} & a_{22} \\ a_{17} & a_{19} & a_{21} & a_{23} \end{pmatrix}.$$

4.2 Approximation Principle

The family of triplets in (6) allows us to construct a sampled curve. Given a construction depth p, let us call $\mathcal{Q}(\mathbf{a}) = \mathcal{Q}(T_0(\mathbf{a}), T_1(\mathbf{a}), P(\mathbf{a}))$ the sampled curve composed of $2^p + 1$ points.

In order to measure the similarity between the original sampled curve $\mathbf{Q}$ and the sampled curve $\mathcal{Q}(\mathbf{a})$ generated with the parameter vector $\mathbf{a}$, we use the corresponding parametrizations $\mathcal{G}_{\mathbf{Q}}$ and $\mathcal{G}_{\mathcal{Q}(\mathbf{a})}$, the functions defined in Eq. (1). Consequently, the approximation problem consists in searching the optimal parameter vector $\mathbf{a}_{\text{opt}}$ that minimizes the following function $\chi^2(\mathbf{a})$:

$$\begin{aligned}\mathbf{a}_{\text{opt}} &= \arg\min_{\mathbf{a}} \chi^2(\mathbf{a}) \\ &= \arg\min_{\mathbf{a}} \mathcal{D}(\mathcal{G}_{\mathbf{Q}}, \mathcal{G}_{\mathcal{Q}(\mathbf{a})}) \\ &= \arg\min_{\mathbf{a}} \sum_{i=0}^{M} \left[d\left(\mathcal{G}_{\mathbf{Q}}\left(\frac{i}{M}\right), \mathcal{G}_{\mathcal{Q}(\mathbf{a})}\left(\frac{i}{M}\right)\right)\right]^2\end{aligned}$$

with M arbitrary.

Now, let us denote:

$$f(\mathbf{a}, t) = d(\mathcal{G}_{\mathbf{Q}}(t), \mathcal{G}_{\mathcal{Q}(\mathbf{a})}(t)) .$$

This formulation is close to a nonlinear fitting problem which can be resolved using the Levenberg-Marquardt algorithm.

4.3 Levenberg-Marquardt

Levenberg-Marquardt[10] algorithm is a numerical resolution of the following fitting problem:

$$\mathbf{a}_{\text{opt}} = \arg\min_{\mathbf{a}} \sum_{i=0}^{M} (z_i - f(\mathbf{a}, t_i))^2$$

where vectors $\mathbf{z}$ and $\mathbf{t}$ are the fitting data and f is the fitting model.

In order to resolve our approximation problem using this algorithm, we have to consider the following fitting data:

$$\mathbf{t} = (t_0, \ldots, t_M) = \left(0, \frac{1}{M}, \frac{2}{M}, \ldots, \frac{M-1}{M}, 1\right) :$$

the uniform arc length points between $[0, 1]$

$$\mathbf{z} = (z_0, \ldots, z_M) = (0, \ldots, 0) .$$

$f(\mathbf{a}, t)$ is a distance between $\mathcal{G}_{\mathbf{Q}}$ and $\mathcal{G}_{\mathcal{Q}(\mathbf{a})}$ for a given value of t. This distance should be close to zero in order to fit these two curves perfectly. That is the reason why we choose a null vector for $\mathbf{z}$.

A necessary condition for applying the Levenberg-Marquardt algorithm is that the function $\mathbf{a} \mapsto f(\mathbf{a}, t)$ belongs to the class C^1. We prove this by using another result from Vrscay and Saupe.[11]

Proposition. *The function* $\mathbf{a} \mapsto d(\mathcal{G}_{\mathbf{Q}}(t), \mathcal{G}_{\mathcal{Q}(\mathbf{a})}(t)) \forall\ t \in [0, 1]$ *belongs to the class* C^1.

Proof. Let us decompose this function into elementary functions:

$$\begin{aligned}\mathbf{a} &\xmapsto{(1)} (T_0, T_1, P) \xmapsto{(2)} P\phi \xmapsto{(3)} \mathcal{Q}(\mathbf{a}) \xmapsto{(4)} \mathcal{G}_{\mathcal{Q}(\mathbf{a})} \\ &\xmapsto{(5)} d(\mathcal{G}_{\mathbf{Q}}(t), \mathcal{G}_{\mathcal{Q}(\mathbf{a})}(t)) .\end{aligned}$$

Each of these functions belongs to the class C^1:

Function (1): This is a linear function that only makes a mapping between a parameter vector $\mathbf{a}$ and the set (T_0, T_1, P).

Function (2): Vrscay and Saupe prove that the function $(T_0, T_1) \to \Phi$ is C^1. The projection operator P is also a linear operator.

Function (3): This is a uniform sampling function belonging to the class C^1.

Function (4): This is a linear interpolation function that consists in picking a set of uniformly spaced points on $\mathcal{Q}(\mathbf{a})$. It belongs to the class C^1.

Function (5): This is a function associated to a distance and belonging to the class C^1.

Consequently, the composition of these five functions belongs to the class C^1. □

In order to apply this algorithm, one needs:

(i) An initial estimation $\mathbf{a}_{\text{init}}$ for the parameter vector $\mathbf{a}$.

(ii) The fitting model that gives the values of $f(\mathbf{a}, t)$ and its partial derivatives $\frac{\partial f}{\partial a_i}(\mathbf{a}, t)$ for a given parameter vector $\mathbf{a}$ and a value of t.

First, we show how the initial parameter vector $\mathbf{a}_{\text{init}}$ is estimated. Then, details on the computation of the partial derivatives are presented.

4.4 Initialization of the Method

With the Levenberg-Marquardt method,[10] the quality of the initial guess of the vector $\mathbf{a}$ may have a

great impact on the final result. We estimate the initial parameter vector $\mathbf{a}_{\text{init}}$ by using geometrical properties of the original curve $\mathbf{Q}$.[12] We build a B-spline curve tangent to the curve $\mathbf{Q}$ at its extremities A and B and fitting $\mathbf{Q}$ to the maximum value C. Figure 1 shows an example of a curve $\mathbf{Q}$ with the associated B-spline $\mathcal{Q}(\mathbf{a}_{\text{init}})$ and the control polygon $P(\mathbf{a}_{\text{init}})$. In this case, T_0 and T_1 are reduced to B-spline transformations:

$$T_0(\mathbf{a}_{\text{init}}) = \begin{pmatrix} 1 & \frac{1}{2} & 0 & 0 \\ 0 & \frac{1}{2} & \frac{3}{4} & \frac{1}{2} \\ 0 & 0 & \frac{1}{4} & \frac{1}{2} \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

and

$$T_1(\mathbf{a}_{\text{init}}) = \begin{pmatrix} 0 & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{4} & 0 & 0 \\ \frac{1}{2} & \frac{3}{4} & \frac{1}{2} & 0 \\ 0 & 0 & \frac{1}{2} & 1 \end{pmatrix}.$$

4.5 Computing of Partial Derivatives

To compute the partial derivatives, we use a small vector $\delta\mathbf{a}_i$ that is used to disturb the parameter vector $\mathbf{a}$:

$$\delta\mathbf{a}_i = (\underbrace{0, \ldots, 0, \varepsilon}_{i}, 0, \ldots, 0).$$

Then, the computation of partial derivatives is approximated by:

$$\frac{\partial f}{\partial a_i}(\mathbf{a}, t) \simeq \frac{f(\mathbf{a} + \delta\mathbf{a}_i, t) - f(\mathbf{a}, t)}{\varepsilon}.$$

5. RESULTS

Two categories of results are proposed. First, results on synthetic curves are shown. Then, results on a curve extracted from a natural picture are presented.

5.1 Synthetic Curves

Table 1 presents the results concerning the approximation process for the four curves given in Figs. 4 and 5. The initial curve $\mathbf{Q}$ and the reconstructed curve $\mathcal{Q}(\mathbf{a}_{\text{opt}})$ are represented with plain line and small dots, respectively. The B-spline model $\mathcal{Q}(\mathbf{a}_{\text{init}})$ used to initialize the fitting process is also represented. The final distance χ^2 is close to zero, except for Fig. 5(b) where we have a complex fractal curve. The visual quality of the approximations obtained by our fractal model is quite good. We also note that logically the algorithm takes more iterations to converge for complex shapes [Figs. 4(b) and 5(b)].

Table 1 Numeric results on synthetic curves.

Figure	Final χ^2	No. of Iterations	Computing Time (s) (Pentium III 450, 64 Mo RAM)
4(a)	0.0025	67	18
4(b)	0.0116	268	59
5(a)	0.0006	66	14
5(b)	0.8468	225	50

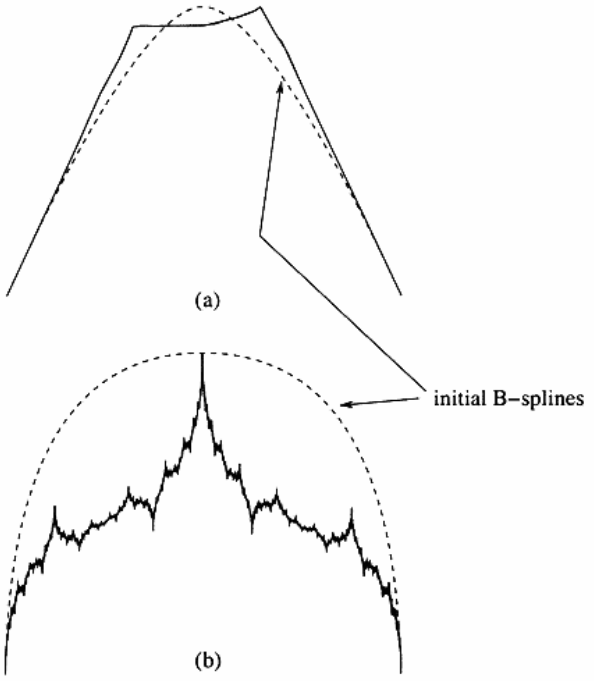

Fig. 4 Visual results on synthetic curves (1).

5.2 "Natural" Curve

The shape of a mountain has been manually extracted from a natural scene. This shape is

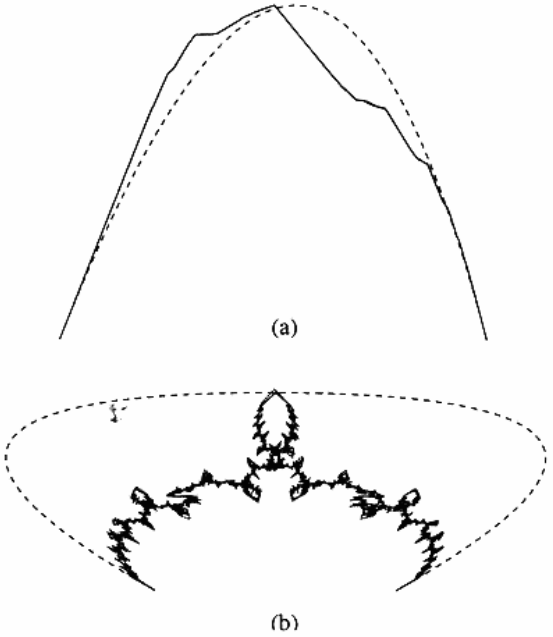

Fig. 5 Visual results on synthetic curves (2).

approximated with the fractal model (Fig. 6). Numerically, 67 iterations and 18 seconds have been necessary to converge; final value of χ^2 is 0.0145. This result is validated by the visual aspect of the approximation. Note that only two elementary transformations and four control points were required to approximate this natural shape initially defined by 210 points.

6. CONCLUSION

In order to handle rough natural shapes, we have proposed an approximation method based on a projected IFS model. We have built a fractal curve family based on this model. The approximation criterion based on a parametrization of sampled curves has been defined. Consequently, the approximation problem has been solved using a nonlinear fitting method. We have tested the whole fractal approximation method on both smooth and rough synthetic curves. A shape extracted from a natural scene has been also modeled. The results obtained for the evaluation criteria (both numerical (χ^2 distance) and visual) are very satisfying. Furthermore, the method can be easily applied to more general models, with more transformations and more control points.

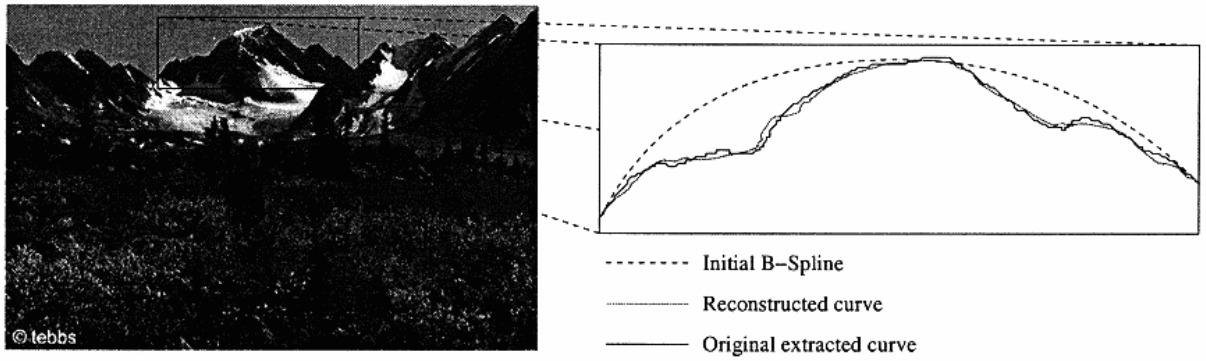

Fig. 6 Approximation of a natural curve with the fractal model.

REFERENCES

1. E. Guérin, E. Tosan and A. Baskurt, "Fractal Coding of Shapes Based on a Projected Ifs Model," *Proc. ICIP2000*, IEEE Signal Processing Society **2**, 203–206 (2000).
2. M. Barnsley, *Fractals Everywhere* (Academic Press, London, 1988).
3. Z. Feng and H. Xie, "On Stability of Fractal Interpolation," *Fractals* **6**(3), 269–273 (1998).
4. K. Berkner, "A Wavelet-Based Solution to the Inverse Problem for Fractal Interpolation Functions," in *Fractals in Engineering '97*, eds. J. Lévy Véhel, E. Lutton and C. Tricot (Springer, London, 1997), pp. 81–92.
5. W. O. Cochran, J. C. Hart and P. J. Flynn, "On Approximation Rough Curves with Fractal Functions," in *Graphics Interface '98* (June 1998), pp. 65–72.
6. A. E. Jacquin, "Image Coding Based on a Fractal Theory of Iterated Contractive Image

Transformations," *IEEE Trans. Image Processing Vol. 1* (January 1992), pp. 18–30.

7. C. E. Zair and E. Tosan, "Fractal Modeling Using Free Form Techniques," *Computer Graphics Forum* **15**(3), 269–278 (August 1996). EUROGRAPHICS '96 (Conference issue).
8. C. E. Zair and E. Tosan, "Computer Aided Geometric Design with IFS Techniques," in *Fractals Frontiers*, eds. M. M. Novak and T. G. Dewey (World Scientific Publishing, Denver, April 1997), pp. 443–452.
9. H. Prautzsch and C. A. Micchelli, "Computing Curves Invariant Under Halving," *Computer Aided Geometric Design* **4**, 133–140 (1987).
10. W. H. Press, B. P. Flannery, S. A. Teukolsky and W. T. Vetterling, *Numerical Recipes in C: The Art of Scientific Computing*, 2nd ed. (Cambridge University Press, Cambridge, 1993).
11. E. R. Vrscay and D. Saupe, "Can One Break the Collage Barrier in Fractal Image Coding," in *Fractals: Theory and Applications in Engineering*, eds. M. Dekking, J. Lévy Véhel, E. Lutton and C. Tricot (Springer, London, 1999), pp. 307–323.
12. E. Guérin, E. Tosan and A. Baskurt, "Description et reconstruction de courbes par une approche fractale," *AFIG'99*, 146–155 (November 1999).