Fractal/wavelet model as a deformation tool

Houssam Hnaidi, Eric Guérin, Samir Akkouche

Research Report
Fractal/wavelet model as a deformation tool

Houssam Hnaidi, Eric Guérin, Samir Akkouche
LIRIS- University of Claude Bernard Lyon 1

Abstract

A Fractal model equipped with detail concept like that used in wavelet transforms is introduced and used to perform global and local deformations to objects. This fractal model based on Projected IFS attractors allows the definition of free form fractal shapes controlled by a set of points. The details concept taken from wavelet theory represents the geometric texture of the object. An approximation step is first done to fit the model to the object, this step is formulated as a non-linear fitting problem and resolved using a modified Levenberg-Marquardt minimization method. Global deformation can be achieved by moving control points or scaling and/or rotating the details extracted from the object. In the same manner local deformation can be applied with additional control points. In this work, we focus on 2D curves.

1 Introduction

Deformation techniques has been taken a lot of attention in geometric modeling. Various methods have been proposed to deform objects. Free form models are a powerful tool for deformation issues, it was used by a lot of deformation methods. Fractal models are very efficient tools to generate natural objects, one of these models is called Projected IFS (Iterated Function System). It employs the concept used in free form modeling for generating fractal objects. With this property, projected IFS can be used as a deformation tool. In the other hand wavelet transform is an important tool for analyzing objects at different scales and precision. With these two models, projected IFS and Wavelet transform, we can change the representation of objects from ordered sampling points to a set of control points and a vector of details. This new representation is efficient and flexible and it can be easily used for deformation issues.

In this work we propose a fractal model equipped with detail concept like that used in wavelet transforms. Our model can perform exact reconstruction, visualization in multi-resolution, and approximation of objects. It also appears to be a good deformation tool.

2 Related work

Barr [Bar84] was the first to introduce the current notion of geometric deformation. Free Form Deformation (FFD) was introduced by Sederberg and Parry [SP86] and consists in enclosing the geometric model within a parallelepiped lattice of control vertices. This deformation method does not depend on the deformed object but on a new model proposed to deform it.

Fractal geometry is an efficient tool for generating self-similar objects. The IFS model [Bar88] is one of its models. Hutchinson [Hut81] and Barnsley [Bar88] developed this formalism and used it in a whole series of applications, in computer graphics and image compression. Zair [ZT96] introduced projected IFS, which uses the concept of free forms (like Bezier Curves) and allows intrinsic deformation capabilities. The usage of projected IFS to approximate real word objects poses the inverse problem which is addressed by Guerin [GTB01]. However, this method does not allow exact reconstruction of any given object.

Fractal models have an intrinsic self-similarity property: the object is composed of parts which resemble it. Wavelet Theory [Mey87, Mal89] is useful for studying that property. Although wavelets are effective for the analysis and synthesis of objects, the functional used (wavelet function and scale function) depends on the target application, rather than the object itself. The self-similarity is a common property between IFS and wavelets. That is why several people used them together in order to analyze the object's self-similarity [DL92].

Our work is based on projected IFS and wavelet theory. We try to take advantage of the two models to develop our model.

3 Theoretical model

We employed a model based on IFS theory and specially on the projected IFS. An IFS is a finite set of contracting mappings defined on a metric space [Bar88]. Projected IFS mixes free forms with IFS model. The principal idea of free forms is to separate the function that represents a curve or a surface in two parts: control polygon and blending functions. When IFS is defined on a barycentric metric space, it's attractor plays the role of the blending functions of the free forms [ZT96]. An IFS is defined as follows.

Let $(\mathcal{X}, d)$ be a complete metric space, we call IFS a finite set $\mathcal{T} = \{T_0, \ldots, T_{N-1}\}$ of contracting mappings on $\mathcal{X}$. This proposition allows to associate to this set a mapping [ref] which is contractive in the complete metric space $(\mathcal{H}(\mathcal{X}), d_H)$ (where $\mathcal{H}(\mathcal{X})$ is the set of all subsets of $\mathcal{X}$ and d_H is the Hausdorff distance). We can then apply the fixed point theorem [Bar88]. For all IFS $\mathcal{T}$ there exists a non-empty compact set A of $\mathcal{H}(\mathcal{X})$ such that:

$$\begin{aligned} A &= \mathcal{T}A \\ &= T_0 A \cup \cdots \cup T_{N-1} A. \end{aligned}$$

A is called the attractor of $\mathcal{T}$ and is denoted $\mathcal{A}(\mathcal{T})$. By indexing the IFS $\mathcal{T} = \{T_0, \ldots, T_{N-1}\}$ with an alphabet $\Sigma = \{0, \ldots, N-1\}$, the address function can be defined as:

$$\begin{aligned} \phi : \Sigma^w &\to \mathcal{X} \\ \theta &\mapsto \phi(\theta) = \lim_{j \to \infty} T_{\theta_1} \ldots T_{\theta_j} \lambda \end{aligned}$$

where Σ^w is the set of infinite words of Σ. The limit formula always exists and is unique for all $\lambda \in \mathcal{X}$ [Bar88]. If we take an IFS defined on the barycentric space $\mathcal{B}^{\mathcal{J}} = \left\{ (\lambda_j)_{j \in \mathcal{J}} \,|\, \sum_{j \in \mathcal{J}} \lambda_j = 1 \right\}$ where $\mathcal{J}$ is a set of indices, we can project the attractor through control points.

Let $\mathcal{J}$ be a set of indices, $\mathcal{T}$ an IFS consisting of mapping taken from from $\mathcal{S}_{\mathcal{J}}$ (where $\mathcal{S}_{\mathcal{J}}$ is a semi group of barycentric matrices defined by $\mathcal{S}_{\mathcal{J}} = \left\{ T | \sum_{j \in \mathcal{J}} T_{ij} = 1, \forall i \in \mathcal{J} \right\}$), and $\mathcal{P} = (p_j)_{j \in \mathcal{J}}$ be a set of control points. The projected IFS attractor associated to $\mathcal{T}$ and $\mathcal{P}$ is defined by:

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

where $\mathcal{P}\lambda = \sum_{j \in \mathcal{J}} \lambda_j p_j$.

A simple formula for visualization of projected IFS can be written [GTB01]:

$$(\mathcal{S}_n)_{n \in \mathcal{N}} = \begin{cases} \mathcal{S}_0 = \{\mathcal{P}\} \\ \mathcal{S}_{n+1} = \mathcal{S}_n \mathcal{T}, \forall n \in \mathcal{N} \end{cases}$$

where $\mathcal{S}_n$ represents a finite set of control polygons. We can represent it by $\mathcal{S}_n = \mathcal{P}\mathcal{T}^n = \{\mathcal{P}T_{\theta_1} \ldots T_{\theta_n} | \, |\theta| = n\}$. Let denote $T_\theta = T_{\theta_1} \ldots T_{\theta_n}$ and $\mathcal{P}_\theta = \mathcal{P}T_\theta$, we can write:

$$\begin{aligned} \mathcal{P}_{\theta i} &= \mathcal{P}T_\theta T_i \\ &= \mathcal{P}_\theta T_i \qquad \text{where} \qquad i \in \Sigma \end{aligned} \tag{1}$$

Inspired by the work of Tosan and al. [TBSS+07] we can add a detail part to the right side of the formula (1) as the following

$$\mathcal{P}_{\theta i} = \mathcal{P}_\theta T_i + \delta \mathcal{P}_\theta U_i \tag{2}$$

where $\delta\mathcal{P}_\theta$ is a detail vector associated to the point set $\mathcal{P}_\theta$ and U_i is a matrix of displacement of details. If we consider n control points and N transforms and we work in $\mathcal{R}^d$ then the matrix dimension of T_i is $n \times n$, of U_i is $n.(N-1) \times n$, of $\mathcal{P}_\theta$ is $d \times n$, and of $\delta\mathcal{P}_\theta$ is $d \times n.(N-1)$. Hence, the concatenation of T_i and U_i matrices forms a square matrix called R (see (5)). If we write the formula (2) for all $i \in \Sigma$ then we have

$$\begin{aligned} \mathcal{P}_{\theta 0} &= \mathcal{P}_\theta T_0 + \delta\mathcal{P}_\theta U_0 \\ \vdots \quad &\quad \vdots \qquad\quad \vdots \\ \mathcal{P}_{\theta N-1} &= \mathcal{P}_\theta T_{N-1} + \delta\mathcal{P}_\theta U_{N-1} \end{aligned} \tag{3}$$

The matrix form of equations (3) can be written as:

$$(\mathcal{P}_{\theta 0}|\ldots|\mathcal{P}_{\theta N-1}) = (\mathcal{P}_\theta|\delta\mathcal{P}_\theta) \left(\begin{array}{c|c|c} T_0 & \ldots & T_{N-1} \\ \hline U_0 & \ldots & U_{N-1} \end{array}\right) \tag{4}$$

we set

$$R = \left(\begin{array}{c|c|c} T_0 & \ldots & T_{N-1} \\ \hline U_0 & \ldots & U_{N-1} \end{array}\right) \tag{5}$$

We can write now our formula as the following:

$$(\mathcal{P}_{\theta 0}|\ldots|\mathcal{P}_{\theta N-1}) = (\mathcal{P}_\theta|\delta\mathcal{P}_\theta)\, R \tag{6}$$

and

$$(\mathcal{P}_\theta|\delta\mathcal{P}_\theta) = (\mathcal{P}_{\theta 0}|\ldots|\mathcal{P}_{\theta N-1})\, R^{-1} \tag{7}$$

we can remark that R is like a synthesis filter and R^{-1} is like an analysis filter used in the wavelet transform.

4 Optimization step

In this section we consider that a curve is an ordered set of points. We would like to set the matrix R to be optimal in term of representation of this curve. This means that the representation of the curve with a small amount of detail data should be as close as possible to the given curve. The optimization method is based on the minimization of the distance between the original curve and the curve reconstructed by our model. We propose initial values for the control points and for the matrix R and we take these initial values as parameters of the optimization method. It is important here to note that our model allows us to reconstruct exactly the input data. We have voluntarily omitted a part of details in the reconstruction. We used the Levenberg-Marquardt method [ref] for minimizing this distance. This method is a non linear regression method based on the derivatives of the function of minimization.

The minimization is achieved in two steps:

- In the first step, we minimize the distance between the original curve and the one reconstructed with our model by using the control points positions and the coefficient of matrices $\mathcal{T}$ as parameters. In this step no detail information has been used to reconstruct the curve.

- In the second step, we minimize this distance by taking the coefficients of matrices U_i as parameters. In this step, only a part of detail information has been used (first three levels for example), otherwise the reconstruction will be perfect and there is nothing to minimize.

The details used in the optimization are produced by applying the analysis formula (7) from the initial set of points to the control points.

5 Applying deformations

Hereafter we will describe how we can use our model to apply global or local deformations on a curve represented by an ordered set of sampled points. As a preliminary, we optimize our model with this curve as described before.

5.1 Global deformation

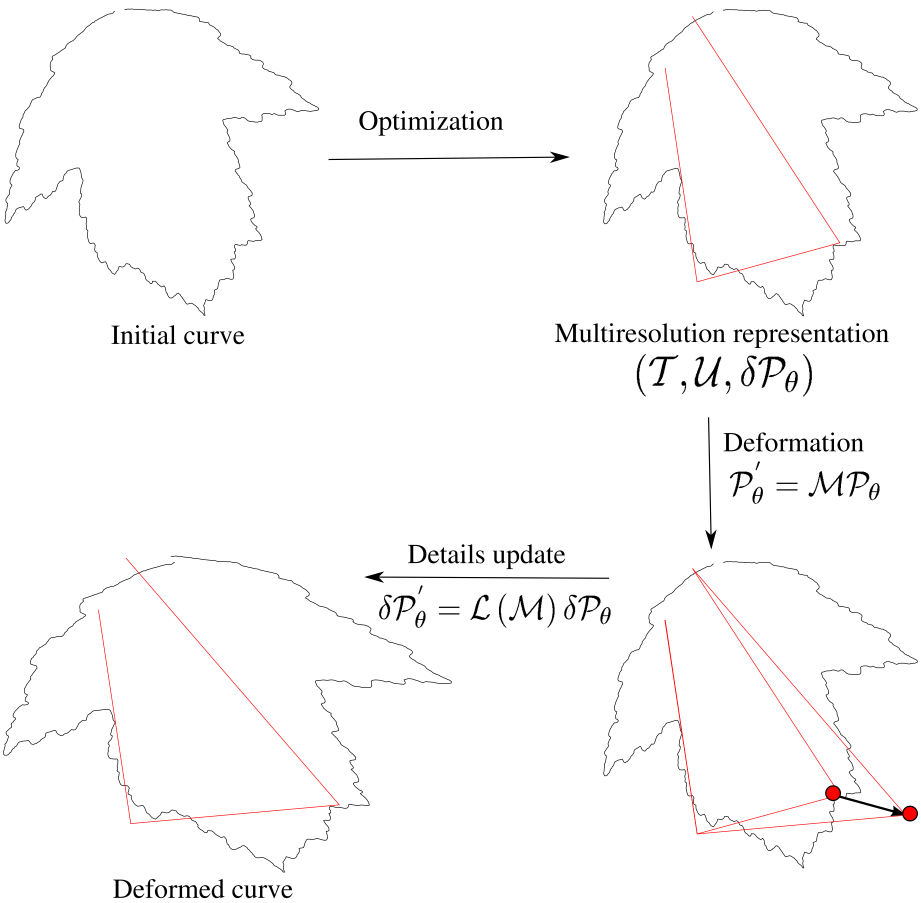

Figure 1: Global deformation steps

Global deformation can be achieved by applying the analysis formula (7) starting by the initial set of points up to the control points (the number of control points is the dimension of the iteration space), then we can move any of the control points to deform the curve.

By applying the analysis formula (7) on the initial set of points, we have a new representation of our curve consisting of a set of control points and a vector of detail's vectors. When we move a control point we must update the vector of details to adapt the deformation.

Details in our model represent the local geometric texture of the curve. They are affected by rotation and scaling but not by translation. For this end we saved the original control points and the original vector of details, and when we move a control point we calculate the transformation matrix M between the original set of control points and the new one by using the pseudo-inverse method, then we extract rotation and scaling parts $L(M)$ and apply them to the original vector of details to compute the new one. The synthesis formula (6) is used to see the deformed curve.

Figure 1 shows the optimization step and the curve deformed by global deformation.

5.2 Appearance deformation

We can employ the vector of details to apply global appearance deformation in the curve without moving any control points by changing the direction of this vector or by amplifying or diminishing it (see Figure 2). Here we don't take into account the first levels of this vector (in our example we deal with the last two levels) because first levels contribute in the curve form. In contrast last levels really represent local geometric features.

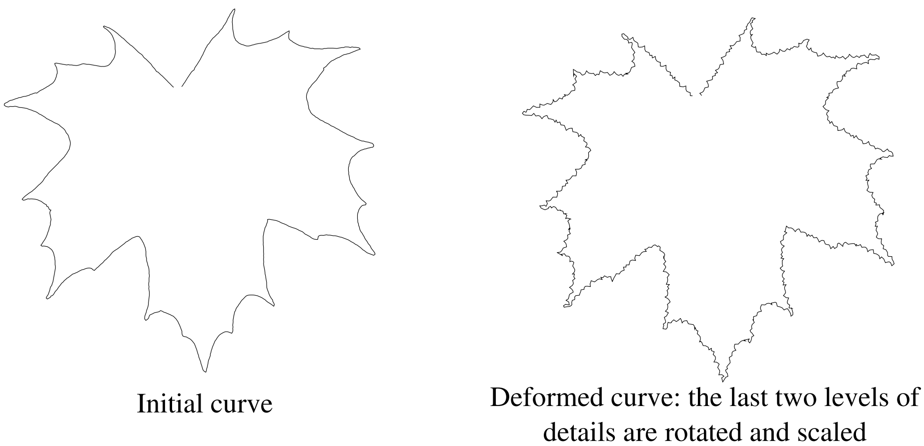

Figure 2: Global deformation by using details

5.3 Local deformation

We can apply local deformation on the curve as the following, first we apply the analysis formula (7) on the initial set of points until we reach the control points, then we store the control points and the vector of details. After that we reapply the analysis formula starting from the initial points until a selected level and we consider the points of this level as new control points. When moving a point from these points we follow the next steps:

1. We apply the analysis formula (7) starting form the updated points up to the last levels.
2. Now we have a new set of control points, we calculate the transformation matrix between this set and the original control points.
3. We compute the new vector of details by dividing it into two parts, the first part is a copy of the result vector of details of first step, and the second part will be rotated and scaled according to the transformation matrix.

Figures 3 and 4 show examples of local deformation.

To have a better rendering result, we observe that neighbor points have also to be moved in the same direction. A fraction of the translation vector is applied to these neighbor points diminishing with the distance to the initial control point that was moved. So what we call local deformation it is not really a local deformation.

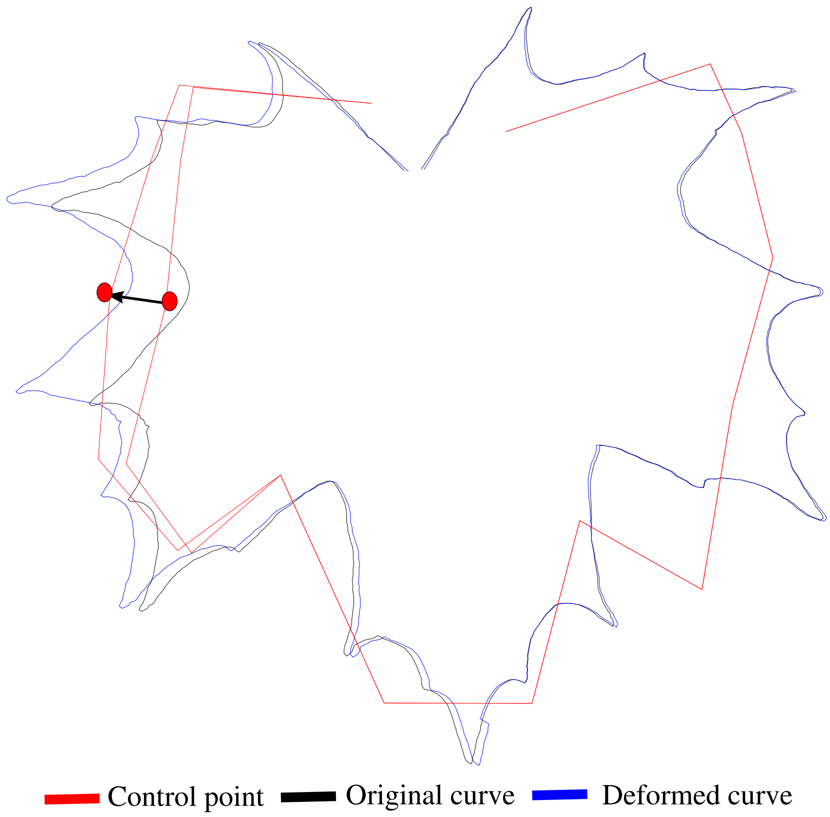

Figure 3: Local deformation

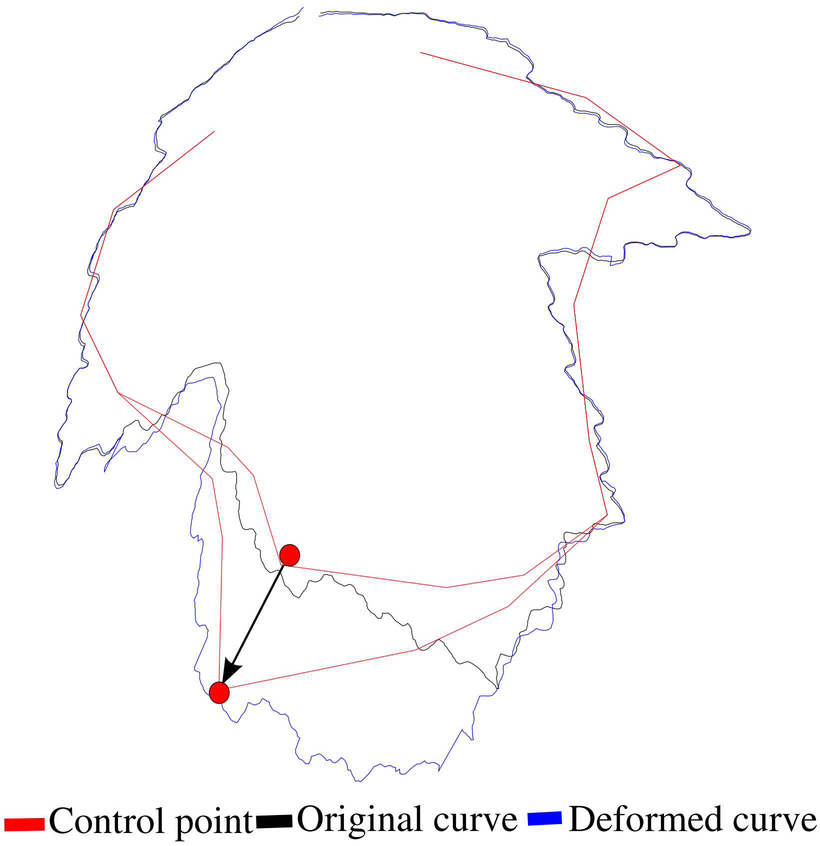

Figure 4: Local deformation

6 Conclusion

We have proposed a fractal model equipped with detail concept and we used it as a deformation tool. This model can be easily used for global deformations by moving control points or by rotating or scaling the vector of details. Local deformations are not totally supported by our model and current work is now to develop it to perform real local deformations. Our model can be used to deal with geometric texture. Dealing with surfaces is one of our future work. Indeed working with height field surfaces is straightforward to our model.

References

[Bar84] Alan H. Barr. Global and Local Deformations of Solid Primitives. *SIGGRAPH84*, 18(3):21–30, July 1984.

[Bar88] Michael Barnsley. *Fractals everywhere.* Academic Press, 1988.

[DL92] Ingrid Daubechies and Jeffrey C. Lagarias. Two-scale difference equations ii. local regularity, infinite products of matrices and fractals. *SIAM J. Math. Anal.*, 23(4):1031–1079, 1992.

[GTB01] Eric Guérin, Eric Tosan, and Atilla Baskurt. A fractal approximation of curves. *Fractals*, 9(1):95–103, mar 2001.

[Hut81] Hutchinson. Fractals and self-similarity. *Indiana University Journal of Mathematics*, 30:713–747, 1981.

[Mal89] Stéphane Mallat. A theory for multiresolution signal decomposition: The waveletrepresentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 11(7):674–693, 1989.

[Mey87] Y. Meyer. Les ondelettes. In *Contributions to nonlinear partial differential equations, Vol. II (Paris, 1985)*, volume 155 of *Pitman Res. Notes Math. Ser.*, pages 158–171. Longman Sci. Tech., Harlow, 1987.

[SP86] Thomas W. Sederberg and Scott R. Parry. Free-form deformation of solid geometric models. In *SIGGRAPH*, pages 151–160, 1986.

[TBSS$^+$07] E. Tosan, I. Bailly-Salins, I. Stotz, G. Gouaty, and Y. Weinand. Modelisation iterative de courbes et surfaces : aspect multiresolution. In *Groupe de travail en Modelisation Geometrique journee de Valenciennes*, pages 55–69, mars 2007.

[ZT96] Chems Eddine Zair and Eric Tosan. Fractal modeling using free form techniques. *Comput. Graph. Forum*, 15(3):269–278, 1996.