Fractal/Wavelet representation of objects

Houssam Hnaidi
University of Claude Bernard Lyon1
Email: houssam.hnaidi@gmail.com

Eric Guérin
University of Claude Bernard Lyon1
Email: eric.guerin@liris.cnrs.fr

Samir Akkouche
University of Claude Bernard Lyon1
Email: samir.akkouche@liris.cnrs.fr

***Abstract*—A Fractal model equipped with detail concept like the one used in wavelet transforms is introduced and used to represent objects in a more efficient way . This new representation can be used to deform object (locally and globally) and to manipulate the geometric texture of these objects. This fractal model based on Projected IFS attractors allows the definition of free form fractal shapes controlled by a set of points. The projected IFS is a type of IFS (Iterated Function System) which mixes free forms models with IFS models. The details concept idea taken from wavelet theory represents the geometric texture of the object. This concept is introduced by wavelet transform. The wavelet transform represents a signal in hierarchic manner. The signal is divided in two parts: one representing the signal in different scales, and the other representing the details of this signal. We proposed a model based on projected IFS and used the idea of details introduced by wavelet theory. 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. Our goal is to change the representation of objects from an ordered set of data(points, pixels,..) to a set of control data and a vector of details such that this new representation facilitate the manipulation of objects. In this work, we focus on 2D curves.**

I. Introduction

Various techniques have been proposed to change the representation of objects from an ordered data set to another form which can be used for a lot of issues. The famous JPEG and JPEG200 compression techniques use this idea for compression issue. The former employs a DCT (Discrete Cosine Transform) to transform the image data form the spatial space to the frequency space and uses the new data to compress the image. The later applies the wavelet transform on the image data and uses the result of this operation as input of a compression algorithm. One of the most known transform techniques is the wavelet theory. Wavelet transform change an object form the ordered data set to a copy of this object at different scales and a vector of details, introducing a multi-resolution representation of the object. This new representation can be used for various operations on the object.

Fractal geometry is very efficient to generate self-similar objects by using some production rules. IFS (Iterated Function System) is a very simple tool to produce fractal figures, it uses some contractive mappings for this issue. Projected IFS is a type of IFS which uses the idea of free form objects (like Bezier curves or surfaces) such that by starting from some control points and some transformations it can recursively generate curves or surfaces. IFS and projected IFS can be used also to represent objects. This operation called the inverse problem consists in approximating the object with a set of transformations and control points. Hereafter we introduce a new method which employs projected IFS and the wavelet theory to change the representation of objects from an ordered data set to a set of control data and a vector of details. This new representation is efficient and fixable.

Thanks to this new representation, our model can perform exact reconstruction, multi-resolution visualization, and approximation of objects.

II. Related Work

Various techniques exist for changing the way of representing objects (curves, surfaces, images), this type of objects is generally represented by an ordered set of points or pixels. Wavelet transform is one of these techniques. In 1980 Morlet and Grossmann introduced the wavelet theory. Wavelet is a kind of mathematical function used to divide a given function or continuous-time signal into different frequency components and study each component with a resolution that matches its scale. Many people used the wavelet theory to introduce a new representation of objects and employed it in different applications. Mallat [1] introduced a multi-resolution representation of images using wavelet functions. Shapiro [2] used this multi-resolution representation and employed it for coding Images. JPEG2000, the famous image compression tool [3], [4] employed the wavelet multi-resolution representation of images for compression issues. Finkelstein [5] described a multi-resolution curve representation, based on wavelets and cubic B-Spline base functions, that conveniently supports a variety of operations: smoothing a curve; editing the overall form of a curve while preserving its details. Elber [6] developed the multi-resolution curve method by adding linear constraints on the curves.

Fractal geometry is an efficient tool for generating self-similar objects. The IFS model [7] is one of these models. Hutchinson [8] and Barnsley [7] developed this formalism and used it in a whole series of applications, in computer graphics and image compression. Many people try to solve the inverse problem for IFS, which means finding an IFS that generates an approximation of a given object. Jacquin [9] introduced a method for image compression by finding an IFS which can generate parts of image by using other parts of the same image. Gurin [10] addressed the inverse problem and presented a solution for it by using another type of IFS called projected IFS [11]. They use it to represent curves by a set of control points and a set of transformations.

Fractal models have an intrinsic self-similarity property: an object is composed of parts which resemble it. Wavelet Theory[12], [1] is useful for studying that property. Although wavelets are efficient 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[13].

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

III. 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[7]. 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[11]. 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 [8] 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 [7]. 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 on $\Sigma = \{0, \ldots, N-1\}$, the address function can be defined as:

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

where Σ^ω is the set of infinite words of Σ.

The limit formula always exists and is unique for all $\lambda \in \mathcal{X}$ [7]. If we take an IFS defined on the barycentric space

$$\mathcal{B}^{\mathcal{J}} = \left\{ (\lambda_j)_{j \in \mathcal{J}} \mid \sum_{j \in \mathcal{J}} \lambda_j = 1 \right\}$$

where $\mathcal{J}$ is a set of indices, we can project the attractor through control points.

Transformations are taken within the semi-group of barycentric matrices $\mathcal{S}_{\mathcal{J}}$ defined by

$$\mathcal{S}_{\mathcal{J}} = \left\{ T \mid \sum_{j \in \mathcal{J}} T_{ij} = 1, \forall i \in \mathcal{J} \right\}$$

Let $\mathcal{P} = (p_j)_{j \in \mathcal{J}}$ be a set of control points. The projected IFS attractor associated to $\mathcal{T}$ and $\mathcal{P}$ is defined by:

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

where $\mathcal{P}\lambda$ is the projection of λ through $\mathcal{P}$:

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

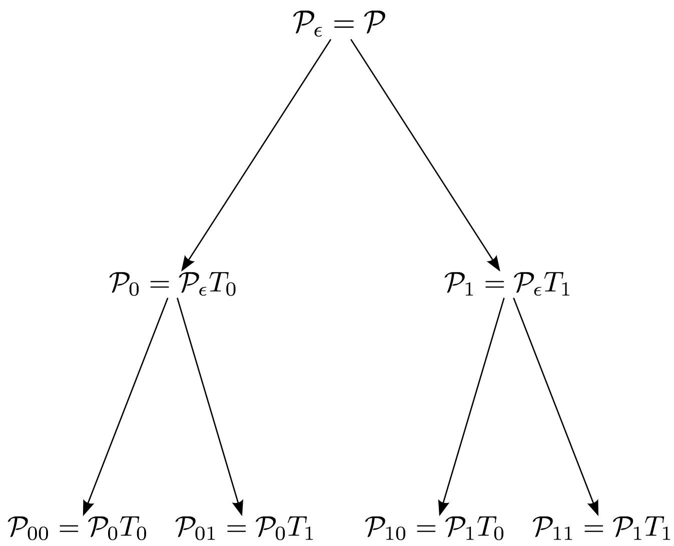


Fig. 1: visualisation tree of Projected IFS

Figure (1) give a simple method to visualize a projected IFS model by using a tree structure which can be represented by the following formula [10]:

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

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

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

Inspired by the work of Tosan et al.[14] 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. Figure (2) shows the modified tree which represents the formula (2).

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

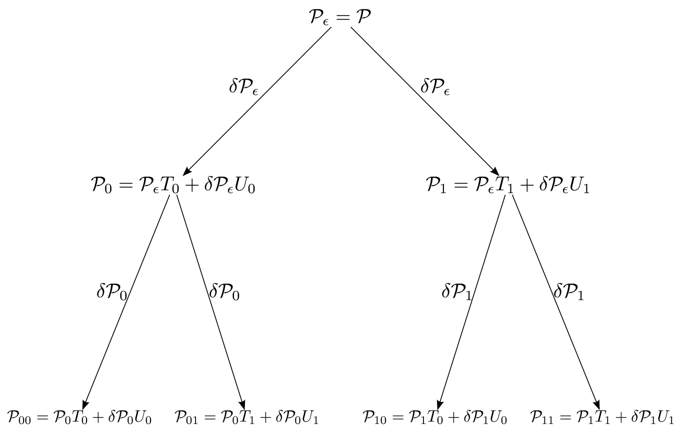


Fig. 2: Visualisation tree of our model

matrix called R (see (5)). If we write the formula (2) for all $i \in \Sigma$ then we have:

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

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

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

we set

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

We can write now our formula as the following:

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

and

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

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

IV. Optimization step

In this section we consider that a curve is an ordered set of points in $\mathbb{R}^2$. 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. To prevent that we have voluntarily omitted a part of details in the reconstruction. We used the Levenberg-Marquardt method [15] for minimizing this distance. This method is a non linear regression method based on the numeric 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 coefficients of matrices $\mathcal{T}$ as parameters. In this step no detail information has been used to reconstruct the curve. In fact in this phase we optimize the control points $\mathcal{P}$ positions and the transformations $\mathcal{T}$ coefficients which represent only the Projected IFS part of our model such that they generate the nearest attractor to the original 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.

V. Applications

A. Deformation

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.

1) Global deformation: 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. 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 3 shows the whole process of global deformation.

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 4). 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.

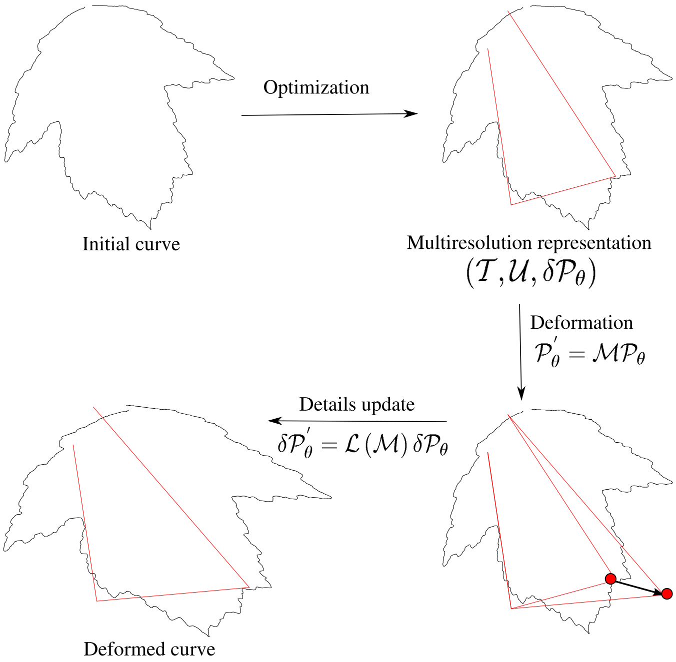


Fig. 3: Global deformation steps

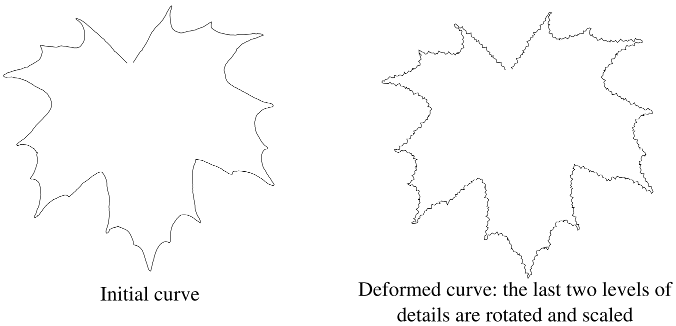


Fig. 4: Global deformation by using details

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 5 and 6 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 is not really a local deformation.

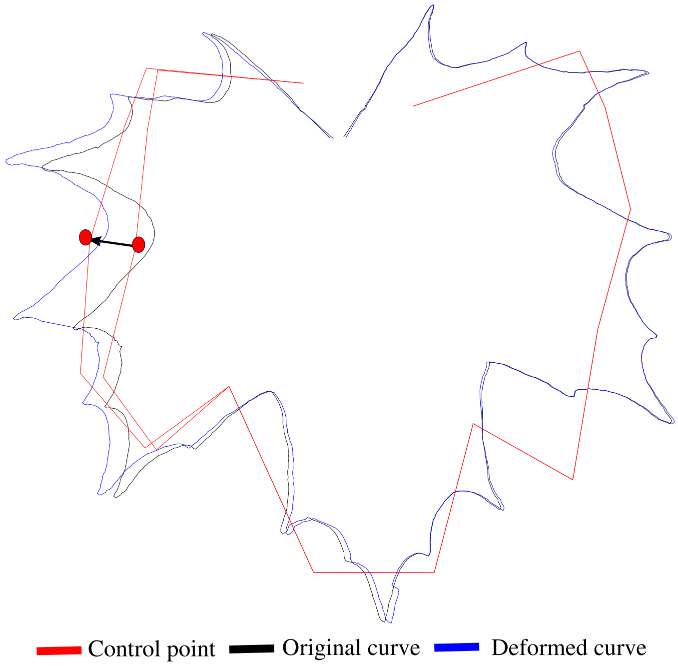


Fig. 5: Local deformation

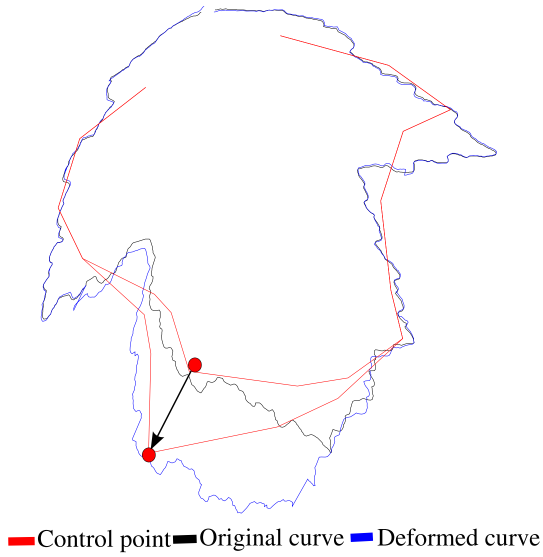


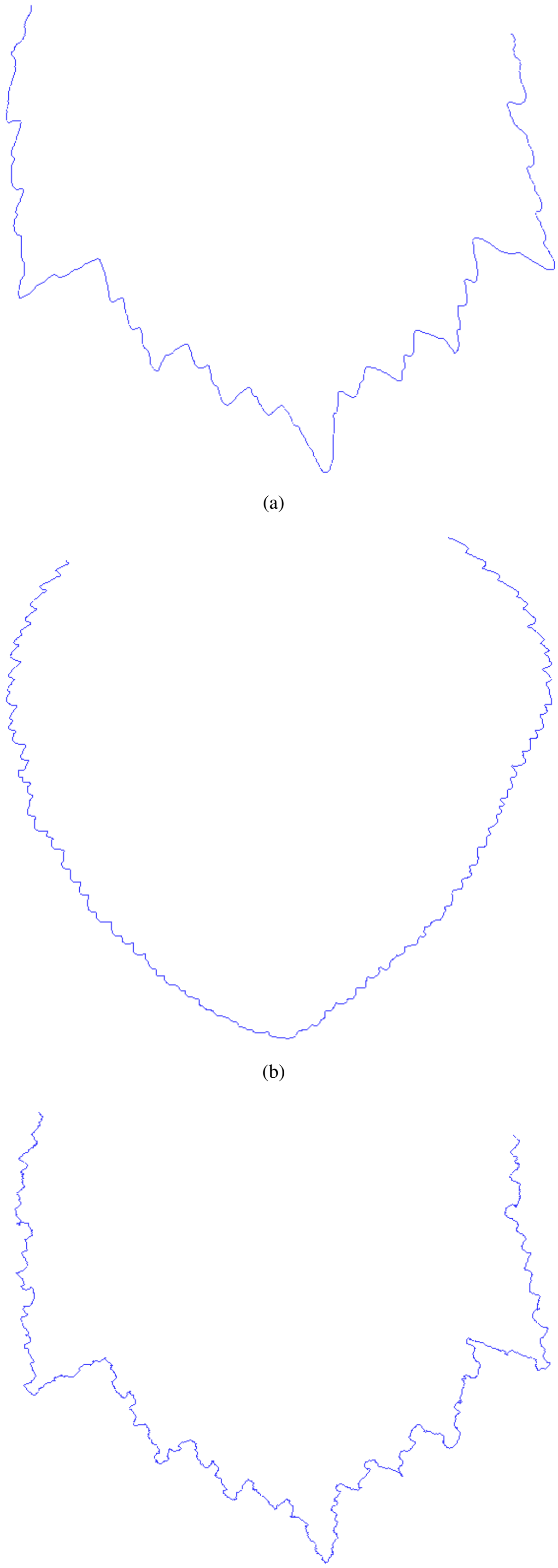
Fig. 6: Local deformation

B. *Transferring of geometric texture*

We can use our model to transfer the geometric texture using the possibility of extracting details. The detail's coefficients (at various levels) may be moved from one object to another. The power and flexibility of our model for the analysis and reconstruction give us the freedom to use it for various applications. For example we optimize our model with an object, Then we apply the model optimized on another object, finally we change the details of the first object by using the details of the second object. The inverse is also offered: we optimizes our model with the second object then we update the details of the first object. Figure (7)

VI. Conclusion

We have proposed a fractal model equipped with detail concept and we used it as a tool for changing the representation of curves from an ordered set of points to a set of control points and details vector. We used this new representation to apply global and local deformation on curves and to transfer geometry texture, but it can also be used in other applications like multi-resolution visualisation. In this work, we focused on curves but dealing with surfaces and images is one of our future work particularly on height field surfaces.



(a) (b) (c)

Fig. 7: transferring the geometric texture. (a) original curve. (b) the curve which contains the texture. (c) the original curve with the texture of the second curve.

References

[1] S. Mallat, "A theory for multiresolution signal decomposition: The wavelet representation." *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 11, no. 7, pp. 674–693, 1989.

[2] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3445–3462, December 1993.

[3] D. S. Taubman, "High performance scalable image compression with ebcot." *IEEE Transactions on Image Processing*, vol. 9, no. 7, pp. 1158–1170, 2000.

[4] T. E. Charilaos Christopoulos, Athanassios Skodras, "The jpeg2000 still image coding system: An overview," *IEEE Transactions on Consumer Electronics*, vol. 46, no. 4, pp. 1103–1127, November 2000.

[5] A. Finkelstein and D. Salesin, "Multiresolution curves," in *SIGGRAPH*, 1994, pp. 261–268.

[6] G. Elber, "Multiresolution curve editing with linear constraints," in *Symposium on Solid Modeling and Applications*, 2001, pp. 109–119.

[7] M. Barnsley, *Fractals everywhere*. Academic Press, 1988.

[8] Hutchinson, "Fractals and self-similarity," *Indiana University Journal of Mathematics*, vol. 30, pp. 713–747, 1981.

[9] A. E. Jacquin, "Image coding based on a fractal theory of iterated contractive image transformations," *IEEE Trans. on Image Processing*, vol. 1, pp. 18–30, jan 1992.

[10] E. Guérin, E. Tosan, and A. Baskurt, "A fractal approximation of curves," *Fractals*, vol. 9, no. 1, pp. 95–103, mar 2001.

[11] C. E. Zair and E. Tosan, "Fractal modeling using free form techniques." *Comput. Graph. Forum*, vol. 15, no. 3, pp. 269–278, 1996.

[12] Y. Meyer, "Les ondelettes," in *Contributions to nonlinear partial differential equations, Vol. II (Paris, 1985)*, ser. Pitman Res. Notes Math. Ser. Harlow: Longman Sci. Tech., 1987, vol. 155, pp. 158–171.

[13] I. Daubechies and J. C. Lagarias, "Two-scale difference equations ii. local regularity, infinite products of matrices and fractals," *SIAM J. Math. Anal.*, vol. 23, no. 4, pp. 1031–1079, 1992.

[14] 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*, mars 2007, pp. 55–69.

[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, ch. Nonlinear Models.