Effective representation of 2D and 3D data using fractal interpolation

Polychronis Manousopoulos, Vassileios Drakopoulos, Theoharis Theoharis, Pavlos Stavrou
Department of Informatics and Telecommunications, Theoretical Informatics,
University of Athens, Panepistimioupolis, 157 84, Athens, Greece.
{polyman, vasilios, theotheo, p.stavrou}@di.uoa.gr

Abstract

Methods for representing curves in $\mathbb{R}^2$ and $\mathbb{R}^3$ using fractal interpolation techniques are presented. We show that such representations are both effective and convenient for irregular or complicated data. Experiments in various datasets, including geographical and medical data, verify the practical usefulness of these methods.

1. Introduction

In many practical applications, it is often necessary to model complicated or irregular data as, for example, coastlines and similar complicated geographical boundaries, or medical data such as boundaries of anatomical structures. Such data have an inherent fractal structure that can be cumbersome to model using traditional techniques based on smooth elementary functions, e.g. polynomials. Fractal interpolation offers a convenient and efficient approach to modelling intricate or irregular shapes.

In this paper, we examine three methods for modelling 2D or 3D curves using fractal interpolation techniques and apply them in a variety of areas, including geographical and medical ones. The results show that these methods are indeed useful in practice, achieving accurate representations along with satisfactory compression ratios. The paper is structured as follows. Section 2 contains a brief review of fractal interpolation functions. In Section 3, we present three methods for constructing fractal interpolation curves, while Section 4 describes how to model shapes using them. In Section 5, we apply the aforementioned methods to a variety of datasets, evaluate their performance and indicate areas of future work.

2. Fractal interpolation functions

Fractal interpolation functions as defined in [1] and [2] are based on the theory of *iterated function systems (IFSs)*. An IFS, denoted by $\{X; w_n, n = 1, 2, \ldots, N\}$, consists of a complete metric space (X, ρ), e.g. $(\mathbb{R}^n, ||\cdot||)$ or a subset of such a space, and a finite set of continuous mappings $w_n\colon X \to X$, $n = 1, 2, \ldots, N$. If w_n are contractions with respective *contractivity factors* s_n,$n = 1, 2, \ldots, N$, the IFS is termed *hyperbolic* and the transformation $W\colon \mathcal{H}(X) \to \mathcal{H}(X)$ with $W(B) = \cup_{n=1}^{N} w_n(B)$, where $\mathcal{H}(X)$ denotes the space of nonempty compact subsets of X, has a unique fixed point $A_\infty = W(A_\infty) = \lim_{n\to\infty} W^n(B)$, for every $B \in \mathcal{H}(X)$, which is called the *attractor* of the IFS.

2.1. Fractal interpolation functions on the plane

Let us represent the given set of *data points* as $\{(u_m, v_m) \in \mathbb{R}^2\colon\ m = 0, 1, \ldots, M\}$. In general, the interpolation is applied to a subset of them, the *interpolation points*, represented as $\{(x_i, y_i) \in \mathbb{R}^2\colon\ i = 0, 1, \ldots, N\}$. Both sets are linearly ordered with respect to their abscissa, i.e. $u_0 < u_1 < \cdots < u_M$ and $u_0 = x_0 < x_1 < \cdots < x_N = u_M$. The interpolation points partition the set of data points into *interpolation intervals* and may be chosen equidistantly or not.

Let $\{\mathbb{R}^2; w_n, n = 1, 2, \ldots, N\}$ be an IFS with affine transformations

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

constrained to satisfy

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

for every $n = 1, 2, \ldots, N$. Solving the above equations results in

$$\begin{aligned} a_n &= \frac{x_n - x_{n-1}}{x_N - x_0} \\ d_n &= \frac{x_N x_{n-1} - x_0 x_n}{x_N - x_0} \\ c_n &= \frac{y_n - y_{n-1}}{x_N - x_0} - s_n \frac{y_N - y_0}{x_N - x_0} \\ e_n &= \frac{x_N y_{n-1} - x_0 y_n}{x_N - x_0} - s_n \frac{x_N y_0 - x_0 y_N}{x_N - x_0}, \end{aligned}$$

i.e. the real numbers a_n, d_n, c_n, e_n are completely determined by the interpolation points, while the s_n are *free* parameters of the transformations satisfying $|s_n| < 1$, in order to guarantee that the IFS is hyperbolic with respect to an appropriate metric. The transformations w_n are *shear transformations*: line segments parallel to the y-axis are mapped to line segments parallel to the y-axis contracted by the factor $|s_n|$. For this reason, the s_n are called *vertical scaling* (or *contractivity*) *factors.*

It is well known (see for example [2]) that the attractor $G = \bigcup_{n=1}^{N} w_n(G)$ of the aforementioned IFS is the graph of a continuous function $f\colon [x_0, x_N] \to \mathbb{R}$ that interpolates the points (x_i, y_i), $i = 0, 1, \ldots, N$. This function is called *fractal interpolation function (FIF)* corresponding to these points. It is a *self-affine* function since each affine transformation w_n maps the entire (graph of the) function to its *section*, i.e. function values between the interpolation points (x_{n-1}, y_{n-1}) and (x_n, y_n) for all $n = 1, 2, \ldots, N$.

Although the FIF passes by definition through the interpolation points, this is not necessarily the case for the remaining data points $\{(u_m, v_m)\} \setminus \{(x_i, y_i)\}$. The accuracy of fit can be measured as the squared error between the ordinates of the original and the reconstructed points

$$\varepsilon = \sum_{m=0}^{M} (v_m - G(u_m))^2 \tag{1}$$

where $G(u_m)$ denotes the ordinate of the attractor for abscissa u_m, or alternatively, as the *Hausdorff distance* between the two sets

$$\varepsilon = h(\{(u_m, v_m)\}, G). \tag{2}$$

The vertical scaling factors of a FIF are usually chosen so as to minimize such an error measure. For example, in [8] and [10] the minimization of (1) is achieved by analytic or geometric methods. Moreover, in [10] a greedy algorithm for finding some proper (but not necessarily globally optimal) interpolation points is presented.

2.2. Generalized fractal interpolation functions

The FIF model described in the previous section can be extended to higher dimensions, producing functions that interpolate points in $\mathbb{R}^k$. Let $\{p_m \in \mathbb{R}^k\colon m = 0, 1, \ldots, M\}$ be the set of data points and $\{q_i \in \mathbb{R}^k\colon i = 0, 1, \ldots, N\}$ the set of interpolation points. Both sets are again assumed to be linearly ordered with respect to their abscissa. Let $\{\mathbb{R}^k; w_n, n = 1, 2, \ldots, N\}$ be an IFS with affine transformations

$$w_n \begin{bmatrix} q^1 \\ q^2 \\ \vdots \\ q^k \end{bmatrix} = \underbrace{\begin{bmatrix} a_n & 0 & \cdots & 0 \\ c_n^1 & s_n^{1,1} & \cdots & s_n^{1,k-1} \\ \vdots & \vdots & \ddots & \vdots \\ c_n^{k-1} & s_n^{k-1,1} & \cdots & s_n^{k-1,k-1} \end{bmatrix}}_{k \times k} \underbrace{\begin{bmatrix} q^1 \\ q^2 \\ \vdots \\ q^k \end{bmatrix}}_{k \times 1} + \begin{bmatrix} d_n^1 \\ d_n^2 \\ \vdots \\ d_n^k \end{bmatrix}$$

constrained to satisfy

$$w_n(q_0) = q_{n-1} \quad \text{and} \quad w_n(q_N) = q_n$$

for every $n = 1, 2, \ldots, N$. The real numbers a_n, c_n^i, d_n^j for $n = 1, \ldots, N$, $i = 1, 2, \ldots, k-1$ and $j = 1, 2, \ldots, k$ are completely determined by the interpolation points by solving the above equations, while the $s_n^{i,j}$, $i, j = 1, 2, \ldots, k-1$ are free parameters of the transformations chosen such that the contractivity factor s_n of the matrix (called *contractivity matrix*)

$$\begin{bmatrix} s_n^{1,1} & \cdots & s_n^{1,k-1} \\ \vdots & \ddots & \vdots \\ s_n^{k-1,1} & \cdots & s_n^{k-1,k-1} \end{bmatrix}$$

has modulus less than unity, in order to guarantee that the IFS is hyperbolic with respect to an appropriate metric. The exact values of the $s_n^{i,j}$ can be determined by minimizing an error measure as in the planar case (see e.g. [9]). In $\mathbb{R}^3$ we have

$$\begin{aligned} a_n &= \frac{x_n - x_{n-1}}{x_N - x_0} \\ d_n^1 &= \frac{x_N x_{n-1} - x_0 x_n}{x_N - x_0} \\ c_n^1 &= \frac{y_n - y_{n-1}}{x_N - x_0} - s_n^{1,1} \frac{y_N - y_0}{x_N - x_0} - s_n^{1,2} \frac{z_N - z_0}{x_N - x_0} \\ c_n^2 &= \frac{z_n - z_{n-1}}{x_N - x_0} - s_n^{2,1} \frac{z_N - z_0}{x_N - x_0} - s_n^{2,2} \frac{z_N - z_0}{x_N - x_0} \\ d_n^2 &= \frac{x_N y_{n-1} - x_0 y_n}{x_N - x_0} - s_n^{1,1} \frac{x_N y_0 - x_0 y_N}{x_N - x_0} \\ &\quad - s_n^{1,2} \frac{x_N z_0 - x_0 z_N}{x_N - x_0} \\ d_n^3 &= \frac{x_N z_{n-1} - x_0 z_n}{x_N - x_0} - s_n^{2,1} \frac{x_N y_0 - x_0 y_N}{x_N - x_0} \\ &\quad - s_n^{2,2} \frac{x_N z_0 - x_0 z_N}{x_N - x_0}. \end{aligned}$$

The attractor $G = \bigcup_{n=1}^{N} w_n(G)$ of the IFS is the graph of a continuous function $f\colon [q_0^1, q_N^1] \to \mathbb{R}^{k-1}$ that interpolates the points q_i, $i = 0, 1, \ldots, N$ (see [2]). It is a self-affine FIF in $\mathbb{R}^k$; however, its orthogonal projections on $\mathbb{R}^2$ are not necessarily self-affine.

3. Fractal interpolation curves

In the case where the interpolation points define a curve rather than a function, i.e. they are not linearly ordered with respect to their abscissa, the direct use of a fractal interpolation function is not possible. In order to construct an IFS whose attractor interpolates the given points and is therefore a curve, we can transform or extend the original points such that the application of a FIF is possible. This is then transformed or projected back to the plane to obtain a *fractal interpolation curve* that interpolates the original points. Here we will present two existing methods for constructing fractal interpolation curves ([4], [11]) and a new one we have introduced in a previous work ([7]). Alternative methods can be found in [3], [5] and [6]. A detailed review of methods for constructing fractal interpolation curves can be found in [7].

3.1. Related work

A straightforward way for constructing fractal interpolation curves involves the use of Generalized FIFs (see e.g. [4]). Let $\{(x_i, y_i) \in \mathbb{R}^2\colon\ i = 0, 1, \ldots, N\}$ be the set of interpolation points. These points do not define a function but a curve on the xy-plane, i.e. it is not necessarily $x_i < x_j$ for $i < j$. We apply the transformation $T\colon \mathbb{R}^2 \to \mathbb{R}^3$ with $(x_i, y_i) \mapsto (t_i, x_i, y_i)$, $i = 0, 1, \ldots, N$, where the introduced index coordinates t_i satisfy $t_0 < t_1 < \cdots < t_N$; usually we set $t_i = i$. The new set of points $(t_i, x_i, y_i), i = 0, 1, \ldots, N$ is the *generalized set* corresponding to the original points and defines a function. We create a FIF that interpolates the generalized set as described in Subsection 2.2. The attractor $G' = \bigcup_{n=1}^{N} w_n(G')$ of the IFS is the graph of a continuous function $f'\colon [t_0, t_N] \to \mathbb{R}^2$ that interpolates the generalized set of points (t_i, x_i, y_i), $i = 0, 1, \ldots, N$. The projection of the attractor to $\mathbb{R}^2$, defined by $P_t\colon G' \to G$ with $(t_i, x_i, y_i) \mapsto (x_i, y_i)$, is the graph of a continuous curve $f\colon [x_0, x_N] \to \mathbb{R}$ that interpolates the points (x_i, y_i), $i = 0, 1, \ldots, N$. Note that although the attactor G' is self-affine, its projection G is not necessarily self-affine. This method will be called Method 1 hereafter.

A similar approach involves the introduction of index coordinates without generalization to a higher-dimensional space ([11]). Specifically, we split the original set of points into two new sets by introducing an index for each coordinate. Then a fractal interpolation function is constructed for each new set and these are finally combined in a single curve that interpolates the original points.

As previously, let $\{(x_i, y_i) \in \mathbb{R}^2\colon\ i = 0, 1, \ldots, N\}$ be the set of interpolation points. We apply the transformations $T_1\colon \mathbb{R}^2 \to \mathbb{R}^2$ with $(x_i, y_i) \mapsto (t_i, x_i)$, $i = 0, 1, \ldots, N$ and $T_2\colon \mathbb{R}^2 \to \mathbb{R}^2$ with $(x_i, y_i) \mapsto (t_i, y_i)$, $i = 0, 1, \ldots, N$, where the introduced index coordinates t_i satisfy $t_0 < t_1 < \cdots < t_N$; usually we set $t_i = i$.

Then, we create a fractal interpolation function for each of the two sets in the way described in Subsection 2.1. Let $G^x = (t_i^x, x_i)$ and $G^y = (t_i^y, y_i)$ be the attractors of the respective IFS. We can merge G^x and G^y in order to obtain $G = (x_i, y_i)$ which is the graph of a continuous curve $f\colon [x_0, x_N] \to \mathbb{R}$ that passes through the interpolation points. Note that although the attactors G^x and G^y are self-affine, this is not necessarily the case for G. This method will be called Method 2 hereafter.

3.2. The proposed method

A new method for creating fractal interpolation curves on the plane without using index coordinates or generalizing to a higher-dimensional space was introduced in [7]. The *fractal curve fitting* (FCF) method involves applying a reversible transformation to the data points, such that they define a function on the plane. Then a FIF is constructed as usual and its attractor is transformed back to the original coordinates, in order to obtain a curve that interpolates the original points.

Let us represent the given set of data points as $\{(u_m, v_m) \in \mathbb{R}^2 : m = 0, 1, \ldots, M\}$ and the respective set of interpolation points as $\{(u_{\mathbb{J}(i)}, v_{\mathbb{J}(i)}) \in \mathbb{R}^2 : i = 0, 1, \ldots, N\}$, where the labelling function $\mathbb{J}\colon \{0, 1, \ldots, N\} \to \{0, 1, \ldots, M\}$ defines the indices of the interpolation points. We apply the transformation $T_1(u_m, v_m) = (u'_m, v'_m)$, $m = 0, 1, \ldots, M$, where

$$\begin{aligned} u'_m &= u_0 + \sum_{j=1}^{m} (|u_j - u_{j-1}| + \varepsilon) = \\ &= u'_{m-1} + (|u_m - u_{m-1}| + \varepsilon), \\ v'_m &= v_m, \end{aligned}$$

where $\varepsilon > 0$ is an arbitrary constant necessary, when all points in an interpolation interval have equal u-coordinates, i.e. $u_m = u_{m-1}$ for every $m = \mathbb{J}(i) + 1, \ldots, \mathbb{J}(i+1)$ and some $i \in \{0, 1, \ldots, N\}$. Otherwise, we set $\varepsilon = 0$. The resulting points (u'_m, v'_m), $i = 0, 1, \ldots, M$ are linearly ordered with respect to their abscissa, i.e. $u'_m < u'_n$ for every $m < n$. This transformation is essentially arraying the data points so as to preserve their horizontal distances.

The next step is to create an IFS whose attractor is the graph of a function that interpolates the points $(u'_{\mathbb{J}(i)}, v'_{\mathbb{J}(i)})$, $i = 0, 1, \ldots, N$. This is achieved by using a $2D$ affine IFS (Subsection 2.1) and the result is its attractor G'.

The final step is to apply a second transformation to G' in order to obtain the graph G of a curve that interpolates the initial points $\{(u_m, v_m)\colon m = 0, 1, \ldots, M\}$. Let $(u', v') \in G'$ be a point of the attractor. We apply the transformation $T_2\colon G' \to G$ with $(u', v') \mapsto (u, v)$, where

$$\begin{aligned} u &= u_{m-1} + (u_m - u_{m-1}) \left(\frac{u' - u'_{m-1}}{u'_m - u'_{m-1}} \right), \\ &\qquad u' \in [u'_{m-1}, u'_m] \\ v &= v' \end{aligned}$$

Note that the overlapping at the endpoints of successive intervals $[u'_{i-1}, u'_i]$ in the above formula is not ambiguous, since the resulting u is the same in both cases. The transformation T_2 can be efficiently computed, if the points of the attractor are first sorted by u' and then the attractor and transformed data points are swept in parallel in order to calculate the appropriate (u, v). Th proposed method will be called Method 3 hereafter.

The fractal interpolation curves constructed by the above three methods are open, assuming that the first and last interpolation points are different. In order to construct a closed fractal interpolation curve, we append to the original interpolation points an additional one that is the same as the first, i.e. we add $(x_{N+1}, y_{N+1}) = (x_0, y_0)$. The curve is afterwards constructed in the same way.

3.3. Three dimensional curves

The three methods of Subsections 3.1–3.2 can be directly extended to model curves in $\mathbb{R}^3$; the additional z coordinate is treated exactly as the y coordinate. In Method 1, the interpolation points $\{(x_i, y_i, z_i) \in \mathbb{R}^3\colon i = 0, 1, \ldots, N\}$ are transformed by $T\colon \mathbb{R}^3 \to \mathbb{R}^4$ with $(x_i, y_i, z_i) \mapsto (t_i, x_i, y_i, z_i)$, $i = 0, 1, \ldots, N$. A generalized FIF in $\mathbb{R}^4$ is subsequently constructed, whose attractor is finally projected on $\mathbb{R}^3$ by eliminating the t coordinate. In Method 2, a third FIF is constructed to interpolate the additional points (t_i, z_i), $i = 0, 1, \ldots, N$ and is subsequently treated similarly. In Method 3, the third coordinate of the data points is treated exactly like the v coordinate; in the intermediate step a 3D generalized FIF is constructed instead.

4. Shape modelling using fractal interpolation curves

In order to model a shape as a fractal interpolation curve, we first need to determine the interpolation points. This can be done at a predetermined step, i.e. by choosing every n-th data point as interpolation point, or using an automated algorithm for locating them such as the one in [10]. The number of interpolation points determines the balance between the accuracy of fit and the compression ratio and should therefore be accordingly set. The fractal interpolation curve can then be constructed by one of the three presented methods. In all cases, we must define an algorithm for calculating the vertical scaling factors such as those mentioned in Subsection 2.1. A shape can then be represented by the vector consisting of the coefficents of all affine transformations of the respective IFS(s). Note that, for the first two methods, it is possible to omit the a_n, d_n^1 and a_n, d_n coefficients respectively, if a consistent indexing scheme is predetermined (e.g. $t_i = i$); in this case a better compression ratio is achieved. Moreover, fractal interpolation curves provide an inherent ability for multi-resolution modelling; any part of the curve can be infinitely magnified revealing detail.

5. Results

We have applied the three described methods to a variety of 2D and 3D data. In all cases, the interpolation intervals have been chosen with fixed length and the contractivity factors of the constructed interpolants have been calculated with the analytic algorithms of [9] and [10]. The compression ratio of each representation is calculated as $r = c_1 N / (c_2 M)$, where N is the number of affine transformations, c_1 the number of affine transformation coefficients depending on the method and the data dimension, M is the number of data points and $c_2 = 2$ or 3 depending on the data dimension.

Figures 1–6 depict six coastlines of Greek islands and their representation as two dimensional curves using all three methods. The interpolation points have been chosen with a fixed step of 20, i.e. every 20th data point is used as interpolation point, resulting in compression ratios from 1:5.00 to 1:8.04. Even though the coastlines are very intricate, their representation is very accurate with considerable compression ratios, as summarized in the following table.

Figures 7–9, depict brain structures, namely the white matter, located in axial anatomical images (cryosections)[1]. Their boundaries are represented us-

[1] The axial anatomical images are from the Visible Human

Coastline	Num. of points	Method	Comp. Ratio
A	4410	1	1:5.01
B	7510	2	1:6.68
C	3898	3	1:8.04
D	7186	1	1:5.00
E	4664	2	1:6.67
F	6172	3	1:8.02

Table 1. The compression ratios achieved by the three methods for representing the coastlines.

ing multiple fractal interpolation curves, since they consist of disjoint parts. All three methods have been used, with the interpolation points chosen with a fixed step of 20 resulting in compression ratios from 1:5.02 to 1:8.03, as summarized in the following table.

Anatom. image	Num. of points	Method	C. Ratio
A	3000	1	1:5.02
B	4010	2	1:6.65
C	5218	3	1:8.03

Table 2. The compression ratios achieved by the three methods for representing the white matter of brain in axial anatomical images.

Figure 10(a) contains a three dimensional stone fragment. Its boundary in the current view is a three dimensional curve. This boundary is represented using the Method 1, and is depicted in Figures 10(b)–10(d). The interpolation points have been chosen with a fixed step of 20 resulting in a comp. ratio of 1:4.00. A similar example of a three dimensional pot fragment is depicted in Figure 11, where the current view boundary is represented with the Method 3. The interpolation points are chosen with a fixed step of 20 resulting in a compression ratio of 1:6.15.

Conclusively, all methods represent accurately the 2D and 3D shapes achieving considerable compression ratios. Even though the represented shapes are very intricate, the three methods are easily applied, yielding satisfactory results. Moreover, it is possible to define the balance between compression ratio and accuracy of representation by simply adjusting the interpolation points selection step. Therefore, the three described methods are both convenient and effective in practice. Future work will focus on using piecewise self-affine fractal interpolation as well as automated efficient methods for locating the optimal interpolation points.

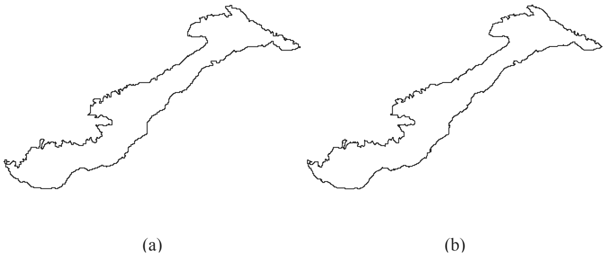


Figure 1. (a) Coastline A (Amorgos) consisting of 4410 points. (b) The reconstructed curve using the Method 1 with comp. ratio = 1:5.01.

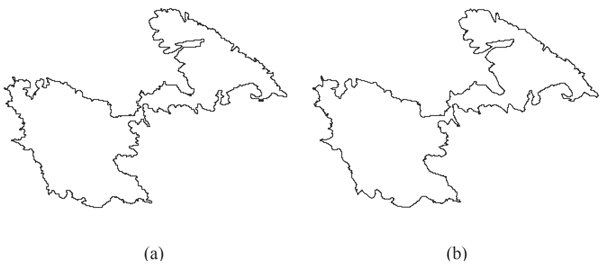


Figure 2. (a) Coastline B (Astypalaia) consisting of 7510 points. (b) The reconstructed curve using the Method 2 with comp. ratio = 1:6.68 .

Acknowledgements

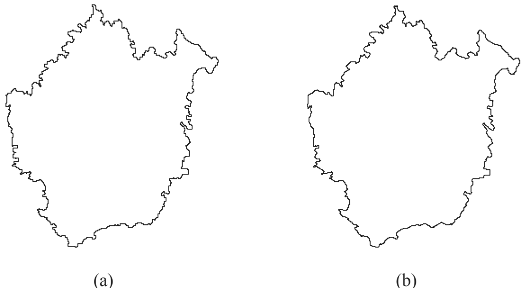
This research project is co-financed by E.U.-European Social Fund (75%) and the Greek Ministry of Development-GSRT (25%), under the PENED programme 70/3/8405. The authors thank Drs. Costas Spengos and Theodoros Papaioannou for their assistance on the interpretation of the brain axial anatomical images.

References

[1] M. F. Barnsley. Fractal functions and interpolation. *Constr. Approx.*, 2:303–329, 1986.

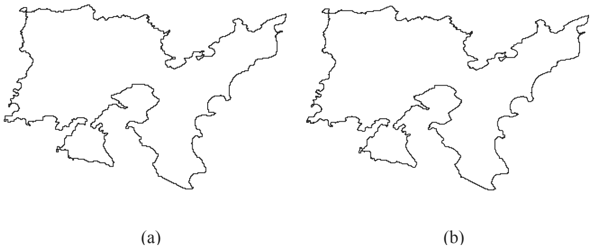
[2] M. F. Barnsley. *Fractals everywhere.* Academic Press Professional, San Diego, 2nd edition, 1993.

Project of the National Library of Medicine which is hereby acknowledged.



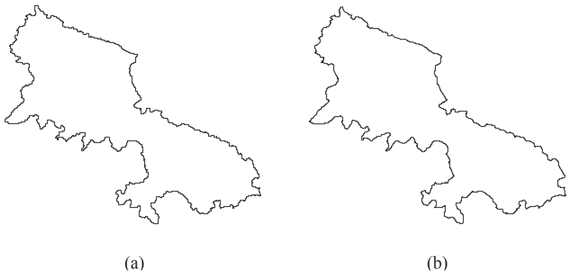
(a) (b)

Figure 3. (a) Coastline C (Kimolos) consisting of 3898 points. (b) The reconstructed curve using the Method 3 with comp. ratio = 1:8.04 .



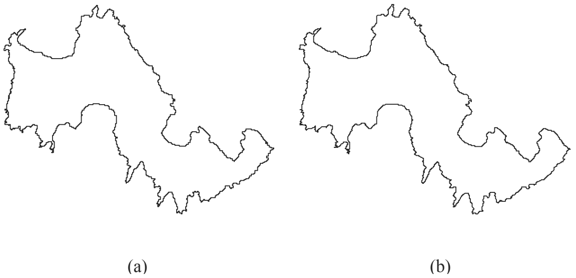
(a) (b)

Figure 4. (a) Coastline D (Lemnos) consisting of 7186 points. (b) The reconstructed curve using the Method 1 with comp. ratio = 1:5.00 .



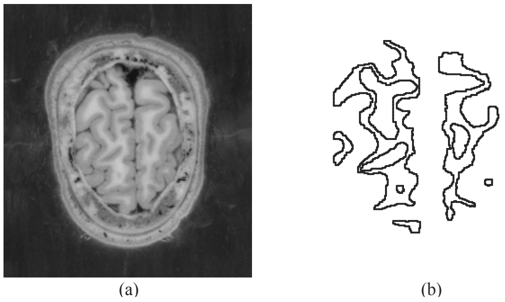
(a) (b)

Figure 5. (a) Coastline E (Skyros) consisting of 4664 points. (b) The reconstructed curve using the Method 2 with comp. ratio = 1:6.67 .



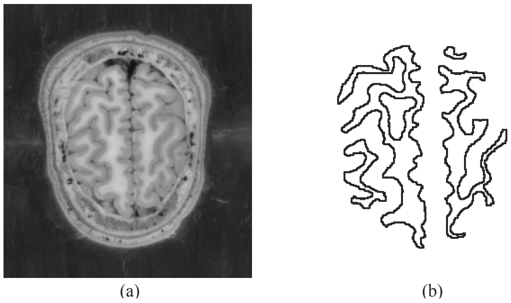
(a) (b)

Figure 6. (a) Coastline F (Tilos) consisting of 6172 points. (b) The reconstructed curve using the Method 3 with comp. ratio = 1:8.02 .



(a) (b)

Figure 7. (a) Axial anatomical image A. (b) The reconstructed curve using the Method 1 with comp. ratio = 1:5.02 (magnified).



(a) (b)

Figure 8. (a) Axial anatomical image B. (b) The reconstructed curve using the Method 2 with comp. ratio = 1:6.65 (magnified).

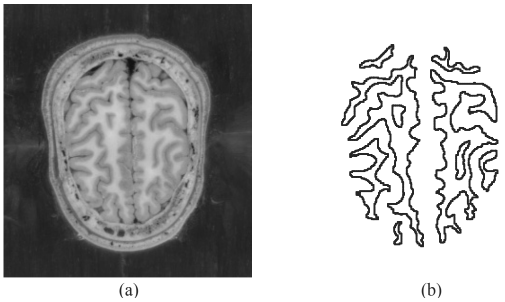


Figure 9. (a) Axial anatomical image C. (b) The reconstructed curve using the Method 3 with comp. ratio = 1:8.03 (magnified).

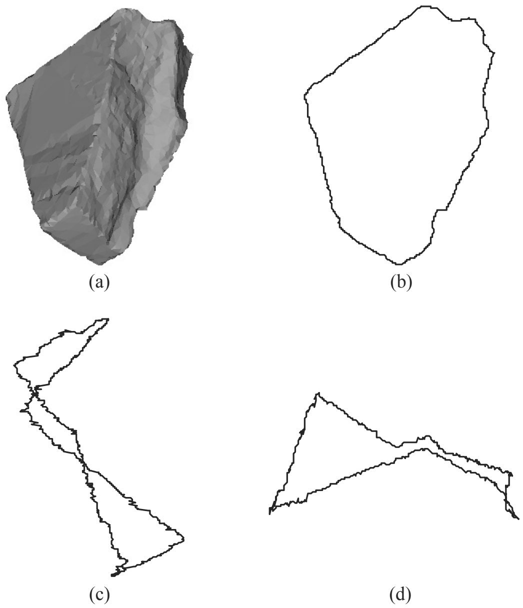


Figure 10. (a) A stone fragment. (b) The reconstructed 3D boundary curve – xy projection. (c) The reconstructed 3D boundary curve – yz projection. (d) The reconstructed 3D boundary curve - xz projection.

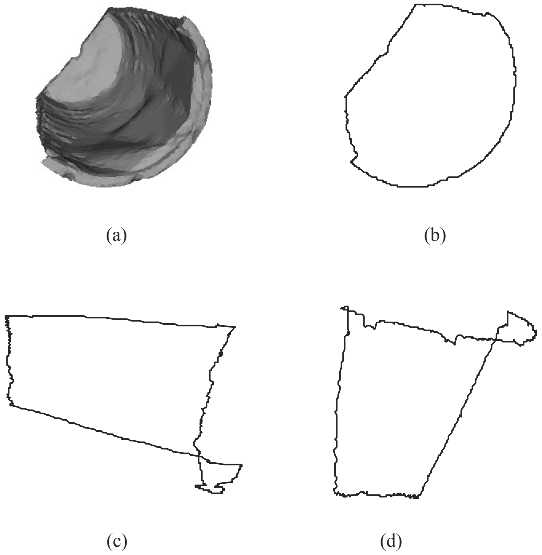


Figure 11. (a) A pot fragment. (b) The reconstructed 3D boundary curve – xy projection. (c) The reconstructed 3D boundary curve – yz projection. (d) The reconstructed 3D boundary curve – xz projection.

[3] A. Cader and M. Krupski. New interpolation method with fractal curves. In Leszek Rutkowski, Ryszard Tadeusiewicz, Lotfi A. Zadeh, and Jacek Zurada, editors, *ICAISC*, volume 4029 of *Lecture Notes in Computer Science*, pages 1071–1081, Berlin and Heidelberg, 2006. Springer-Verlag.

[4] W. O. Cochran, J. C. Hart, and P. J. Flynn. On approximating rough curves with fractal functions. In *Proc. Graphics Interface*, volume 1, pages 65–72, 1998.

[5] E. Guérin, E. Tosan, and A. Baskurt. Fractal coding of shapes based on a projected IFS model. In *ICIP*, volume 2, pages 203–206. IEEE Computer Society, 2000.

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

[7] P. Manousopoulos, V. Drakopoulos, and T. Theoharis. Curve fitting by fractal interpolation. *To appear in Transactions on Computational Science.*

[8] M.A. Marvasti and W.C. Strahle. Fractal geometry analysis of turbulent data. *Signal Processing*, 41:191–201, 1995.

[9] D. S. Mazel. Representation of discrete sequences with three-dimensional iterated function systems. *IEEE Trans. Signal Processing*, 42:3269–3271, 1994.

[10] D. S. Mazel and M. H. Hayes. Using iterated function systems to model discrete sequences. *IEEE Trans. Signal Processing*, 40:1724–1734, 1992.

[11] S. Uemura, M. Haseyama, and H. Kitajima. Efficient contour shape description by using fractal interpolation functions. In *IEEE Proc. ICIP*, volume 1, pages 485–488, 2002.