

Parameter identification of 1D recurrent fractal interpolation functions with applications to imaging and signal processing

Polychronis Manousopoulos, Vassileios Drakopoulos,
Theoharis Theoharis

Department of Informatics and Telecommunications, Theoretical Informatics,
University of Athens, Panepistimioupolis, 157 84, Athens, Greece.
{polyman, vasilios, theotheo}@di.uoa.gr

Abstract

Recurrent fractal interpolation functions are very useful in modelling irregular (non-smooth) data. Two methods that use bounding volumes and one that uses the concept of box-counting dimension are introduced for the identification of the vertical scaling factors of such functions. The first two minimize the area of the symmetric difference between the bounding volumes of the data points and their transformed images, while the latter aims at achieving the same box-counting dimension between the original and the reconstructed data. Comparative results with existing methods in imaging applications are given, indicating that the proposed ones are competitive alternatives for both low and high compression ratios.

Keywords: fractal interpolation, recurrent iterated function system, vertical scaling factor, symmetric difference metric, Hausdorff metric, box-counting dimension.

1 Introduction

Affine fractal interpolation as defined in [1] and [2] (but see also [11]) offers an alternative to traditional interpolation techniques aiming mainly at data that exhibit an irregular, non-smooth structure which can not be conveniently described using functions such as polynomials. Examples of such data include projections of physical objects such as coastlines or plants as well as experimental data that have non-integral dimension. The affine fractal interpolation which is based on the theory of *iterated function systems* provides a procedural way to reconstruct real data; in other words, we do not store any direct information about the data but rather the parameterisation of a procedure used to reconstruct them. Recurrent fractal interpolation (see [3]) is essentially a generalisation of fractal interpolation, providing a more efficient way to reconstruct data that present self-affinity in a piecewise form and not in their entirety.

The closeness of fit of a *recurrent fractal interpolation function* is mainly influenced by the determination of its vertical scaling factors. No direct way to

find the optimal values of these factors exists but various approaches have been proposed. The most popular one [12] employs analytic (algebraic) or geometric methods. The algebraic approach derives an analytic expression for the vertical scaling factors by minimizing the sum of squared vertical distances between the original and the reconstructed points. According to the geometric approach, the factors are obtained by calculating ratios of vertical distances between the *data points* and the straight lines connecting the endpoints of the *address*, or the *interpolation intervals*. In the case of an *affine fractal interpolation function*, there is a wider variety of algorithms for determining the vertical scaling factors. The most popular are probably the affine versions of the aforementioned geometric and algebraic algorithms, which are also proposed in [12]. An alternative approach using the concept of fractal dimension is suggested in [18]. The use of wavelets is employed in various works, such as [4] and [7].

Our aim is, firstly, to create an alternative methodology for determining the vertical scaling factors of a recurrent fractal interpolation function by using bounding volumes of appropriately chosen data points such that the resulting fractal function provides a closer fit, with respect to some metric, to the original data points. This approach has been successfully applied to affine fractal interpolation functions in our previous work [9]. In this paper, we extend it to recurrent fractal interpolation and present two such methods, an analytic and an algorithmic one. Moreover, we present a new method for calculating the vertical scaling factors of a recurrent fractal interpolation function such that the resulting function has the same *box-counting dimension* with the original data. Finally, we examine the possibility of using wavelets for calculating the vertical scaling factors of a recurrent fractal interpolation function, following the rationale of the existing literature for the affine case.

As far as the evaluation of the proposed methods and their comparison with existing ones is concerned, we examine their applicability to imaging and signal processing. The test samples used in our experiments include geographic data, often occurring in real-world applications. Specifically, we examine the representation and compression of geographic region boundaries, such as coastlines.

The paper is organised as follows. In Section 2 we briefly review recurrent fractal interpolation functions and in Section 3 we describe our methods for computing the vertical scaling factors. Section 4 contains the experimental results of our methods as applied to geographic data as well as a comparison to existing methods. Section 5 summarises our conclusions and points out areas of future work.

2 Recurrent fractal interpolation functions

Let Δ_1, Δ_2 be two partitions of the real compact interval $I = [a, b]$, i.e. $\Delta_1 = \{u_0, u_1, \dots, u_M\}$ satisfying $a = u_0 < u_1 < \dots < u_M = b$ and $\Delta_2 = \{x_0, x_1, \dots, x_N\}$ satisfying $u_0 = x_0 < x_1 < \dots < x_N = u_M$, such that Δ_1 is a *refinement* of Δ_2 . Let us represent as $P = \{(u_m, v_m) \in I \times \mathbb{R} : m = 0, 1, \dots, M\}$ the given set of *data points* and as $Q = \{(x_i, y_i) \in I \times \mathbb{R} : i = 0, 1, \dots, N \leq M\}$ a subset of them, the *interpolation points*. The subintervals of Δ_2 are known as *interpolation intervals* and may be chosen equidistantly or not. The data points within the n th interpolation interval are represented as $P_n = \{(u_i, v_i) : i \in M_n\}$, $n = 1, 2, \dots, N$, where M_n is an index set of P_n such

that $\bigcup_{n=1}^N M_n = \{0, 1, \dots, M\}$ and $P = \bigcup_{n=1}^N P_n$. Each interpolation interval is associated with a pair of data points called *address points*. Specifically, the interpolation interval $I_n = [x_{n-1}, x_n]$ is associated with the points $(x'_{n,1}, y'_{n,1})$ and $(x'_{n,2}, y'_{n,2})$ for $n = 1, 2, \dots, N$, where $(x'_{n,k}, y'_{n,k}) = (u_m, v_m)$ for all $k \in \{1, 2\}$ and some $m = 0, 1, \dots, M$. Each pair of address points defines the *address interval* $[x'_{n,1}, x'_{n,2}]$ and it is $x'_{n,1} < x'_{n,2}$ for every $n = 1, 2, \dots, N$ by definition. Note that the address points need not necessarily be distinct and that each address interval has strictly greater length than its corresponding interpolation interval.

Let w_n , $n = 1, 2, \dots, N$ be affine transformations defined as

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

and constrained to satisfy

$$w_n \begin{bmatrix} x'_{n,1} \\ y'_{n,1} \end{bmatrix} = \begin{bmatrix} x_{n-1} \\ y_{n-1} \end{bmatrix} \quad \text{and} \quad w_n \begin{bmatrix} x'_{n,2} \\ y'_{n,2} \end{bmatrix} = \begin{bmatrix} x_n \\ y_n \end{bmatrix}$$

for every $n = 1, 2, \dots, N$, i.e. each address interval is mapped to its corresponding interpolation interval. Solving the above equations results in

$$\begin{aligned} a_n &= \frac{x_n - x_{n-1}}{x'_{n,2} - x'_{n,1}} \\ d_n &= \frac{x'_{n,2}x_{n-1} - x'_{n,1}x_n}{x'_{n,2} - x'_{n,1}} \\ c_n &= \frac{y_n - y_{n-1}}{x'_{n,2} - x'_{n,1}} - s_n \frac{y'_{n,2} - y'_{n,1}}{x'_{n,2} - x'_{n,1}} \\ e_n &= \frac{x'_{n,2}y_{n-1} - x'_{n,1}y_n}{x'_{n,2} - x'_{n,1}} - s_n \frac{x'_{n,2}y'_{n,1} - x'_{n,1}y'_{n,2}}{x'_{n,2} - x'_{n,1}} \end{aligned}$$

for every $n = 1, 2, \dots, N$, i.e. the real numbers a_n, d_n, c_n, e_n are completely determined by the interpolation and address points, while the s_n are *free* parameters of the transformations satisfying $|s_n| < 1$, so that the transformations w_n are contractive 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 s_n are called *vertical scaling*, or *contractivity*, *factors*.

Let $\mathcal{H}(\mathbb{R}^2)$ be the metric space of all non-empty, compact subsets of \mathbb{R}^2 with respect to the *Hausdorff metric*

$$h(A, B) = \max\left\{\max_{a \in A} \min_{b \in B} \|a - b\|, \max_{b \in B} \min_{a \in A} \|a - b\|\right\}, \quad A, B \in \mathcal{H}(\mathbb{R}^2).$$

Moreover, let $W(A) = \bigcup_{n=1}^N w_n(A^{[n]})$, where $A \in \mathcal{H}(\mathbb{R}^2)$ and $A^{[n]} = \{(x, y) \in A : x'_{n,1} \leq x \leq x'_{n,2}\}$, for $n = 1, 2, \dots, N$. The unique set $G \equiv A_\infty = \lim_{k \rightarrow \infty} W^k(A_0)$, for every starting set $A_0 \in \mathcal{H}(\mathbb{R}^2)$, is the graph of a continuous function $f: [x_0, x_N] \rightarrow \mathbb{R}$ that passes through the interpolation points (x_i, y_i) , for all $i = 0, 1, \dots, N$ (see [3], [12]). This function is called *recurrent fractal interpolation function*, or *RFIF* for short, corresponding to these points. A *section* is defined as the function values between interpolation points. A RFIF is a

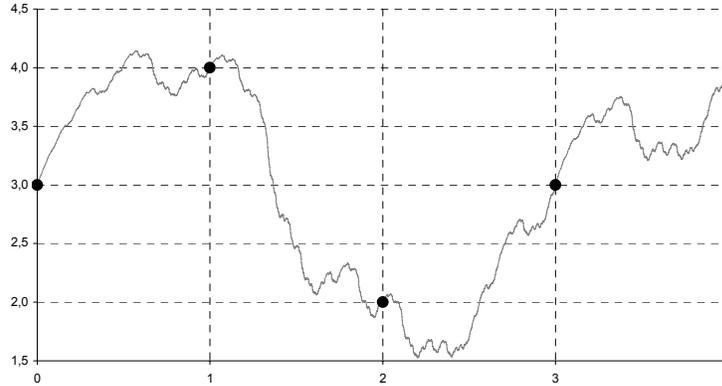


Figure 1: A recurrent fractal interpolation function (gray) for a set of five interpolation points (black).

piecewise self-affine function since each affine transformation w_n maps the part of the (graph of the) function defined by the corresponding address interval to each section. An example is depicted in Figure 1, where a RFIF is constructed for the set of interpolation points $Q = \{(0, 3), (1, 4), (2, 2), (3, 3), (4, 4)\}$ and respective address intervals $[0, 2], [0, 4], [1, 4], [0, 3]$ using vertical scaling factors $s_n = 0.35$, for every $n = 1, 2, 3, 4$.

Let $A \in \mathcal{H}(\mathbb{R}^d)$ and $\mathcal{N}(A, \varepsilon)$ denote the smallest number of d -dimensional boxes of side ε needed to cover A . If

$$D = \lim_{\varepsilon \rightarrow 0} \frac{\ln(\mathcal{N}(A, \varepsilon))}{\ln(1/\varepsilon)}$$

exists, then D is called the *box-counting dimension* of A .

In case of address intervals defined by interpolation points, i.e. $(x'_{n,k}, y'_{n,k}) \in Q$ for every $n = 1, 2, \dots, N$ and $k \in \{1, 2\}$, the *connection matrix* of an RFIF is defined as $C = (c_{ij})$, where $c_{ij} = 1$ if $[x_{j-1}, x_j] \subset [x'_{i,1}, x'_{i,2}]$ and $c_{ij} = 0$ otherwise, for every $i, j = 1, 2, \dots, N$. Moreover, let $S(d) = \text{diag}\{|s_1| |a_1|^{d-1}, \dots, |s_N| |a_N|^{d-1}\}$ and D be the unique value such that $\rho(CS(D)) = 1$, where $\rho(\cdot)$ denotes the *spectral radius* of a matrix. If $\rho(CS(1)) > 1$ and there exists an address interval with noncollinear interpolation points, then the *box-counting dimension* of the RFIF is $\dim_B(G) = D$ (see [3]).

3 Identifying the vertical scaling factors

3.1 Problem formulation

Although a RFIF passes by definition through its interpolation points, this is not necessarily the case for the remaining data points $P \setminus Q$. The closeness of fit depends solely on each vertical scaling factor s_n , $n = 1, 2, \dots, N$, the only free parameters for a given P , and can be measured as the squared error between the ordinates of the original and the reconstructed points $\sum_{m=0}^M (v_m - G[u_m])^2$, where $G[u_m]$ denotes the ordinate of the RFIF point with abscissa u_m or as the

Hausdorff distance $h(P, G)$. Because of the sensitivity of the Hausdorff metric to noise or to isolated points that stems from its ‘worst-case’ nature, a useful alternative can be the *Modified Hausdorff Distance*, or *MHD* for short (see [19]),

$$h_{MHD}(A, B) = \max \left\{ \frac{1}{|A|} \sum_{a \in A} \min_{b \in B} \|a - b\|, \frac{1}{|B|} \sum_{b \in B} \min_{a \in A} \|a - b\| \right\}$$

for $A, B \in \mathcal{H}(\mathbb{R}^2)$ and $|A|$ denoting the cardinality of the set A . The squared error measure is employed by existing methods (see e.g. [12]) in order to calculate the vertical scaling factors and evaluate their accuracy. The two others are adopted in this paper because the Hausdorff distance is considered more appropriate in the case of RFIF’s as defined previously, since it resides in the core of their definition.

Each affine transformation w_n transforms the set $[x'_{n,1}, x'_{n,2}] \times \mathbb{R} \supset P^{[n]} = \{(u, v) \in P : x'_{n,1} \leq u \leq x'_{n,2}\}$ into the set $[x_{n-1}, x_n] \times \mathbb{R} \supset P_n$, $n = 1, 2, \dots, N$. Although

$$h \left(P, \bigcup_{n=1}^N w_n(P^{[n]}) \right) = h \left(\bigcup_{n=1}^N P_n, \bigcup_{n=1}^N w_n(P^{[n]}) \right) \leq \max_{1 \leq n \leq N} \{h(P_n, w_n(P^{[n]}))\},$$

the direct evaluation of the optimal s_n that minimize $h(P_n, w_n(P^{[n]}))$ is not feasible. Therefore, their calculation should be achieved differently.

3.2 Parameter identification using bounding volumes

Following the approach of [9], we propose to work with *bounding volumes* of $P^{[n]}$ and P_n in order for the transformed points $w_n(P^{[n]})$ to best approximate the data points within P_n . Let $B^{[n]} \in \mathcal{K}_0^2$ be a bounding volume of $P^{[n]}$, where \mathcal{K}_0^2 denotes the set of convex, compact subsets of \mathbb{R}^2 with non-empty interior, and $B_n \in \mathcal{K}_0^2$ be convex bounding volumes of P_n for every $n = 1, 2, \dots, N$. In other words, it is $P^{[n]} \subset B^{[n]}$ and $P_n \subset B_n$, for every $n = 1, 2, \dots, N$. We use the *symmetric difference metric*

$$\delta^S(K, L) = \mathcal{H}^2(K \triangle L) = \mathcal{H}^2((K \setminus L) \cup (L \setminus K)), \quad K, L \in \mathcal{K}_0^2 \quad (1)$$

where \mathcal{H}^2 denotes the Hausdorff measure in \mathbb{R}^2 , in order to minimize the area of the symmetric difference $B_n \triangle w_n(B^{[n]})$, $n = 1, 2, \dots, N$. Notice that since we are constrained in \mathcal{K}_0^2 the Hausdorff measure coincides with the Lebesgue measure, i.e. the area, in \mathbb{R}^2 . So, Eq. 1 can be written in the form

$$\delta^S(K, L) = \text{area}(K \setminus L) + \text{area}(L \setminus K) = \text{area}(K \cup L) - \text{area}(K \cap L). \quad (2)$$

Therefore, by selecting the values of s_n that result in the maximum overlap of the respective bounding volumes we are able to produce a better approximation of the data points. As explained in [9], this approach has the advantage that, for suitably chosen bounding volumes $B^{[n]}$ and B_n , we are able to efficiently obtain the optimal s_n using either analytic expressions or efficient algorithms. Two types of bounding volume for the minimization of $\delta^S(B_n, w_n(B^{[n]}))$, for all $n = 1, 2, \dots, N$ are selected, namely the *bounding rectangle* and the *convex hull*. The first type allows the calculation of the optimal vertical scaling factors

using analytic expressions, while the second provides tighter bounds and efficient algorithmic calculations.

The first method (*Minimum Bounding Rectangle Method* or *MBRM* for short) employs bounding rectangles aligned with the axes of the co-ordinate system. Let R_n be the MBR of P_n and $R^{[n]}$ the MBR of $P^{[n]}$. In view of Eq. 1, our aim is the minimization of $\delta^S(R_n, w_n(R^{[n]}))$, for every $n = 1, 2, \dots, N$. This is achieved by minimizing the area of the non-overlapping parts of R_n and $w_n(R^{[n]})$. The possible cases of intersection of R_n and $w_n(R^{[n]})$ along with the analytic expressions for the optimal s_n that minimize the area of the non-overlapping parts in each case are presented in detail in [9].

The second method (*Convex Hull Method* or *CHM* for short) employs the convex hull as bounding volume. This provides a tighter bound than the rectangle and is actually the smallest convex volume containing the data points. Similarly to the case of bounding rectangles, we want to minimize the area of the non-overlapping parts of the convex hull of the points in the n -th interpolation interval and the transformation of the convex hull of $P^{[n]}$ under w_n . According to Eq. 2, this is

$$\delta^S(CH(P_n), CH(w_n(P^{[n]}))) = \text{Area}\{CH(P_n)\} + \text{Area}\{CH(w_n(P^{[n]}))\} - 2\text{Area}\{CH(P_n \cap w_n(P^{[n]}))\}, \quad (3)$$

where $CH(\cdot)$ is the convex hull of a set of points. The calculation of the optimal s_n cannot be performed analytically as in the MBRM. As implied by Eq. 3, the calculation of δ^S is algorithmic and involves the computation of convex hulls, polygon intersections and areas. As suggested in [9], a method for one-dimensional minimization without derivatives should be used, such as Brent's method which is a *bracketing method* with parabolic interpolation. The detailed parameterisation of this method can be found in the afore-mentioned work, where it is also shown that the method's time complexity is linear to the number of data points.

3.3 Parameter identification using the box-counting dimension

An alternative approach is based on the concept of box-counting dimension. This is suggested in [18], where the vertical scaling factors of an affine FIF are determined by first calculating the sum of their absolute values. This is achieved by using the equation

$$\sum_{n=1}^N |s_n| a_n^{D-1} = 1,$$

where a_n , s_n are the respective affine transformation coefficients and D is the box-counting dimension of the affine FIF. Assuming equidistant interpolation points, we have that

$$\sum_{n=1}^N |s_n| = N^{D-1}.$$

The value of D used in this equation is the box-counting dimension of the data points, thus guaranteeing that both data points and FIF have the same box-counting dimension. Each vertical scaling factor is then calculated as $|s_n| =$

$weight(n) \cdot \sum_{n=1}^N |s_n|$, where the weight is calculated as the proportion of the n -th interpolation interval's contribution to the complexity of the whole data, using the box-counting dimension as a measure of complexity. This is based on the observation that larger values of the vertical scaling factors result into more complicated FIFs with higher box-counting dimension. Thus, the resulting FIF has the same box-counting dimension as the original data.

We are extending this idea to RFIFs. As mentioned in Section 2, in the case when the endpoints of the address intervals are interpolation points, the box-counting dimension D of a RFIF satisfies the equation $\rho(CS(D)) = 1$ under certain conditions. Here, we examine the case when additionally for every $n = 1, 2, \dots, N$ we have that (a) the vertical scaling factors are the same, i.e. $s_n = s$, (b) the address intervals are of equal length, i.e. $x'_{n,2} - x'_{n,1} = L'$ and $\sum_{j=1}^N c_{ij} = c$, for every $i = 1, 2, \dots, N$, (c) the interpolation intervals are of equal length, i.e. $x_n - x_{n-1} = L$, so $a_n = L/L' = a$.

We have that

$$CS(D) = (c_{ij}|s_j||a_j|^{D-1}) = (c_{ij}|s||a|^{D-1}),$$

for $i, j = 1, 2, \dots, N$ and

$$r_i \equiv \sum_{j=1}^N c_{ij}|s||a|^{D-1} = c|s||a|^{D-1} \equiv r.$$

The matrix $CS(D)$ is non-negative and in the context of the Perron-Frobenius theorem (see [6]) it is

$$\min_{i=1,2,\dots,N} \{r_i\} \leq \rho(CS(D)) \leq \max_{i=1,2,\dots,N} \{r_i\}$$

Therefore, we have that

$$r = \rho(CS(D)) = 1 \Rightarrow c|s||a|^{D-1} = 1 \Rightarrow |s| = \frac{1}{c|a|^{D-1}}.$$

The box-counting dimension D in the above equation is calculated for the data points P (see e.g. [17] for an efficient algorithm). The sign of the vertical scaling factor is determined by selecting the one that minimizes the Hausdorff distance between the original and the reconstructed points. Henceforth, we will call this method the *Box-Counting Dimension Method*, or *BCDM* for short.

3.4 Parameter identification using wavelets

A hybrid algorithm using wavelets for the parameter identification of affine FIFs which is motivated by earlier works is presented in [4]; see also the references therein. The paper focuses on data sets that are known a priori to be attractors of IFSs that define affine FIFs. The existing data's self-affinity results into specific structure of their continuous wavelet transform (CWT), thus allowing the detection of specific points of the CWT, the so-called "toppoints", that are connected to the affine transformation coefficients. In the examples examined in [4], the proposed hybrid algorithm yields better results than earlier methods.

In this paper, we focus on the general case of arbitrary data sets and not only on attractors of interpolating IFSs or RFIFs which are rather unlikely to occur

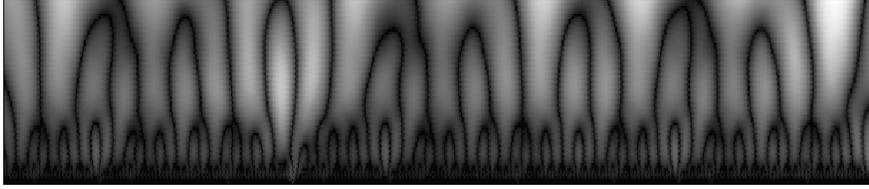


Figure 2: The modulus of the CWT of an affine FIF plotted against the scale (ordinate) and translation (abscissa) parameters.

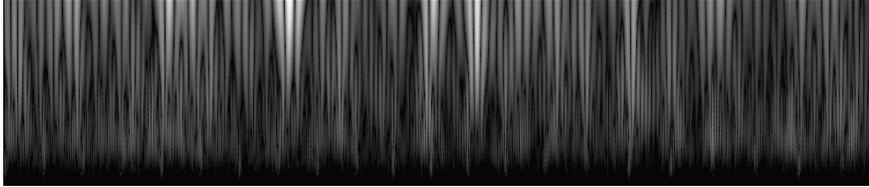


Figure 3: The modulus of the CWT of an EEG signal plotted against the scale (ordinate) and translation (abscissa) parameters.

in practical applications. In the general case, the use of wavelets is questionable; see e.g. [7] where the application of the CWT using a similar rationale to non self-affine data has not yielded satisfactory results even though the self-affine case was successful. Another example is given in Figures 2 and 3. In the first figure, the modulus of the CWT¹ of the FIF defined by the points $(0, 0)$, $(\frac{1}{3}, 1)$, $(\frac{2}{3}, 2)$ and $(1, 0.4)$ is depicted. The axes of the plot correspond to the scale and translation parameters of the CWT, while brighter areas indicate greater modulus. The structure and the corresponding toppoints described in [4] that stem from the self-affinity of the data are evident. In the latter figure, the CWT of an EEG signal is depicted. In this case, the lack of self-affinity results in a different structure of the CWT that renders the location and usefulness of such points for the parameter identification questionable. We conclude that the use of the CWT, as proposed in the aforementioned literature, is not expected to be fruitful for every arbitrary data set and thus will not be further examined in this paper.

4 Applications

In this section, we examine the application of the previously described methods to geographic data. The use of fractal interpolation in geographic applications, such as GIS, has been suggested in various works, since geographic data often present an intrinsic fractal structure. For example, coastline representation using curves constructed by affine fractal interpolation functions is examined in [8], [10] and [18].

¹The CWT in both figures has been calculated with biorthogonal spline wavelets (see e.g. [5]); specifically, “bior5.5” from the “bior” wavelet family of Matlab has been used.

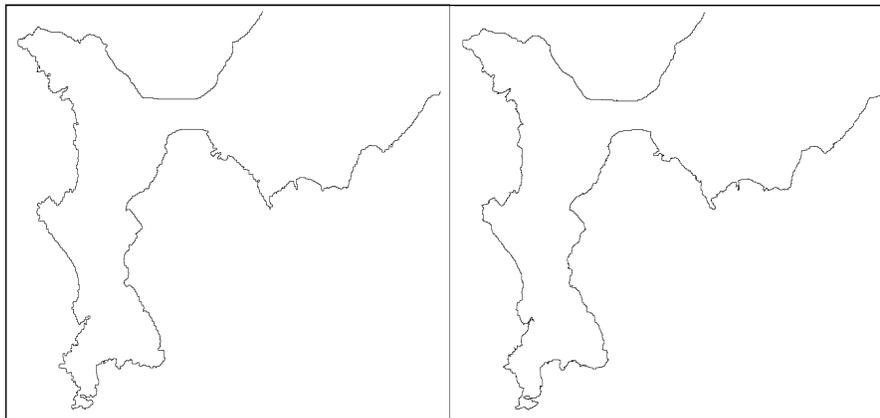


Figure 4: The original coastline (left) as well as its reconstruction by the CHM (right).

The left part of Figure 4 contains a coastline sample (from the Greek island of Lemnos) consisting of 5070 points. This has been extracted from a satellite image of the region using standard image processing techniques. The right part of the same figure contains the coastline reconstruction using the CHM for interpolation intervals of 40 points, i.e. every 40th point is chosen as interpolation point, and address intervals of 507 points. Following the approach of [12], the address intervals have been chosen to be consecutive and non-overlapping. The optimal address interval for each interpolation interval has been chosen by testing all address intervals and selecting the one that results in the lowest Hausdorff distance between the original and the reconstructed points. Note that an address interval may correspond to more than one interpolation interval or may not correspond to any at all. In case where an address interval results in a vertical scaling factor of modulus greater than or equal to unity, the pre-determined value of ± 0.99 is used instead. This approach is adopted in order to handle the case when, for a given interpolation interval, there is no address interval with vertical scaling factor of modulus less than unity.

We note that in this example the data form a curve rather than a function, i.e. they are not linearly ordered with respect to their abscissa. In order to represent them with a RFIF, we apply the methodology of [8] and [10] where the affine case is considered. As can be seen from the figure, the reconstruction of the coastline is successful despite the considerable sparsity of the interpolation points. This is more clear in Figure 5 that depicts a zoom in a subinterval of the previous figure; the black curve represents the original coastline while the gray curve represents its reconstruction. As shown in the figure, the reconstructed curve approximates faithfully the details of the original one even in this subinterval which corresponds to the most complicated part of the coastline; in less complicated intervals the approximation is even better. More specifically, for 4577 out of the 5070 points of the original curve, i.e. 90.28% of them, the corresponding point of the reconstructed curve is exactly the same or lies within a 1-pixel neighbourhood.

Table 1 contains the Hausdorff distance whereas Table 2 contains the Mod-

L	Method				
	Geometric	Algebraic	MBRM	CHM	BCDM
10	1.8894	1.6985	1.9185	1.5928	2.0149
20	2.9587	2.9320	2.9447	2.7072	3.1331
30	4.1633	3.9420	3.9222	3.8459	4.9534
40	4.9413	4.8098	4.7483	4.6214	6.1051
50	5.9980	5.7727	5.8813	5.3423	6.7804
60	6.2397	6.1179	6.0951	5.7334	7.5651
70	6.9534	7.0796	7.0105	6.5097	7.9214
80	7.9844	8.0110	7.8182	7.5527	9.0486
90	8.8117	9.1088	8.9053	8.3439	10.8379
100	10.6510	10.0125	10.0359	9.1993	12.5079

Table 1: The Hausdorff distance between the original and the reconstructed data using the five methods for various interpolation interval lengths.

L	Method				
	Geometric	Algebraic	MBRM	CHM	BCDM
10	0.3961	0.3949	0.4025	0.3902	0.4263
20	0.5699	0.5645	0.5660	0.5541	0.6327
30	0.7350	0.7232	0.7207	0.7164	0.8489
40	0.8727	0.8684	0.8593	0.8475	1.0292
50	1.0982	1.0893	1.0715	1.0461	1.3329
60	1.2546	1.1977	1.2033	1.1386	1.4186
70	1.4778	1.4284	1.4233	1.3869	1.6361
80	1.5967	1.5580	1.5785	1.4798	1.9787
90	1.6609	1.6183	1.6022	1.5411	2.0406
100	2.0343	2.0054	2.0141	1.9047	2.3724

Table 2: The Modified Hausdorff distance between the original and the reconstructed data using the five methods for various interpolation interval lengths.

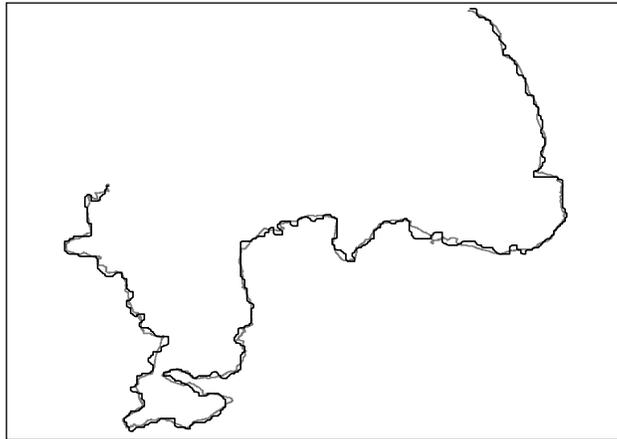


Figure 5: The original coastline (black) as well as its reconstruction by the CHM (gray).

ified Hausdorff distance between the original and the reconstructed coastline. We have compared five methods, the geometric and the algebraic methods of [12] against the bounding volume (MBRM and CHM) and the box-counting dimension (BCDM) methods proposed in this paper. The first two methods are widely used and are considered to be the best known for recurrent fractal interpolation functions. The interpolation intervals have been chosen with fixed length L of 10 to 100, i.e. by taking every 10th to 100th point as interpolation point respectively. The address intervals have been chosen with fixed length $L' = 507$ and their correspondance to the interpolation intervals has been determined as previously explained. In the case of the BCDM only, the address intervals have been chosen with length equal to the multiple of L that is closest to 507, e.g. for $L = 10$ we have chosen $L' = 510$, while for $L = 70$ we have chosen $L' = 490$. This approach was adopted in order to conform with Condition (b) of Section 3.3.

In terms of both error measures, as shown in the tables, MBRM performs better than the algebraic and geometric methods in most of the cases, while CHM performs better than all methods in all cases. The advantage of CHM over the MBRM is expected, since the convex hulls provide a tighter bound than the rectangles. On the other hand, MBRM provides an efficient, analytic calculation of the vertical scaling factors. BCDM has the poorest performance among all methods, which is not unexpected for two reasons. Firstly, the requirement of equal vertical scaling factors for all interpolation intervals limits the ability of more accurate fitting. Secondly, the method focuses on preserving the box-counting dimension of the data and not on minimizing some error measure. Nevertheless, the performance of BCDM is acceptable and comparable to the other methods. This implies that it is an interesting and efficient alternative, especially in applications where the preservation of the dimension is of importance. Such cases occur when the dimension is an intrinsic characteristic of the data. In medical applications for instance, the dimension can be used for diagnosis (see e.g. [13], [14], [15], [16]). Therefore it is desirable for the reconstructed data to preserve the dimension of the original, so as to receive the

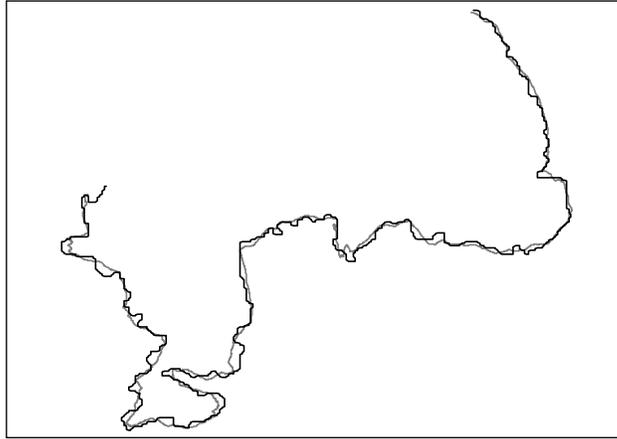


Figure 6: The original coastline (black) as well as its reconstruction as an affine FIF by the CHM (gray).

same dimension-based characterization or categorization.

It is worth mentioning that a RFIF represents the data more accurately, in general, than an affine FIF. This is expected since RFIFs are a generalisation of affine FIFs. Such an example is depicted in Figure 6, where the data of Figure 4 are modelled by an affine FIF (see also [9]) constructed, as previously, with interpolation intervals of 40 points and vertical scaling factors computed by the CHM. Comparison of Figures 5 and 6 which zoom in the same subinterval indicates that the RFIF represents the data more accurately than its affine counterpart.

5 Conclusions and further work

Three novel methods for calculating the vertical scaling factors of 1D RFIFs have been presented. The first two, following the approach of [9], use bounding volumes of appropriately chosen data points and the minimization of their symmetric difference metric in order to minimize both the Hausdorff distance as well as its modified version between the original and the reconstructed data points. One method (MBRM) uses bounding rectangles which allow analytic calculation of the vertical scaling factors, while the other (CHM) uses convex hulls with the factors calculated by an efficient algorithm of linear time complexity. The third method (BCDM) provides analytic calculation of the vertical scaling factors such that the original and reconstructed data have the same box-counting dimension.

The proposed methods have been compared against two existing ones, the geometric and the algebraic of [12], which are considered to be the best known. The test cases have included geographic data that often arise in practical applications. The results show that the proposed bounding volume methods are able to yield comparable or better results than the two afore-mentioned ones. This is especially evident for the CHM that yielded better results in all cases. In general, it is expected that the CHM performs better than the MBRM since

it provides a tighter bounding volume. On the other hand, MBRM has the advantage of analytic calculations. The proposed BCDM has poorer performance than the geometric and the algebraic methods. This is reasonable since it aims at preserving the box-counting dimension of the original data rather than minimizing some error measure. However, the results indicate that its performance is acceptable and therefore should be considered in applications where the preservation of the dimension is of importance, such as those mentioned in the previous section. The proposed methods have also been successfully tested on EEG and MRA data, indicating their applicability in medical applications.

As in the case of affine interpolation, the methods presented in this paper can be efficiently combined with an algorithm for finding the optimal interpolation intervals, such as the iterative algorithm of [12]. Future work will focus on extending these methods to fractal interpolation surfaces.

Acknowledgements

Financial support was provided by the Hellenic General Secretariat for Research & Technology, PENED Programme 70/3/8405.

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.
- [3] M. F. Barnsley, J. H. Elton, and D. P. Hardin. Recurrent iterated function systems. *Constr. Approx.*, 5:3–31, 1989.
- [4] R. Brinks. A hybrid algorithm for the solution of the inverse problem in fractal interpolation. *Fractals*, 13(3):215–226, 2005.
- [5] I. Daubechies. *Ten lectures on wavelets*. SIAM, Philadelphia, 1992.
- [6] F. R. Gantmacher. *Matrix Theory*, volume 2. Chelsea Publishing Company, New York, 2000.
- [7] L. I. Levkovich-Maslyuk. Wavelet-based determination of generating matrices for fractal interpolation functions. *Regular And Chaotic Dynamics*, 3(2):20–29, 1998.
- [8] P. Manousopoulos, V. Drakopoulos, and T. Theoharis. Curve fitting by fractal interpolation. *Transactions on Computational Science*, 1:85–103, 2008.
- [9] P. Manousopoulos, V. Drakopoulos, and T. Theoharis. Parameter identification of 1D fractal interpolation functions using bounding volumes. *Journal of Computational and Applied Mathematics*, 233(4):1063–1082, 2009.
- [10] P. Manousopoulos, V. Drakopoulos, T. Theoharis, and P. Stavrou. Effective representation of 2D and 3D data using fractal interpolation. In

Proceedings of International Conference on Cyberworlds, pages 457–464. IEEE Computer Society, 2007.

- [11] P. R. Massopust. *Interpolation and approximation with splines and fractals*. Oxford University Press, 2009.
- [12] D. S. Mazel and M. H. Hayes. Using iterated function systems to model discrete sequences. *IEEE Trans. Signal Processing*, 40:1724–1734, 1992.
- [13] M. A. Navascues and M. V. Sebastian. Fitting curves by fractal interpolation: An application to the quantification of cognitive brain processes. In M. M. Novak, editor, *Thinking in Patterns: Fractal and Related Phenomena in Nature*, pages 143–154. World Scientific, Singapore, 2004.
- [14] M. A. Navascues and M. V. Sebastian. Spectral and affine fractal methods in signal processing. *International Mathematical Forum*, 1(29):1405–1422, 2006.
- [15] A. I. Penn, L. Bolinger, M. D. Schnall, and M. H. Loew. Discrimination of MR images of breast masses with fractal interpolation function models. *Academic Radiology*, 6(3):156–163, 1999.
- [16] A. I. Penn and M. H. Loew. Estimating fractal dimension with fractal interpolation function models. *IEEE Transactions on Medical Imaging*, 16(6):930–937, 1997.
- [17] D. Saupe. Efficient computation of julia sets and their fractal dimension. *Physica D*, 28(3):358–370, 1987.
- [18] 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.
- [19] C. Zhao, W. Shi, and Deng Y. A new Hausdorff distance for image matching. *Pattern Recognition Lett.*, 26:581–586, 2005.