International Journal of Pure and Applied Mathematics
Volume 50 No. 3 2009, 421-430

FRACTAL INTERPOLATION IN MODELING OF 2D CONTOURS

Krzysztof Gdawiec

Institute of Mathematics
University of Silesia
Bankowa 14, Katowice, 40-007, POLAND
e-mail: kgdawiec@math.us.edu.pl

Abstract: The problem of fractal modeling is very simple when we know the mathematical description of a fractal. We just apply one of the well-known algorithms. The inverse problem of finding the mathematical description for given fractal is not so trivial and we do not know any general method to solve this problem. So there are several approaches to this problem e.g. via Bézier curves, fractal compression. In this paper we present automatic method for finding fractal description of 2D contours. Our algorithm uses fractal interpolation for this purpose. We also present some of practical examples.

AMS Subject Classification: 68U05, 68P30, 28A80, 41A05
Key Words: fractal modeling, fractal interpolation, iterated function system, 2D contours

1. Introduction

The notion of fractal was introduced by Benoit Mandelbrot in the 1970's. But fractals existed considerably earlier. They were perceived as exceptional object, mathematical monsters. With time they became a very important tool in many disciplines and found very wide practical applications e.g. image compression, generating shore lines, mountains, clouds, pattern recognition, image processing or in medicine and economy.

One of a such applications is shape modeling. The fractal methods of shape

Received: January 7, 2009

modeling have gained much popularity in recent years. The problems of fractal modeling is very simple when we have fractal description because we use one of the well-known algorithms for generating fractals. The inverse problem, i.e. finding fractal description for a given fractal, is very difficult and we do not know any general method to solve this problem [1]. There are several approaches to this problem. In [5], [6] we find methods for fractal generation of contours. These methods used the fact that every contour can be divided into sum of linear segments and Bézier curves [5] or sum of linear segments and Chaikins curves [6] and the fact that for these curves and linear segments we know fractal description. Using results from [9] we can expand these methods to any parametric curve for which we know the subdivision scheme. In this paper we present a different approach to fractal modeling of contours. In our method we use fractal interpolation.

In Section 2 we introduce the notion of fractal which we will use in this paper. Moreover, we present two algorithms for generating fractals from their description. In Section 3, we introduce the idea of fractal interpolation which we will use in Section 4 in automatic method for finding fractal description of a contour. Furthermore, in Section 4 we present some examples presenting our algorithm. Finally, in Section 5 we present our conclusion and future work.

2. Fractal as Attractor

There are many definitions of a fractal [1], [7], so in this section we introduce the definition that we will use in this paper. But first we must bring in some notations.

Let us take any complete metric space (X, ρ) and denote as $H(X)$ the space of compact subsets of the X. In this space we introduce function $h : H(X) \times H(X) \to \mathbb{R}_+$ which is defined as follows

$$h(R, S) = \max\{D(R, S), D(S, R)\}, \tag{1}$$

where $R, S \in H(X)$ and the mapping $D : H(X) \times H(X) \to \mathbb{R}_+$ is defined as follows

$$D(R, S) = \max_{x \in R} \min_{y \in S} \rho(x, y). \tag{2}$$

It turns out that the function h is a metric (Haussdorf metric) and the space $(H(X), h)$ is a complete metric space [1]. Another important notion in our considerations is the notion of iterated function system (IFS).

We say that a set $W = \{w_1, \ldots, w_n\}$, where w_i is contraction mapping for $i = 1, \dots, n$ is an *iterated function system.* So defined IFS determines the Hutchinson operator which is defined as follows

$$\forall_{A\in H(X)} \quad W(A) = \bigcup_{i=1}^{n} w_i(A) = \bigcup_{i=1}^{n}\{w_i(a) : a \in A\}. \tag{3}$$

This operator is a contraction with contractivity factor $s = \max\{s_1, \dots, s_n\}$, where s_i is contractivity factor for w_i for $i = 1, \dots, n$ [1]. Let us consider the following recurrent sequence

$$\begin{cases} W^0(A) = A \\ W^k(A) = W(W^{k-1}(A)), \quad k \geq 1\,, \end{cases} \tag{4}$$

where $A \in H(X)$.

The following theorem is consequence of the Banach Fixed Point Theorem.

Theorem 1. *Let (X, ρ) be a complete metric space and $W = \{w_1, \dots, w_n\}$ be an IFS. Then exists only one set $B \in H(X)$ such that $W(B) = B$. Furthermore the sequence defined by equation (4) is convergent and*

$$\forall_{A\in H(X)} \quad \lim_{k\to\infty} W^k(A) = B. \tag{5}$$

Now we are ready to give the definition of fractal.

Definition 2. The limit from Theorem 1 is called an *attractor* of the IFS or fractal.

In our further considerations we will need an algorithm for generating the attractor of an IFS. The first algorithm called deterministic method appears in theorem 1 and is the following: we take any $A \in H(X)$ and calculate $W^k(A)$ for $k = 1, 2, \dots$. After several iterations we achieve a good approximation of the attractor [1], [8]. This approximation is sufficient for our purposes but the iteration process is very time consuming. Therefore we will use another algorithm called the chaos game which is one of the fastest algorithms for generating attractor for given IFS [1], [8].

In chaos game we have IFS $W = \{w_1, \dots, w_n\}$ and with each mapping w_i we associate a probability $p_i > 0$ such that $\sum_{i=1}^{n} p_i = 1$. In our case it is sufficient that with each mapping we associate the same probability. In this algorithm first we choose an initial point x_0 and the number of iterations k. Next we pick randomly a mapping from IFS according to the given probability distribution and transform x_0 using this mapping receiving a new point x_1 which we draw. Now x_1 is starting point for next iteration. We repeat this process k times. If the initial point x_0 belongs to the attractor then each of the points generated in the iteration process will also belong to the attractor. In the case when the initial point does not belong to the attractor then a finite number of points generated at the beginning of the iteration process will lay outside the attractor.

3. Fractal Interpolation

In this section we introduce the fundamentals of fractal interpolation which we can find in [1], [4].

Let a set of data $\{(x_i, y_i) \in \mathbb{R}^2 : i = 0, 1, \ldots, N\}$ be given where $x_0 < x_1 < \ldots < x_N$. The points of this set are called interpolation points (knots). We search for IFS in $\mathbb{R}^2$ such that its attractor is the graph of a continuous function $f : [x_0, x_N] \to \mathbb{R}$ such that $f(x_i) = y_i$ for $i = 0, 1, \ldots, N$.

We consider an IFS which consists of N affine mappings of the special structure

$$w_n \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} a_n & 0 \\ c_n & d_n \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} e_n \\ f_n \end{bmatrix} \tag{6}$$

for $n = 1, \ldots, N$. These mappings must satisfy the following constraints

$$w_n \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} = \begin{bmatrix} x_{n-1} \\ y_{n-1} \end{bmatrix} \quad \text{and} \quad w_n \begin{bmatrix} x_N \\ y_N \end{bmatrix} = \begin{bmatrix} x_n \\ y_n \end{bmatrix} \tag{7}$$

for $n = 1, \ldots, N$.

Because the mappings are specified by five numbers a_n, c_n, d_n, e_n, f_n and the constraints give us four linear equations we must choose one of the five numbers to be a free parameter. We choose d_n. In this case after solving the equations we achieve the following formulas

$$a_n = \frac{x_n - x_{n-1}}{x_N - x_0} \tag{8}$$

$$e_n = \frac{x_N x_{n-1} - x_0 x_n}{x_N - x_0} \tag{9}$$

$$c_n = \frac{y_n - y_{n-1}}{x_N - x_0} - d_n \frac{y_N - y_0}{x_N - x_0} \tag{10}$$

$$f_n = \frac{x_N y_{n-1} - x_0 y_n}{x_N - x_0} - d_n \frac{x_n y_0 - x_0 y_N}{x_N - x_0} \tag{11}$$

for $n = 1, \ldots, N$.

Figure 1 presents an example of fractal interpolation for following points $(0, 0)$, $(30, 50)$, $(60, 40)$, $(100, -10)$ and the free parameters 0.5, -0.5, 0.23. The interpolation points are marked as circles and the chaos game with 20000 iterations was used to generate this graph.

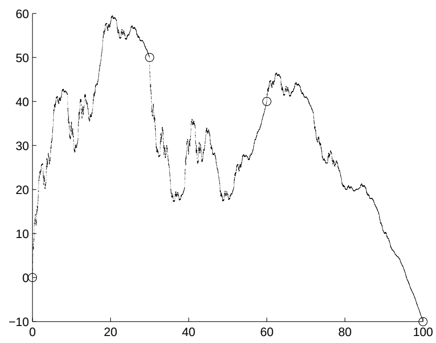

Figure 1: Example of fractal interpolation

This fundamental method of fractal interpolation is not sufficient to be used in generating contours because usually we cannot satisfy the condition $x_0 < x_1 < \ldots < x_n$. In [4] we find a method to interpolate any curve not only that satisfies the mentioned condition. This method is similar to Barnsley's one.

In this method we also have $\{(x_i, y_i) \in \mathbb{R}^2 : i = 0, 1, \ldots, N\}$ as input data. Firstly we extend our points by one co-ordinate $\{(x_i, y_i, t_i) \in \mathbb{R}^3 : i = 0, 1, \ldots, N\}$ where $0 = t_0 < t_1 < \ldots < t_N = 1$ is any parametrization of $[0, 1]$. Now we search for two IFS's $X = \{X_1, \ldots, X_N\}$ and $Y = \{Y_1, \ldots, Y_N\}$ which satisfy the conditions

$$X_i \begin{bmatrix} 0 \\ x_0 \end{bmatrix} = \begin{bmatrix} t_{i-1} \\ x_{i-1} \end{bmatrix} \quad \text{and} \quad X_i \begin{bmatrix} 1 \\ x_N \end{bmatrix} = \begin{bmatrix} t_i \\ x_i \end{bmatrix}, \tag{12}$$

$$Y_i \begin{bmatrix} 0 \\ y_0 \end{bmatrix} = \begin{bmatrix} t_{i-1} \\ y_{i-1} \end{bmatrix} \quad \text{and} \quad Y_i \begin{bmatrix} 1 \\ y_N \end{bmatrix} = \begin{bmatrix} t_i \\ y_i \end{bmatrix}, \tag{13}$$

for $i = 1, \ldots, N$. Next we combine these two IFSs by using set of maps of the form

$$w_i \begin{bmatrix} t \\ x \\ y \end{bmatrix} = \begin{bmatrix} a_i & 0 & 0 \\ c_{x_i} & d_{x_i} & 0 \\ c_{y_i} & 0 & d_{y_i} \end{bmatrix} \begin{bmatrix} t \\ x \\ y \end{bmatrix} + \begin{bmatrix} e_i \\ f_{x_i} \\ f_{y_i} \end{bmatrix} \tag{14}$$

for $i = 1, \ldots, N$ and where indices c_{x_i}, d_{x_i}, f_{x_i} mark the coefficients of the i-th mapping of the X IFS (similarly for c_{y_i}, d_{y_i}, f_{y_i}), a_i, e_i mark the coefficients of the i-th mapping of the X or Y IFS (these two coefficients are equal in both IFSs).

The mappings $w_1, \ldots, w_n$ create IFS for the curve given by the points

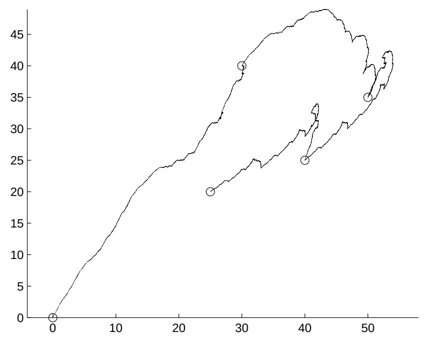

Figure 2: Example of interpolation of arbitrary points

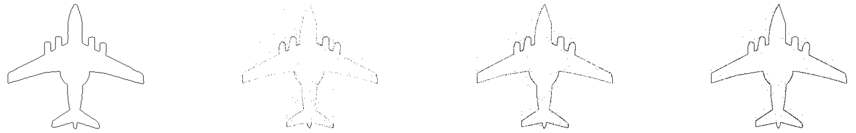

Figure 3: From left to right: original contour, chaos game with 20, 100, 300 iterations per IFS respectively

$\{(x_i, y_i, t_i) \in \mathbb{R}^3 : i = 0, 1, \ldots, N\}$. To obtain the graph of the function interpolating points $\{(x_i, y_i) \in \mathbb{R}^2 : i = 0, 1, \ldots, N\}$ we simply generate points with this IFS and draw on the plane points given by the second and third co-ordinate.

Figure 2 presents an example of interpolation of the following points $(0, 0)$, $(30, 40)$, $(50, 35)$, $(40, 25)$, $(25, 20)$, the free parameters were all equal 0.2 for the X IFS and 0.25 for the Y IFS. The interpolation points are marked as circles and the chaos game with 20000 iterations was used to generate this graph.

4. Fractal Contours

In this section we introduce an algorithm that for a given image finds several IFSs called Partitioned Iterated Function System (PIFS) which approximate contour of an object from this image. To find this fractal description we use fractal interpolation.

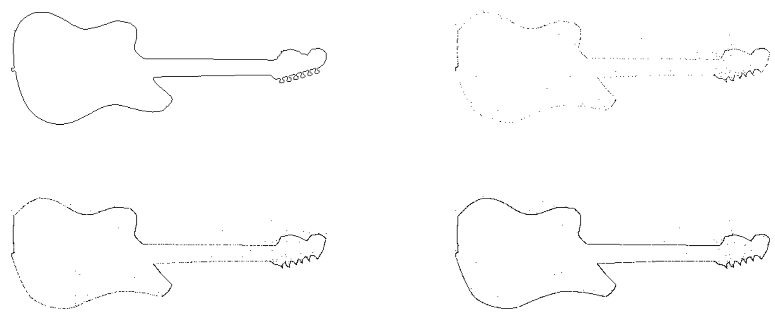

Figure 4: Top (from left): original contour, chaos game with 20 iterations per IFS. Bottom (from left): chaos game with 100, 300 iterations per IFS respectively

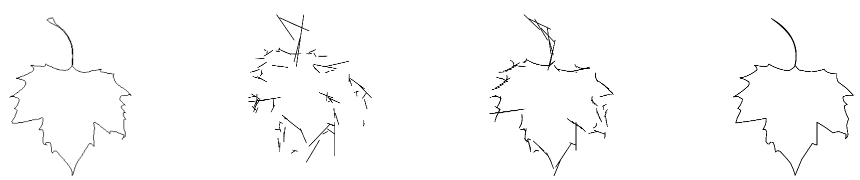

Figure 5: From left to right: original contour, deterministic method with 1, 2, 6 iterations per IFS respectively

For a given image we must do some preprocessing steps which help us in finding fractal description of a contour. Our algorithm consists of three basic steps:

1. Extracting the contour from a given image.
2. Finding points with the highest curvature.
3. Determination of the set of IFSs.

In the first step we simply binarize the image and then find the contour of an object [2]. The second step uses the very efficient IPAN99 [3] algorithm for finding points with the highest curvature. When we have contour and points with the highest curvature we can find the fractal description in the following way. Let us assume that hc is the list of points with high curvature and P is set of the IFSs approximating the contour then:

1. choose error e for approximation of contour

2. $i = 1$, $P = \{\}$

3. while $i < length(hc)$

   (a) $p_1 = hc(i)$, $p_3 = hc(i+1)$,
   $p_2 =$ point laying on contour in half way between p_1 and p_3

   (b) $er = +\infty$, $W = \{\}$

   (c) for $d_x = 0$ to 0.3 with step 0.05
   for $d_y = 0$ to 0.3 with step 0.05

      i. find IFS W_1 interpolating points p_1, p_2, p_3 with free parameters d_x, d_y

      ii. generate approximation of the attractor of the IFS W_1

      iii. calculate error e_1 between approximation of the attractor and a part of contour laying between p_1 and p_3

      iv. if $e_1 < er$ then $er = e_1$, $W = W_1$

   (d) if $er \leq e$ then $P = P \cup W$, $i = i + 1$
   else $hc = \{hc(1), \ldots, hc(i), p_2, hc(i{+}1), hc(i{+}2), \ldots, hc(length(hc))\}$

In our algorithm we use only three points in fractal interpolation so the IFS consist of two mappings. Moreover for each variable the free parameters for these two mappings are equal $d_{x_1} = d_{x_2} = d_x$ and $d_{y_1} = d_{y_2} = d_y$. Of course, in interpolation we can use more points and different values for the free parameters.

In Figure 3 we see first example presenting our algorithm. On left we see original contour of an airplane. Next three images presents approximations of attractor achieved by our algorithm. The PIFS consists from 37 IFSs. To render these images we used chaos game with 20, 100 and 300 iterations per IFS, respectively. Figure 4 presents another example of fractal contour. Like in the first example on left we have original contour of a guitar and next three images are approximations of attractor rendered with the help of chaos game with 20, 100, 300 iterations per IFS. This time PIFS consists from 35 IFSs. The last example (Figure 5) presents contour of a leaf, image on the left. The PIFS achived by our algorithm consist from 41 IFSs. This time to render images we used deterministic method. Starting shape was triangle and we used 1, 2 and 6 iterations per IFS, respectively.

5. Conclusions

In this paper we demonstrated a new algorithm for obtaining fractal description of contours. In our algorithm we used fractal interpolation. Presented examples show that the contours are generated fractally in a progressive way. Moreover, because the contours are generated fractally they posses the property of resolution independence, i.e. they look the same independently of the scale in which they are drawn.

As we mentioned before we can use in fractal interpolation more points and different values for each of the free parameters. This will lengthen the time needed for finding the best IFS but the fitting to the original contour could be more accurate. Also it seems to be possible to extend papers result to 3D shapes using the results from [10] where fractal interpolation of 3D data is presented.

References

[1] M. Barnsley, *Fractals Everywhere*, Academic Press, Boston (1988).

[2] A. Bovik, *Handbook of Image and Video Processing*, Academic Press (2000).

[3] D. Chetverikov, Z. Szabó, *Detection of High Curvature Points in Planar Curves*, http://visual.ipan.sztaki.hu/corner/cornerweb.html.

[4] W.O. Cochran, J.C. Hart, P.J. Flynn, On approximating rough curves with fractal functions, In: *Proc. Graphics Interface'98* (1998), 65-72.

[5] W. Kotarski, A. Lisowska, On Bézier-Fractal modeling of 2D shapes, *International Journal of Pure and Applied Mathematics*, **24**, No. 1 (2005), 123-134.

[6] W. Kotarski, A. Lisowska, Chaikin's approach to fractal modeling of 2D contours, In: *Proceedings of Eurocon'05 Conference* (2005), 1192-1195.

[7] B. Mandelbrot, *The Fractal Geometry of Nature*, W.H. Freeman and Company, New York (1983).

[8] S. Nikiel, *Iterated Funtion Systems for Real-Time Image Synthesis*, Springer-Verlag, London (2007).

[9] S. Schaefer, D. Levin, R. Goldman, Subdivision schemes and attractors, *Eurographics Symposium on Geometry Processing* (2005), 171-180.

[10] C.M. Wittenbrink, IFS fractal interpolation for 2D and 3D visualization, In: *Proceedings of the 6-th IEEE Visualization Conference* (1995), 77-84.