MODELING AND APPROXIMATION OF FRACTAL SURFACES WITH PROJECTED IFS ATTRACTORS

E. GUÉRIN, E. TOSAN AND A. BASKURT
LIGIM - EA 1899 - Computer Graphics, Image and Modeling Laboratory
Claude Bernard University, Lyon, France
Bât. 710 - 43, bd du 11 novembre 1918 - 69622 Villeurbanne Cedex
Tel.: (33) 4.72.43.26.10 - Fax : (33) 4.72.43.13.12
E-mail: [eguerin|et|abaskurt]@ligim.univ-lyon1.fr

A method for modeling and approximating rough surfaces is introduced. A fractal model based on projected IFS attractors allows the definition of free form fractal shapes controled with a set of points. This flexible model has good fitting properties for recovering surfaces. The approximation is formulated as a non-linear fitting problem and resolved using a modified LEVENBERG-MARQUARDT minimisation method. The main applications are surface modeling, shape description and geometric surface compression.

1 Introduction

Basically, the problem of approximating the surface of 3D objects consists in finding a model that represents a set of data points:

$$(x_i, y_i, z_i) \in \mathbb{R}^3, \ \forall i = 0, \ldots, n$$

A wide variety of representation methods have been proposed for modeling these surfaces[1]. Unfortunately, these models do not recover rough surfaces, i.e. surfaces defined by continuous functions that are nowhere differentiable. Models that are able to produce rough surfaces are mostly based on random processes. This is the reason why these models are not suitable for approximation. In order to propose an efficient solution to the problem of rough surface approximation, the current study proposes a parametric model based on a deterministic fractal approach.

In [2] and [3], we have proposed a model for fractal curves and surfaces. This model combines two classical models: a fractal model (IFS attractors) and a CAGD model (free form shapes). This model is called projected IFS attractors. A set of control points allows an easy and flexible control of the fractal shape generated by the IFS model and provide a high quality fitting, even for surfaces with sharp transitions. In [4] and [5], we have proposed an approximation method for curves based on this model. In [6], we give the extension of this method to surfaces. In this paper, we present a general formulation of surface modeling and some approximation results.

2 Model

2.1 IFS

Introduced by BARNSLEY[7] in 1988, the IFS (Iterated Function Systems) model generates a geometrical shape or an image [8] with an iterative process. An IFS-based modeling system is defined by a triple $(\mathcal{X}, d, \mathcal{S})$ where:

- $(\mathcal{X}, d)$ is a complete metric space, $\mathcal{X}$ is called *iteration space*;

- $\mathcal{S}$ is a semigroup acting on points of $\mathcal{X}$ such that: $\lambda \in \mathcal{X} \mapsto T\lambda \in \mathcal{X}$ where T is a contractive operator, $\mathcal{S}$ is called *iteration semigroup*.

An *IFS* $\mathbb{T}$ *(Iterative Function System)* is a finite subset of $\mathcal{S}$: $\mathbb{T} = \{T_0, ..., T_{N-1}\}$ 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:

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

This operator is contractive in the new complete metric space $\mathcal{H}(\mathcal{X})$ and admits a fixed point, called *attractor* [7]:

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

2.2 Parameterisation of attractors

By introducing a finite set Σ, the IFS can be indexed $\mathrm{T} = (T_i)_{i\in\Sigma}$ and the attractor $\mathcal{A}(\mathbb{T})$ has an *address function* [7] defined on Σ^ω, the set of infinite words of Σ:

$$\sigma \in \Sigma^\omega \mapsto \phi(\sigma) = \lim_{n\to\infty} T_{\sigma_1}...T_{\sigma_n}\lambda \in \mathcal{X} \text{ with } \lambda \in \mathcal{X} \, . \tag{1}$$

When operators match joining condition [2,3,9], this function defines parameterised curves or surfaces. For curves, a single indexing $\tilde{\Sigma} = \{0, ..., N-1\}$ is sufficient [10]:

$$\Phi(s) = \phi(\sigma) \text{ with } s = \sum_{i=1}^{\infty} \frac{1}{N^i}\sigma_i$$

where $\sigma = \sigma_1 \dots \sigma_n \dots$ corresponds to the development of the scalar s in base N. For surfaces, it is more convenient to use a double indexing $\Sigma = \{0, \dots, N-1\} \times \{0, \dots, N-1\}$ [11]:

$$\Phi(s,t) = \phi(\rho) \text{ with } \rho = (\sigma_1, \tau_1) \dots (\sigma_n, \tau_n) \dots \in \Sigma^\omega$$

where $\sigma = \sigma_1 \dots \sigma_n \dots$ and $\tau = \tau_1 \dots \tau_n \dots$ are respectively the development of s and t in base N.

2.3 Projected attractors

The main idea of our model is drawn from the formula of free form surfaces used in CAGD:

$$F(s,t) = \sum_{j\in J} \Phi_j(s,t) p_j$$

where p_j constitutes a grid of control points (see Fig. 1), and Φ_j are blending functions. These blending functions have the following property:

$$\forall (s,t) \in [0,1]^2 \ \sum_{j\in J} \Phi_j(s,t) = 1$$

The way to obtain the same property for IFS attractors is to use a barycentric metric space.

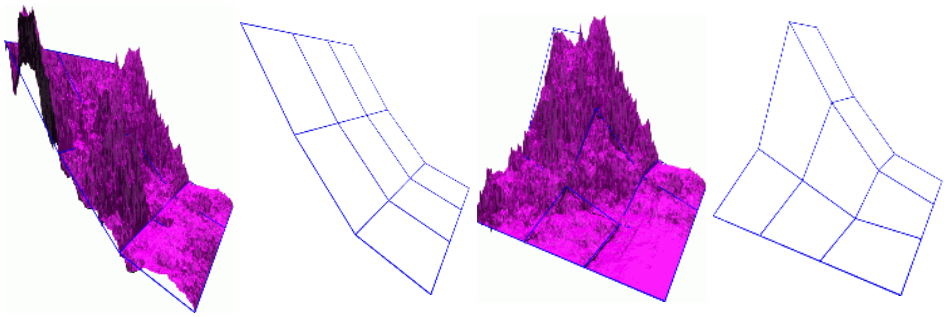


Figure 1. Deformation of a free form surface using the control grid.

In classical fractal interpolation [7,12] or fractal compression [8], the complete metric space $\mathcal{X}$ used is $\mathbb{R}^2$ or $\mathbb{R}^3$, and the iteration semigroup is constituted of contractive affine operators. Our work consists in enlarging iteration spaces [2,3]. This model uses a barycentric space $\mathcal{X} = \mathcal{B}^J$:

$$\mathcal{B}^J = \{(\lambda_j)_{j\in J} \,|\, \sum_{j\in J} \lambda_j = 1\}\,.$$

For curves, this barycentric space is used with $\tilde{J} = \{0,\ldots,m\}$, for surfaces with $J = \{0,\ldots,m\} \times \{0,\ldots,m\}$. Then, the iteration semigroup is constituted of matrices with barycentric columns:

$$S_J = \{T \,|\, \sum_{j\in J} T_{ij} = 1, \forall i \in J\}\,.$$

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

$$P\mathcal{A}(\mathbb{T}) = \{P\lambda \,|\, \lambda \in \mathcal{A}(\mathbb{T})\}$$

where P is a grid of control points $P = (p_j)_{j\in J}$ and $P\lambda = \sum_{j\in J} \lambda_j p_j$. In this way, we can construct a fractal function [2,3,9] using the projection:

$$F(s,t) = P\Phi(s,t) = \sum_{j\in J} \Phi_j(s,t) p_j$$

where $\Phi(s,t)$ is a vector of functions $\Phi(s,t) = (\Phi_j(s,t))_{j\in J}$ and J is the double index set $J = \{0,\ldots,m\} \times \{0,\ldots,m\}$. Fig. 1 shows the action of the control grid on the *free form surface* defined by the function.

2.4 Tabulation of parametric surfaces

With a tabulation process[5,6], considering only the values of s and t multiple of $\frac{1}{N^p}$ leads to a simplification in the computing without any loss of information. The surface tabulation is a grid defined by:

$$F(\frac{i}{N^p}, \frac{j}{N^p}) = P\Phi(\frac{i}{N^p}, \frac{j}{N^p}) \text{ with } (i,j) \in \{0,\ldots,N^p-1\} \times \{0,\ldots,N^p-1\}$$

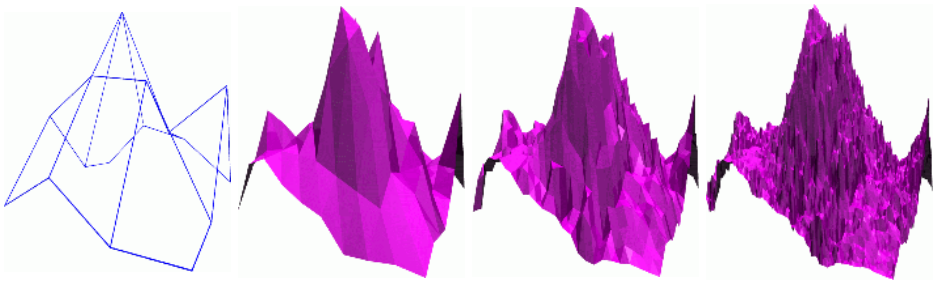


Figure 2. Three first iterations of the construction process beginning with the initial 4×4 control grid (Step 0).

In this special case, developments of $\frac{i}{N^p}$ are ended by a infinite sequences of 0:

$$\begin{cases} \sigma = \sigma_1 \dots \sigma_p 00 \dots \\ \tau = \tau_1 \dots \tau_p 00 \dots \end{cases}$$

Then, $F(\frac{i}{N^p}, \frac{j}{N^p})$ simplifies in:

$$\begin{aligned} F(\tfrac{i}{N^p}, \tfrac{j}{N^p}) &= P\phi\left((\sigma_1, \tau_1) \dots (\sigma_p, \tau_p) \dots (0,0) \dots (0,0) \dots\right) \\ &= PT_{\sigma_1\tau_1} \dots T_{\sigma_p\tau_p}\phi\left((0,0) \dots (0,0) \dots\right) = PT_{\sigma_1\tau_1} \dots T_{\sigma_p\tau_p}\Phi(0,0) \end{aligned}$$

By choosing simplifications (but no restrictions) such as $\Phi(0,0) = e_{00}$, the surface tabulation can be generated computing only p iterations without any loss of information:

$$F(\frac{i}{N^p}, \frac{j}{N^p}) = PT_{\sigma_1\tau_1} \dots T_{\sigma_p\tau_p} e_{00}$$

Fig. 2 shows the three first iterations of the construction process.

3 Surface representation

Any couple $(P, \mathbb{T})$ does not describe a parametric surface $F(s,t)$. A joining condition must apply to the IFS to be sure the generated object is a surface. Classically, this joining condition is obtained by tensor products of curves. This method, very used in CAGD, can be extended to surfaces defined by IFS. Therefore, it can be applied to the modeling of fractal surfaces [2,3,11,12]. The function $\Phi(s,t) = \tilde{\Phi}^u(s) \otimes \tilde{\Phi}^v(t)$ is defined by the IFS $T_{i,j} = \tilde{T}_i^u \otimes \tilde{T}_j^v$.

However, the fractal familly of surfaces defined by IFS is much larger. This section explains in details how it is possible to provide a joining condition such that the function $(s,t) \in [0,1]^2 \mapsto \Phi(s,t)$ is well defined [9].

3.1 Joining condition

The joining condition of the quadrangular patch is drawn from the structure of $[0,1]^2$ (see fig. 3).

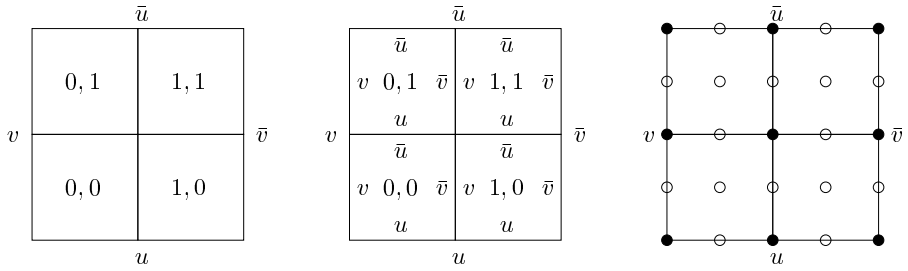


Figure 3. Quadrangular subdivision scheme: boundaries, joinings and grid.

Boundary curves $\Phi_u(s) = \Phi(s,0)$, $\Phi_{\bar{u}}(s) = \Phi(s,1)$, $\Phi_v(s) = \Phi(0,s)$, $\Phi_{\bar{v}}(s) = \Phi(1,s)$ are defined by IFS T^γ with indices in $\tilde{\Sigma} = \{0,...,N-1\}$ extracted from patch IFS (see Fig. 3):

$$T_i^\gamma = T_{\xi_i^\gamma} \text{ with } \xi_i^u = (i,0),\ \xi_i^{\bar{u}} = (i,N-1),\ \xi_i^v = (0,i),\ \xi_i^{\bar{v}} = (N-1,i)$$

for $i = 0,...,N-1$. These boundary curves are drawn from the corresponding address functions:

$$\Phi_\gamma(s) = \phi^\gamma(\sigma) = \phi(\xi^\gamma(\sigma)) \text{ with } \xi^\gamma : \sigma \in \tilde{\Sigma}^\omega \mapsto \xi^\gamma_{\sigma_1}...\xi^\gamma_{\sigma_n}... \in \Sigma^\omega$$

A necessary and sufficient condition for the function $\Phi : (s,t) \mapsto \phi(\rho)$ to be defined and to be continuous is that the composants $T_{i,j}\phi$ of the address function ϕ join along their borders $T_{i,j}\phi^\gamma$ (see proof in [9]). This condition is an equation system:

$$\begin{cases} T_{i,j}\phi^{\bar{u}}(\sigma) = T_{i,j+1}\phi^u(\sigma) \ \text{ for } \ i = 0,...,N-1 \text{ and } j = 0,...,N-2 \\ T_{i,j}\phi^{\bar{v}}(\sigma) = T_{i+1,j}\phi^v(\sigma) \ \text{ for } \ i = 0,...,N-2 \text{ and } j = 0,...,N-1 \end{cases}$$

This equation system can be expressed as linear constraints on the $T_{i,j}$ matrices.

3.2 Constraints

Now, consider that each patch boundary is the embedding of a curve defined on a grid $\tilde{J} = \{0,...,m\}$ in the corresponding boundary of the patch control grid (see Fig. 4):

$$\phi^\gamma(\sigma) = \mathrm{II}_\gamma\tilde{\phi}^\gamma(\sigma) = \mathrm{II}_\gamma \sum_{k=0}^{m} \tilde{\phi}_k^\gamma(\sigma)\tilde{e}_k = \sum_{k=0}^{m} \tilde{\phi}_k^\gamma(\sigma)\ \mathrm{II}_\gamma\ \tilde{e}_k \tag{2}$$

The embeddings associated with the grid boundaries are defined by:

$$\mathrm{II}_u\tilde{e}_k = e_{k,0},\ \mathrm{II}_{\bar{u}}\tilde{e}_k = e_{k,m},\ \mathrm{II}_v\tilde{e}_k = e_{0,k},\ \mathrm{II}_{\bar{v}}\tilde{e}_k = e_{m,k} \ \text{ for } k = 0,..,m.$$

Let us denote $\tilde{\mathrm{T}}^\gamma = (\tilde{T}_i^\gamma)_{i=0,...,N-1}$ the IFS that defines $\tilde{\phi}^\gamma$, $T_{i,j}$ matrices must satisfy the following embedding constraints [9]:

$$\phi^\gamma = \mathrm{II}_\gamma\tilde{\phi}^\gamma \ \Leftrightarrow\ T_{\xi_i^\gamma}\mathrm{II}_\gamma = \mathrm{II}_\gamma\tilde{T}_i^\gamma \ \text{ pour } \ i = 0,..,N-1$$

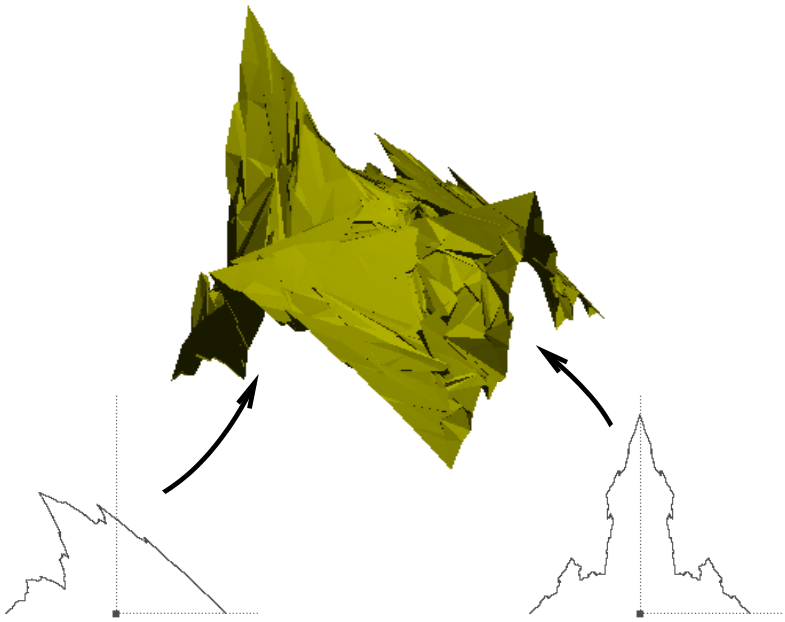


Figure 4. Quadrangular patch and the corresponding boundary curves.

Then, the joining equations can be expressed as constraints on $T_{i,j}$ and on boundary IFS $\tilde{\mathrm{T}}^{\gamma}$ [9]:

$$\begin{aligned} T_{i,j}\phi^{\bar{v}} = T_{i+1,j}\phi^{v} &\Leftrightarrow T_{i,j}\,\Pi_{\bar{v}}\,\tilde{\phi}_{\bar{v}} = T_{i+1,j}\,\Pi_{v}\,\tilde{\phi}^{v} \\ &\Leftrightarrow T_{i,j}\Pi_{\bar{v}} = T_{i+1,j}\,\Pi_{v} \quad \text{et} \quad \tilde{\mathrm{T}}^{\bar{v}} = \tilde{\mathrm{T}}^{v}. \end{aligned}$$

The argument is also valid with $\tilde{\mathrm{T}}^{\bar{u}} = \tilde{\mathrm{T}}^{u}$ (see Fig. 4).

3.3 Description

An IFS defined by N matrices $T_{i,j}$ that satisfy the previous constraints is equivalent to a subdivision scheme characterised by a grid of points (see Fig. 3).

In this way, the classical process of point interpolation to build a fractal curve in $\mathbb{R}^2$ [7] has been extended to fractal surfaces in $\mathbb{R}^3$ [12,13,14].

The proposed grid is composed of $(mN+1)\times(mN+1)$ points of $\mathcal{B}^J$ corresponding to the matrices columns of $T_{i,j}$:

$$T_{i,j}e_{kl} = \lambda_{mi+k,mj+l}$$

The joining equations are satisfied:

$$\begin{aligned} T_{i,j}\Pi_{\bar{v}} = T_{i+1,j}\Pi_{v} &\Leftrightarrow \forall l = 0,..,m \;\; T_{i,j}\,\Pi_{\bar{u}}\,\tilde{e}_l = T_{i+1,j}\,\Pi_{u}\,\tilde{e}_l \\ &\Leftrightarrow \forall l = 0,..,m \;\; T_{i,j}e_{m,l} = T_{i+1,j}e_{0,l} \\ &\Leftrightarrow \forall l = 0,..,m \;\; \lambda_{mi+m,mj+l} = \lambda_{m(i+1),mj+l} \end{aligned}$$

However, embedding equations are expressed as restrictions on grid boundaries:

$$\begin{aligned} T_{\xi_j^v}\Pi_v = \Pi_v\tilde{T}_j^v &\Leftrightarrow \forall l=0,...,m \;\; T_{\xi_j^v}\,\Pi_v\,\tilde{e}_l = \Pi_v\tilde{T}_j^v\tilde{e}_l \\ &\Leftrightarrow \forall l=0,...,m \;\; T_{0,j}e_{0l} = \Pi_v\tilde{T}_j^v\tilde{e}_l \\ &\Leftrightarrow \forall l=0,...,m \;\; \lambda_{0,mj+l} = \Pi_v\tilde{\lambda}_{mj+l}^v \end{aligned}$$

where $(\tilde{\lambda}_k^v)_{k=0,..,mb}$ is a set of points of $\mathcal{B}^{\tilde{J}}$ that characterise the curve $\tilde{\Phi}^v$.

The points $\lambda_{mi,mj}$ are transformed patch corners $T_{i,j}e_{00} = \lambda_{mi,mj}$, $T_{i,j}e_{m0} = \lambda_{mi+m,mj}$, $T_{i,j}e_{mm} = \lambda_{mi+m,mj+m}$, $T_{i,j}e_{0m} = \lambda_{mi,mj+m}$. If one takes $N = m$, a pretty good choice would be $\lambda_{mi,mj} = e_{ij}$. Changing of system coordinates R with $Re_{kl} = r_{kl}$ allows to consider the general case.

The choice of the description grids $(mN+1)\times(mN+1)$ is not any. Practically, the points are taken in the subspaces corresponding to parts of the control grid [6].

Considering the coefficients of the operators T_i and the coordinates of the control grid P as elements of a parameter vector $\mathbf{a}$ allows us to construct a family of functions $F_{\mathbf{a}}$ defined by couples $(P_{\mathbf{a}}, \mathbb{T}_{\mathbf{a}})$.

4 Approximation

Given a sampled surface $(s_i, t_j, \mathbf{Q}_{ij}) \in \mathbb{R}^3$, the challenge is to determine the projected IFS model which provides a good quality approximation of this surface. The approach proposed in the current study is similar to the one we introduced in [4,5] for curves. It is based on a non-linear fitting formalism. Then, we show how the approximation problem can be seen as a standard non-linear fitting problem.

4.1 Non linear fitting

Let $\mathbf{Q}_{ij\,(i=0,...,N^p\;\,j=0,...,N^p)}$ be a given surface to approximate. Let $F_{\mathbf{a}}$ be the function associated with the parameter vector $\mathbf{a}$. The approximation problem consists in determining the parameter vector $\mathbf{a}$ that minimizes the distance between the sampled surface $\mathbf{Q} = \{(\frac{i}{N^p}, \frac{j}{N^p}, \mathbf{Q}_{ij})\}$ and the function $F_{\mathbf{a}}$:

$$\mathbf{a}_{opt} = \underset{\mathbf{a}}{\operatorname{argmin}}\; d(\mathbf{Q}, F_{\mathbf{a}})$$

where:

$$d(\mathbf{Q}, F_{\mathbf{a}}) = \sum_{ij} ||\mathbf{Q}_{ij} - F_{\mathbf{a}}(\frac{i}{N^p}, \frac{j}{N^p})||^2$$

4.2 Solving the non-linear fitting problem

Our resolution method is based on the Levenberg-Marquardt algorithm [15]. This algorithm is a numerical resolution of the following fitting problem:

$$\mathbf{a}_{opt} = \underset{\mathbf{a}}{\operatorname{argmin}} \sum_{i=0}^{M} (v_i - f(\mathbf{a}, u_i))^2$$

where vectors $\mathbf{v}$ and $\mathbf{u}$ 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 data:

$$\mathbf{v} = (v_0, \ldots, v_M) = (0, \ldots, 0) \; ; \; \mathbf{u} = (u_0, \ldots, u_M) = (0, \ldots, M)$$

where $M = (N^p + 1)^2$. Then, the fitting model is:

$$f(\mathbf{a}, k) = ||\mathbf{Q}_{i_k j_k} - F_{\mathbf{a}}(\frac{i_k}{N^p}, \frac{j_k}{N^p})||$$

where $i_k = k \mod N^p$ and $j_k = k/N^p \;\; \forall k = 0, \ldots, (N^p + 1)^2$.

The Levenberg-Marquardt method combines two types of approximation for minimizing the square distance. The first consists in a quadratic approximation. When this fails, the method tries a simple linear approximation. These approximations are computed with the provided partial derivatives of the fitting model. In our case, these partial derivatives are numerically computed by a perturbation vector [4,5]:

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

Then, the computation of partial derivatives is approximated by:

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

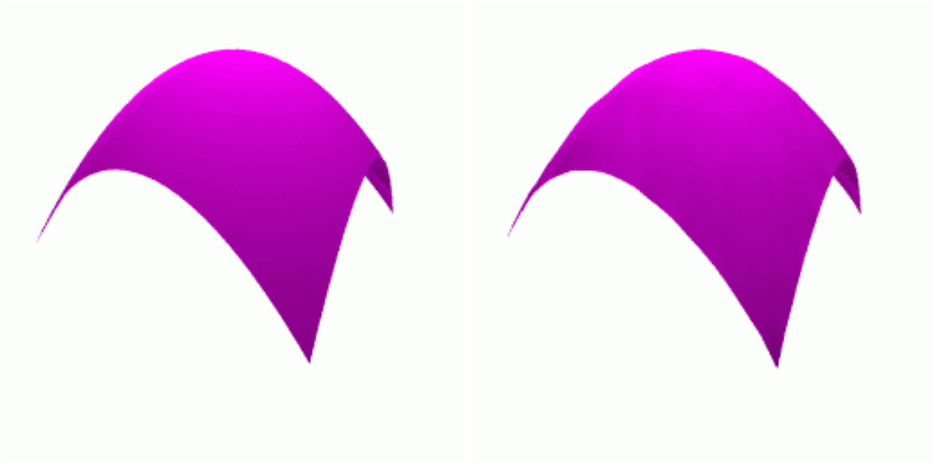
4.3 Results

We show three examples of approximation. The model used is composed of a 3×3 patch IFS and a 4×4 control grid. A total of 232 parameters is needed to code the entire model (subdivision schema and control grid). Fig. 5 shows the original elliptic paraboloid shape and the approximated surface reconstructed with our method. The equation of this shape is:

$$z = z_0 - (\frac{x - x_0}{\sigma_x})^2 - (\frac{y - y_0}{\sigma_y})^2$$

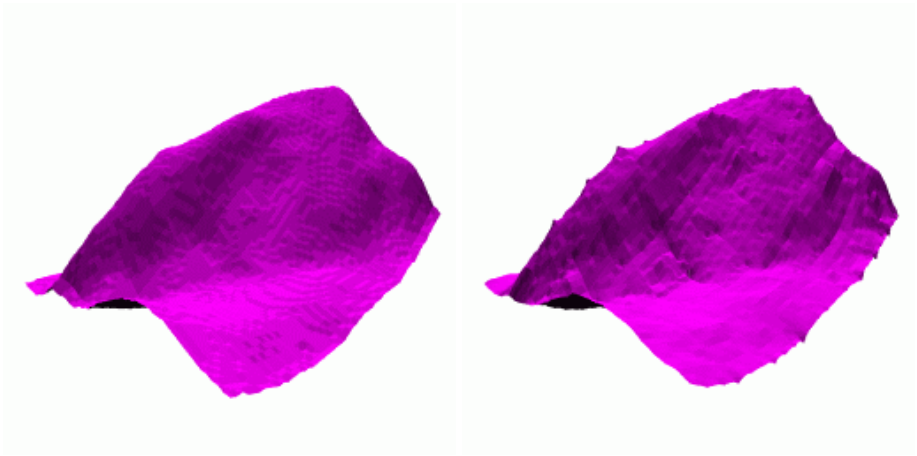
Note that the original and the approximated shapes are very similar. Our model allows to reconstruct smooth surfaces[2,3], and not only rough surfaces. Fig. 6 shows our first experiments on a natural surface. The original surface has been extracted from a geological database (found at the United States Geological Survey Home page `http://www.usgs.org`). The general aspects of the approximated surfaces are similar to the original ones.

A grey-level image can be seen as a surface. Fig. 7 shows the approximation of a grey-level image with both image and surface point of view. This image is an astronomy picture (a white dwarf). Fig. 8 shows the comparison between our approximation method and JPEG algorithm. Table 1 gives numerical results related to this comparison in terms of error (PSNR) and compression ratio. To obtain the same error, the JPEG algorithm provide a compression ratio that is lower than the one obtained with our method. We can also see that the performance is not very altered when each model parameter is quantified to 8 bits.



Original Approximation

Figure 5. Smooth surface.

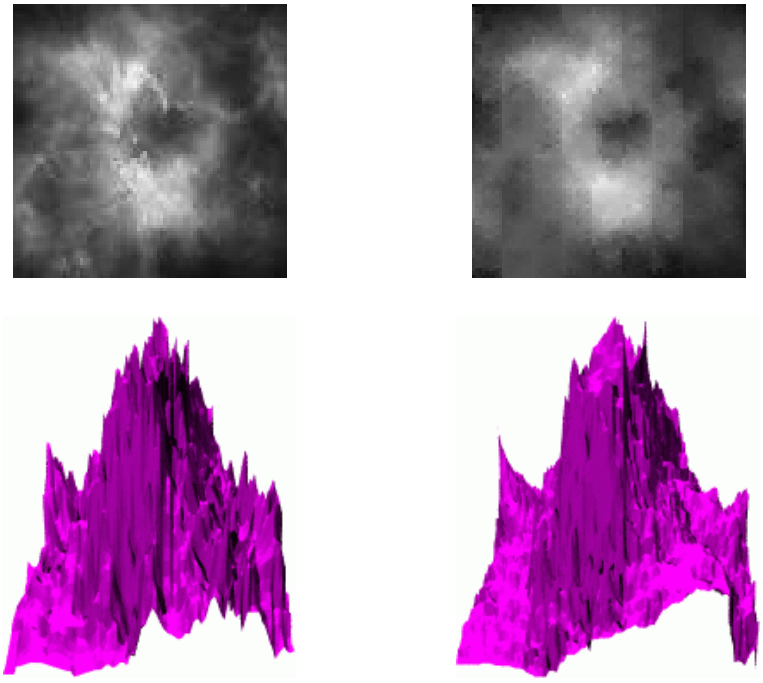


Original Approximation

Figure 6. Geological surface.

5 Conclusion

We presented a new approach for modeling and approximating both rough or smooth objects. This method is based on a fractal model named projected IFS attractors. This model is a parametric description which has the advantage of compactly describing the surface shape making it useful for geometric modeling and image synthesis. First results show that our approximation method is an interesting approach for rough object reconstruction and grey-level image description.



Original Approximation

Figure 7. Grey-level image.

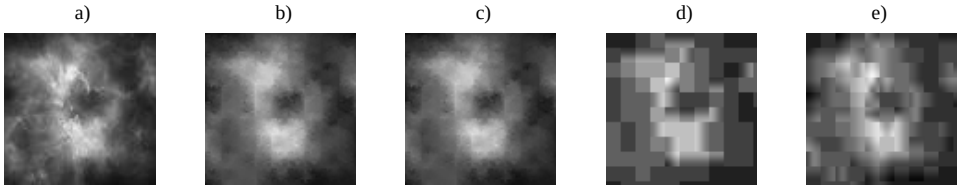


Figure 8. Compression of the original image (a) with our model (b) and (c), and with JPEG algorithm (d) and (e).

Compression type	PSNR (dB)	Code length	Compression ratios
Projected IFS model (b) 16 bits / scalar	26.6	464 bytes	14.5
Projected IFS model (c) 8 bits / scalar	26.4	232 bytes	29
JPEG (d)	23.6	545 bytes	12.3
JPEG (e)	26.5	592 bytes	11.4

Table 1. PSNR and compression ratios

References

1. R. M. Bolle and B. C. Vemuri, *On Three-Dimensional Surface Reconstruction Methods*, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **13**, 1, 1 (1991).
2. C. E. Zair and E. Tosan, *Fractal modeling using free form techniques*, *Computer Graphics Forum* **15**, 3, 269 (1996), EUROGRAPHICS'96 Conference issue.
3. C. E. Zair and E. Tosan, *Computer Aided Geometric Design with IFS techniques*, in M. M. Novak and T. G. Dewey, eds., *Fractals Frontiers*, 443–452 (World Scientific Publishing, 1997).
4. E. Guérin, E. Tosan and A. Baskurt, *Fractal coding of shapes based on a projected IFS model*, in *ICIP 2000*, volume II, 203–206 (2000).
5. E. Guérin, E. Tosan and A. Baskurt, *A fractal approximation of curves*, *Fractals* **9**, 1, 95 (2001).
6. E. Guérin, E. Tosan and A. Baskurt, *Fractal Approximation of Surfaces based on projected IFS attractors*, in *Proceedings of EUROGRAPHICS'2001, short presentations* (2001).
7. M. Barnsley, *Fractals everywhere* (Academic Press, 1988).
8. A. E. Jacquin, *Image coding based on a fractal theory of iterated contractive image transformations*, *IEEE Trans. on Image Processing* **1**, 18 (1992).
9. E. Tosan, *Surfaces fractales définies par leurs bords*, in L. Briard, N. Szafran and B.Lacolle, eds., *Journées "Courbes, surfaces et algorithmes", Grenoble* (1999).
10. H. Prautzsch and C. A. Micchelli, *Computing curves invariant under halving*, *Computer Aided Geometric Design* , 4, 133 (1987).
11. C. A. Micchelli and H. Prautzsch, *Computing surfaces invariant under subdivision*, *Computer Aided Geometric Design* , 4, 321 (1987).
12. P. Massopust, *Fractal Functions, Fractal Surfaces and Wavelets* (Academic Press, 1994).
13. P. Massopust, *Fractal surfaces*, *Journal of Mathematical Analysis and Applications* , 151, 275 (1990).
14. H. Xie and H. Sun, *The study of Bivariate Fractal Interpolation Functions and Creation of Fractal Interpolated Surfaces*, *Fractals* **5**, 4, 625 (1997).
15. W. H. Press, B. P. Flannery, S. A. Teukolsky and W. T. Vetterling, *Numerical Recipes in C : The Art of Scientific Computing* (Cambridge University Press, 1993).