Available online at www.sciencedirect.com
SCIENCE DIRECT®
ELSEVIER
Physica D 179 (2003) 129–152
PHYSICA D
www.elsevier.com/locate/physd

# Lacunarity definition for ramified data sets based on optimal cover

Charles R. Tolle*, Timothy R. McJunkin, David T. Rohrbaugh[1], Randall A. LaViolette[2]

*Idaho National Engineering and Environmental Laboratory, Department of Industrial and Material Technologies, Idaho Falls, ID 83415-2210, USA*

Received 31 December 2000; received in revised form 30 September 2002; accepted 10 January 2003
Communicated by R.P. Behringer

**Abstract**

Lacunarity is a measure of how data fills space. It complements fractal dimension, which measures how much space is filled. This paper discusses the limitations of the standard gliding box algorithm for calculating lacunarity, which leads to a re-examination of what lacunarity is meant to describe. Two new lacunarity measures for ramified data sets are then presented that more directly measure the *gaps* in a ramified data set. These measures are rigorously defined. An algorithm for estimating the new lacunarity measure, using Fuzzy-C means clustering algorithm, is developed. The lacunarity estimation algorithm is used to analyze two- and three-dimensional Cantor dusts. Applications for these measures include biological modeling and target detection within ramified data sets.


*PACS:* 02.90.+p; 02.60.−x; 42.30.Sy; 5.45

*Keywords:* Fractals; Pattern recognition; Ramified data; Mathematical methods in physics

## 1. Introduction

Many modeling and pattern recognition applications need methods to numerically *quantify* the visual *look* of measured data sets. A number of mathematical approaches are currently used to quantify a data set's *look*, including fractal-based methods such as fractal dimension, lacunarity, and connectivity [1], as well as non-fractal-based methods such as variograms [2,3]. Many researchers have successfully used fractal dimension to measure a data set's *look* for target detection [4–8]. However, fractal dimension alone does not fully describe this visual *look* because it does not fully describe the space-filling characteristics of data. Fractal dimension only measures *how much* space is filled. Lacunarity complements fractal dimension by measuring *how* the data fills the space.

* Corresponding author. Tel.: +1-208-526-1895; fax: +1-208-526-0690.
*E-mail address:* tollcr@inel.gov (C.R. Tolle).
[1] Member of the Advanced Information and Communication Systems Department at the INEEL.
[2] Member of the Chemistry Department at the INEEL.

0167-2789/03/$ – see front matter 
doi:10.1016/S0167-2789(03)00029-0

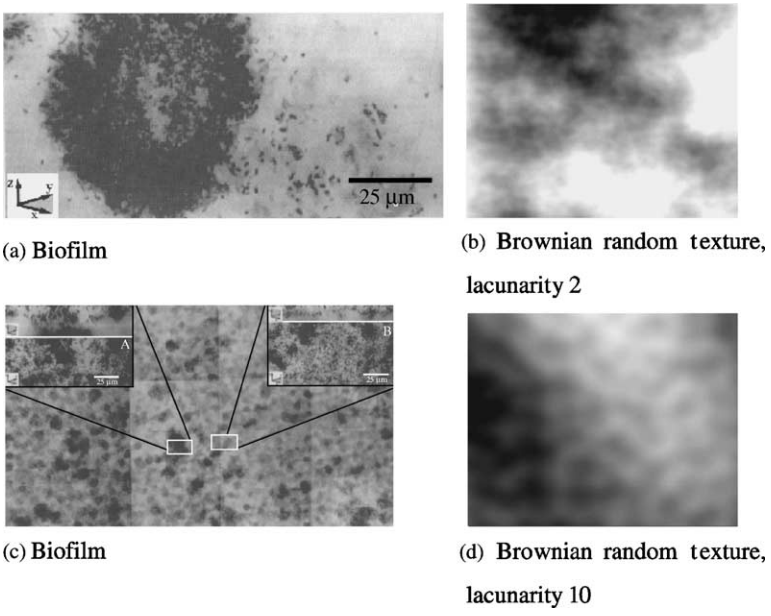



Fig. 1. Comparison of real biofilms from Möller et al. [11] with random Brownian fractal textures with a fractal dimension of 2.1, and lacunarity of 2 and 10 per Musgrave's definition.

In this paper we investigate lacunarity, the second important spatial characteristic of data sets mentioned by Mandelbrot in [1]. We propose new measures for lacunarity of ramified data sets. These measures provides a basis for developing cross-cutting technology that will be used by both the Department of Energy (DOE) and the Department of Defense (DOD) in many modeling, detection, and control applications. We intend to use these measures of lacunarity, along with fractal dimension estimations, to develop biological models that quantitatively describe biofilm structure and growth (similar to [9–11]) so that a control strategy can be developed based on these feedback terms.

Synthetic fractal textures have been produced that have a qualitative *look* that is very similar to that of biofilms. In Fig. 1 we compare biofilms from Möller et al.'s work [11] to synthetic Brownian fractal textures produced using Musgrave's [12] texture generation algorithms. Note that while both synthetic fractal textures (Fig. 1b and d) have the same fractal dimension, their lacunarities are quite different according to Musgrave's definition.

Fractal dimension and lacunarity apply to images in many research fields. For example, they have been used to quantify and segment images into background clutter and desired targets [4–8]. The lacunarity measures for ramified data that we propose scale easily to data in any dimensional space and are related to the formal definition of fractal dimension. This scaling ability allows direct application of fractal dimension and lacunarity to data fusion problems where multi-spectral sensors are gathering and fusing scene data into a hyper-dimensional set for quantification.

The next section provides background on fractal dimension and the problems inherent in today's leading method for calculating lacunarity. To overcome these problems, we return to the basic definition and concepts of lacunarity. Mandelbrot [1] introduces the term lacunarity from the Latin "lacuna" meaning gap. This paper will define *gap* as the distance between optimal covers, with the definition of optimal cover being used in the definition of the Hausdorff–Besicovitch (HB) fractal dimension measure. Formal definitions for new lacunarity measures for ramified data sets are presented in Section 3. Since HB covers are impractical to calculate, our algorithm will use the distances between sub-optimal covers [6,7] generated with the Fuzzy-C means clustering algorithm [13]. The Fuzzy-C means

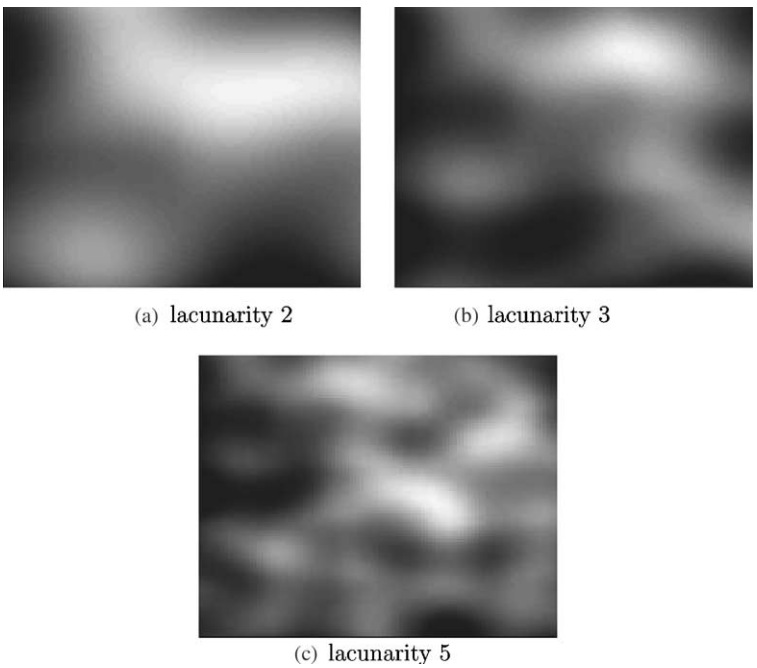
implementations of these cover-based lacunarity measures are presented in Sections 4 and 5 and compared with analytical results in Section 6. Section 7 shows that the new lacunarity measures discriminate between sets with diverse structures and identify sets that have similar structures. Finally, the paper closes with a discussion of the future direction of this research, a new method for measuring lacunarity of dense data sets.

## 2. Background

Of all the methods for quantifying a data set's *look* [1–8], fractal dimension is perhaps the easiest to understand. However, as discussed above, fractal dimension does not fully quantify a texture's *look*—textures with identical fractal dimensions but small variations in lacunarity can vary substantially in their *look*. In general, fractal dimension describes the space-filling capabilities of a set, i.e. the amount of space-filled, or the mass in some sense, while lacunarity describes the spatial size of gaps and their structure within a set [1], i.e. how the space is filled or the mass distribution.

The three images in Fig. 2, representing textures produced by Musgrave's Brownian random texture generation algorithm, illustrate lacunarity—they have the same fractal dimension but different lacunarities. (The lacunarity parameter in Musgrave's algorithm alters the spatial scaling factor between self-similar levels, which are then super-imposed by addition to form the final image. So, for a lacunarity parameter of 2, the second self-similar level will have approximately twice the spatial frequency of the base surface.)

To date, the leading method for estimating lacunarity is the gliding box algorithm, which is based on a localized mass calculation [1,14,15]. The algorithm, as discussed in Plotnick et al.'s paper [14], is described here for completeness. The gliding box algorithm is similar to the box counting algorithm used to estimate fractal dimension. One chooses a unit box of size $r$ and counts the number of set points within it (the mass). This procedure is then repeated



(a) lacunarity 2 (b) lacunarity 3

(c) lacunarity 5

Fig. 2. Musgrave's Brownian random fractal textures with fractal dimension of 2.5 and lacunarities of 2, 3, and 5.

Table 1
Data sets constructed to demonstrate continuous variations in lacunarity estimates obtained via the gliding box algorithm, see Fig. 3

| Set name | Description | Definition |
|---|---|---|
| *Baseline sets for glide box and* $\hat{\mathcal{L}}_{\max}$ *comparisons* | | |
| $R_{s1}$ | Regularly spaced data | {21.0, 24.5, 28.0, . . . , 241.5} |
| $R_{s2}$ | $R_{s1}$ with five points added to the right | {21.0, 24.5, 28.0, . . . , 259.0} |
| $R_{s3}$ | $R_{s1}$ with five points added to the left | {3.5, 7.0, 10.5, . . . , 241.5} |
| $R_{s4}$ | $R_{s1}$ with five points added each side | {3.5, 7.0, 10.5, . . . , 259.0} |
| $R_{s5}$ | $R_{s1}$ with five points removed each side | {38.5, 42.0, 45.5, . . . , 224.0} |

as the box is centered, in turn, about each point within the set, creating a distribution of box masses $B(p, r)$, where $B$ is the number of boxes with $p$ points and radius $r$. This distribution is converted into a probability distribution, $Q(p, r)$, by dividing by the total number of boxes of size $r$. Next, one calculates the first and second moments of the box mass probability distribution:

$$Z^{(1)}(r) = \sum_p pQ(p, r), \tag{1}$$

$$Z^{(2)}(r) = \sum_p p^2 Q(p, r). \tag{2}$$

The gliding box lacunarity is then defined as:

$$\mathcal{L}_{\mathrm{GB}} = \frac{Z^{(2)}(r)}{Z^{(1)}(r)^2}. \tag{3}$$

As an illustration, we applied this lacunarity measure, $\mathcal{L}_{\mathrm{GB}}$, to the data sets $R_{si}$ given in Table 1; the resulting lacunarity estimates are shown in Fig. 3. $\mathcal{L}_{\mathrm{GB}}$ strongly differentiates between these data sets, which is highly

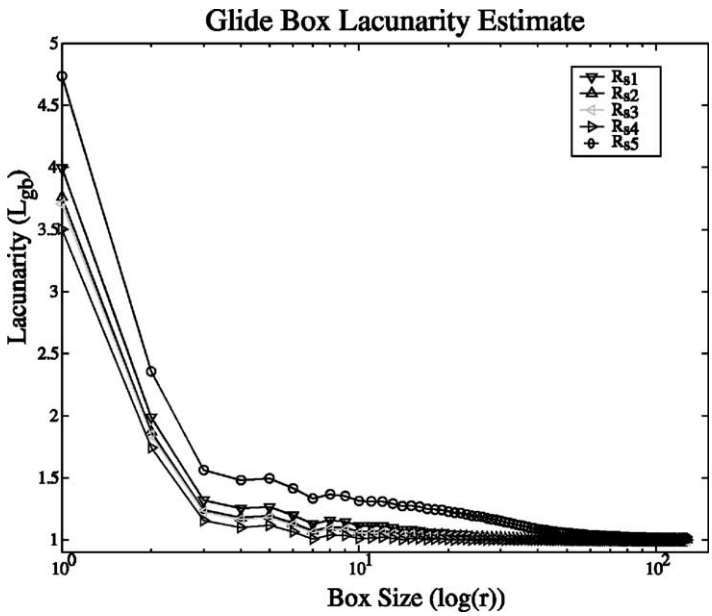



Fig. 3. Gliding box algorithm lacunarity estimations for the $R_{si}$ data sets defined in Table 1.

disconcerting considering that the basic internal gap contained within each of these sets is exactly the same, 3.5. Such differentiation become problematic when generic data sets are compared, as in the case of biofilm images. The only comforting result for the $\mathcal{L}_{\mathrm{GB}}$ measure is that the results for $R_{s2}$ and $R_{s3}$ lay on top of each other. $R_{s2}$ and $R_{s3}$ have the same mass and internal distribution, the mass is merely shifted in the space.

These results point out the $\mathcal{L}_{\mathrm{GB}}$ measure's strong sensitivity to mass change. For the $R_{si}$ sets, it appears that $\mathcal{L}_{\mathrm{GB}}$ is more sensitive to mass change than to the gaps that lacunarity is intended to measure. This conclusion is reinforced by the minimum radius gap measurements given for these sets, i.e. the $y$-intercepts of Fig. 3. Only $R_{s4}$, which fills the space uniformly was found to have the known gap size, 3.5. Moreover, $\mathcal{L}_{\mathrm{GB}}$ does not consistently quantify changes in the lacunarity of well-known fractal families and sets.

Examples of sets from two families of two-dimensional Cantor sets that have been created to maintain a constant fractal dimension while allowing the gaps to vary in size are shown in Fig. 4. (For notation purposes, we group all Cantor sets with the same fractal dimension but different construction into what will be known as a family of Cantor sets.) They were formed by applying a scaling factor and changing the number of copies of the set made on each successive construction level. Meakin [16] points out that the similarity dimension of the set can be maintained by applying a scaling factor adjustment as the number of copies is changed, thereby holding the fractal dimension constant while allowing the gaps sizes within the set to change.

The two-dimensional Cantor sets in Fig. 4 and three-dimensional set in Fig. 5 were created as follows. First, choose the similarity dimension, $D_s$ between $L-1$ and $L$, where $L$ is the topological dimension, e.g. $L=2$ for a two-dimensional Cantor set. Second, choose the number of copies in each dimension, $N_{\mathrm{c}}$, so the total number of copies is $N_{\mathrm{c}}^L$. Third, apply the following constraint to find the scaling factor, $K$, that maintains $D_s$:

$$\log(K) = \frac{\log(N_{\mathrm{c}}^L)}{D_s}. \tag{4}$$

This constraint only allows creation of a symmetric generalized Cantor set with changing gap sizes.

The generalized 1D symmetric Cantor set can be defined by first defining the base interval, $a_0^0$, and its left endpoint, $b_0^0$ as:

$$a_0^0 = [0,1] = [b_0^0, 1], \tag{5}$$

$$b_0^0 = 0. \tag{6}$$

The notation $a_k^n$ indicates the $k$ interval of the Cantor generation set on the $n$th scale. Likewise, $b_k^n$ indicates the left endpoint of each interval, $k$, for the Cantor generation set on the $n$th scale. The generalized gap, $G^n$, between subsets on a particular scale is defined in terms of the remaining empty space:

$$G^n = \left(\frac{1}{K}\right)^{n-1} - N_{\mathrm{c}}\left(\frac{1}{K}\right)^n, \quad \text{for } n > 0. \tag{7}$$

The intervals for the $n$th scale are:

$$a^n_{(N_{\mathrm{c}}m)+j} = \left[b_m^{n-1} + jG^n + j\left(\frac{1}{K}\right)^n, b_m^{n-1} + jG^n + (j+1)\left(\frac{1}{K}\right)^n\right]$$
$$\text{for } j \in \{0, N_{\mathrm{c}}-1\}, \quad m \in \{0, (N_{\mathrm{c}})^{n-1}-1\}. \tag{8}$$

Using this generalized interval definition, we can write the $n$th scale Cantor generation set as:

$$V_n = \bigcup_{k=0}^{(N_{\mathrm{c}})^n-1} a_k^n. \tag{9}$$

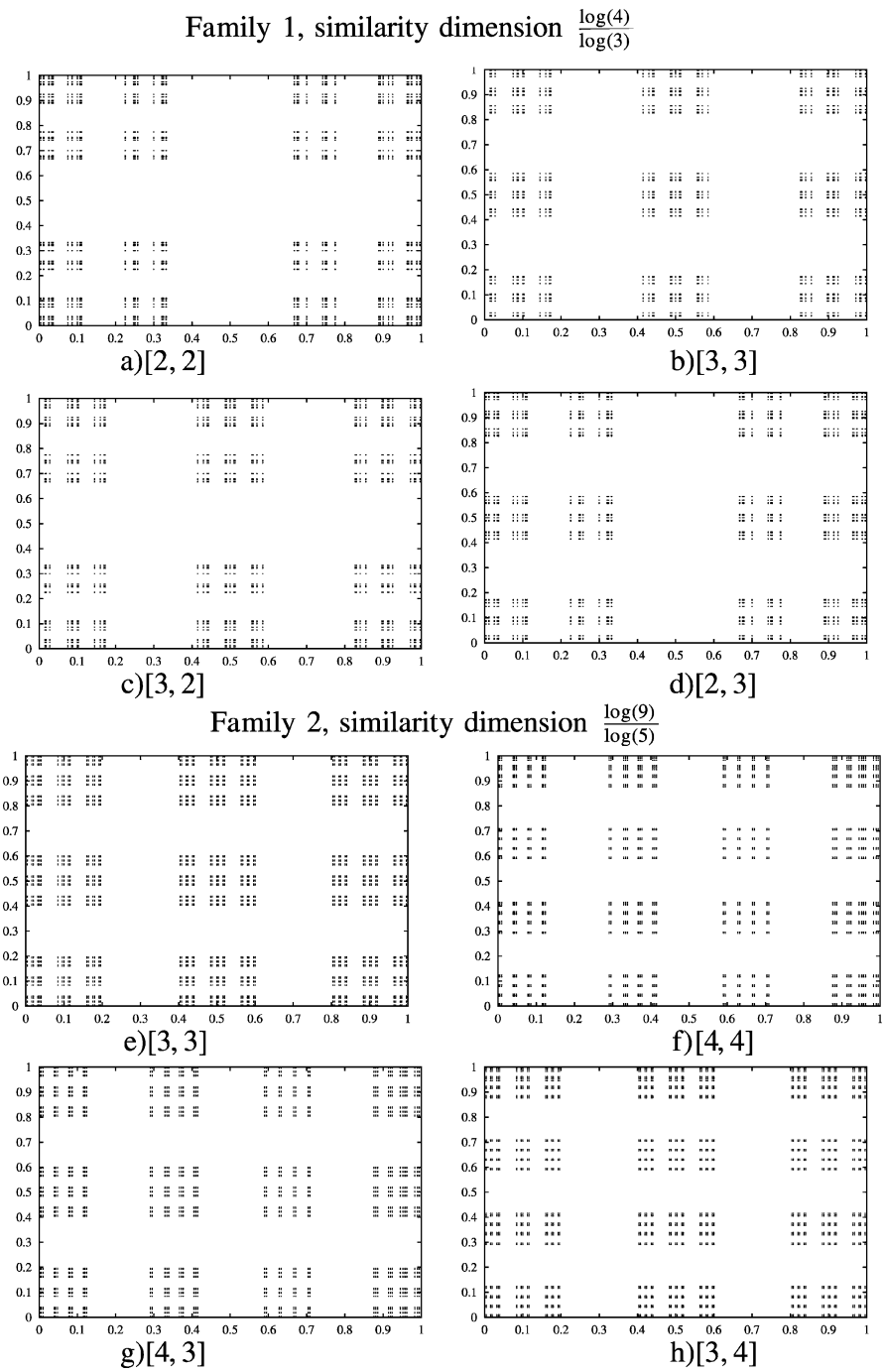



Fig. 4. Two families of two-dimensional Cantor sets with various fractal dimensions defined by the number of copies in each dimension, i.e. $[N_c^x, N_c^y]$. Family 1 has similarity dimension $\log(4)/\log(3)$, Family 2 has similarity dimension $\log(9)/\log(5)$.

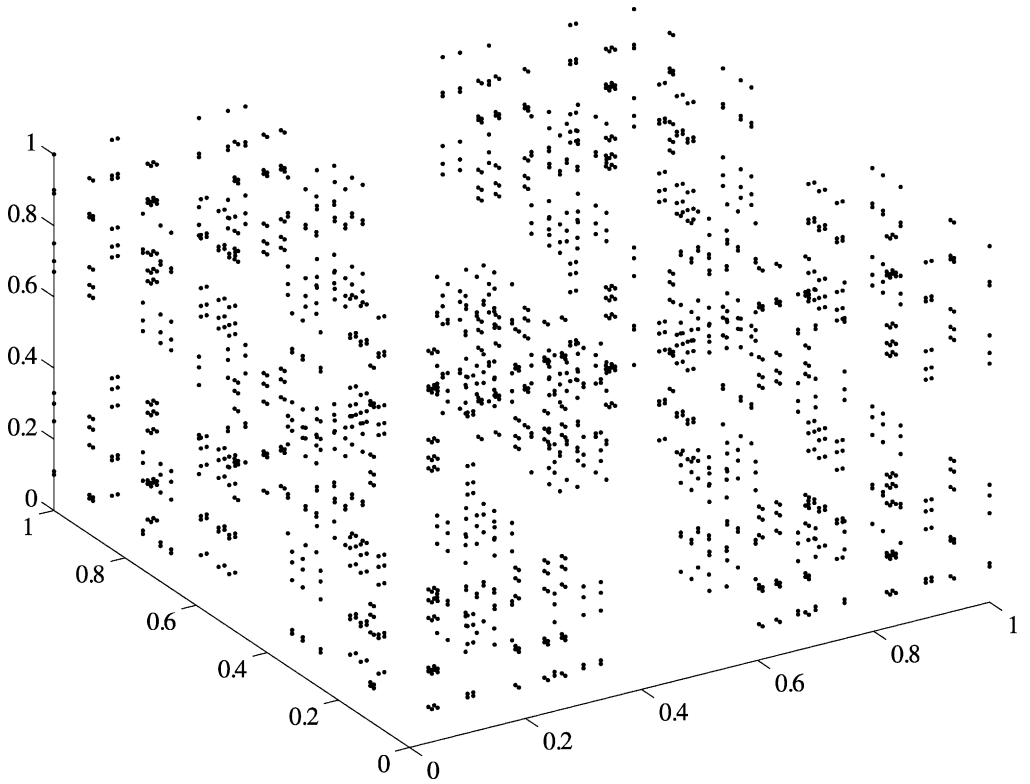



Fig. 5. Visualization of a standard three-dimensional 1/3 removal Cantor set ([2, 2, 2] Cantor with fractal dimension log(8)/log(3)) used in testing the new lacunarity measures.

The generalized 1D symmetric Cantor set is then formally defined as:

$$\mathcal{V} = \bigcap_{n=1}^{\infty} V_n. \tag{10}$$

Asymmetric $L$-dimension Cantor sets can also be achieved via the constraint:

$$\log(K_d) = \frac{L \log(N_{c,d})}{D_s}. \tag{11}$$

The number of copies, $N_{c,d}$, can be defined to be different for any dimension, $d$, allowing the *gaps* in each dimension to vary as:

$$G_{n,d} = \left(\frac{1}{K_d}\right)^{n-1} - N_{c,d}\left(\frac{1}{K_d}\right)^{n}. \tag{12}$$

Here we define the $L$-dimension asymmetric Cantor set, $\mathcal{V}^L$, as an $L$-tuple of 1D generalized symmetric Cantor sets, $\mathcal{V}_d$:

$$\mathcal{V}^L = \{[\{x_1, x_2, \ldots, x_L\}] | x_d \in \mathcal{V}_d, \quad d = 1, 2, \ldots, L\}. \tag{13}$$

Visual inspection shows that the gaps in the [2, 2] Cantor set with $D_s = \log(4)/\log(3)$ are larger than those in the [3, 3] Cantor set with $D_s = \log(9)/\log(5)$, see Figs. 4a and e, respectively. However, $\mathcal{L}_{GB}$ quantifies them as nearly

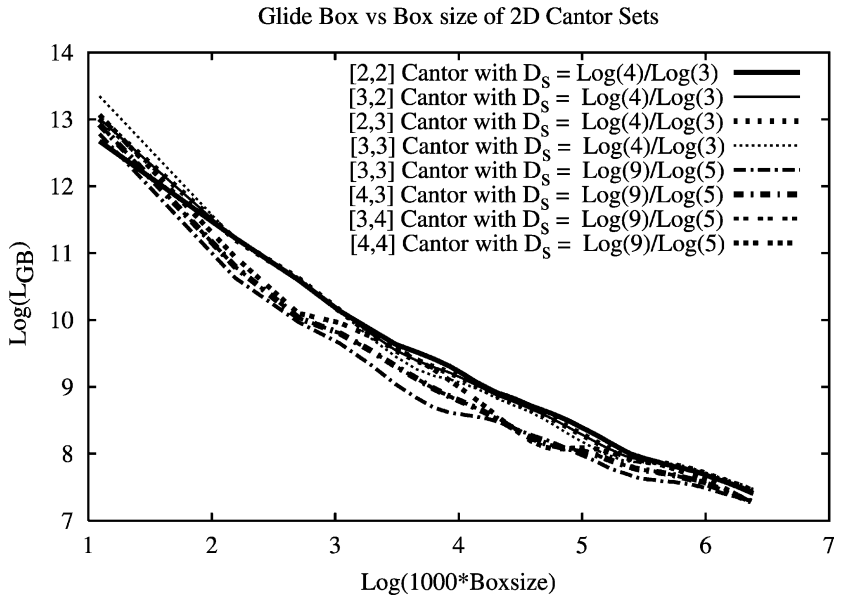



Fig. 6. Glide box lacunarity analysis, $\mathcal{L}_{\mathrm{GB}}$, for $[N_{\mathrm{c}}^{x}, N_{\mathrm{c}}^{y}]$ Cantor sets shown in Fig. 4. It is difficult, if not impossible, to distinguish among their *gap* structures using data such as this.

the same, see Fig. 6. Moreover, comparison of results within families in Fig. 6 reveals another inconsistency. In the family of $D_s = \log(9)/\log(5)$ Cantor sets, the $\mathcal{L}_{\mathrm{GB}}$ for $N_{\mathrm{c}} = [3, 3]$ is below $\mathcal{L}_{\mathrm{GB}}$ for $N_{\mathrm{c}} = [4, 4]$. This result is contradicted in the family $D_s = \log(4)/\log(3)$, where $\mathcal{L}_{\mathrm{GB}}$ for $N_{\mathrm{c}} = [2, 2]$ is above $\mathcal{L}_{\mathrm{GB}}$ for $N_{\mathrm{c}} = [3, 3]$ for some box sizes ($\log(1000 \times \text{box size}) > 3$). This inconsistency between these two Cantor set families makes it nearly impossible to compare the results among fractal families using the glide box method. Furthermore, these types of inconsistencies makes it impossible to use this method for applications such as biofilm quantification where one is not assured that any particular global fractal or family of fractals exists in the image being analyzed.

## 3. Lacunarity measure based on optimal cover

Our measure of lacunarity is based on the definition of optimal cover used to define the HB dimension, which is the traditional definition of fractal dimension. The HB dimension, $D_h(A)$, where $A$ denotes the data set, is defined in the following manner [17]:

Let

$$\Re^p = \{x | x = (x_1, \ldots, x_p), \quad x_i \in \Re\} \tag{14}$$

for some natural number $p$, which is the dimension of the data set. Define the diameter of an open ball, $C_i$, in $\Re^p$:

$$\mathrm{diam}(C_i) = \sup\{d_{\mathrm{e}}(x, y) | x, y \in C_i\}, \tag{15}$$

where $d_{\mathrm{e}}(x, y)$ denotes the Euclidean distance function. Next, define an open cover of $A$:

$$A \subset \bigcup_{i=1}^{\infty} C_i. \tag{16}$$

Define:

$$h_\epsilon^s(A) = \inf\left\{ \sum_{i=0}^{\infty} \operatorname{diam}(C_i)^s \,\middle|\, \begin{matrix}\{C_1, C_2, \ldots\} \text{ open cover} \\ \text{of } A \text{ with } \operatorname{diam}(C_i) \le \epsilon\end{matrix} \right\}. \tag{17}$$

Finally, define the $s$-dimensional Hausdroff measure of $A$ as:

$$h^s(A) = \lim_{\epsilon \to 0} h_\epsilon^s(A). \tag{18}$$

Given the above definitions, the HB dimension is defined as:

$$D_h(A) = \inf\{s | h^s(A) = 0\} = \sup\{s | h^s(A) = \infty\}. \tag{19}$$

We define the measure of the distance (i.e. the *gaps*) between the covering elements, $C_i$, in the Hausdorf fractal dimension as a measure for lacunarity. This measure is independent of the traditional fractal dimension, i.e. Eq. (19), but highly consistent with fractal dimension because they share much of the underlying mathematics. The basic concept is: find a cover of the points within the set, then find a measure of the cover separations that will quantify the lacunarity of the set, i.e. measure the gaps between the covering elements. If we assume that the HB dimension exists and can be determined, one can use the optimal cover obtained there to calculate the lacunarity of $A$.

We begin by finding a cover, $C^\bullet$, for a given number of covering elements, denoted by the calligraphy $\mathcal{C}$:

$$C^\bullet = \{C_1, C_2, \ldots, C_\mathcal{C}\} = \left\{ C_k | k = 1, 2, \ldots, \mathcal{C};\ A \subset \bigcup_{i=1}^{\mathcal{C}} C_i \right\}, \tag{20}$$

$$c_i = \text{center of ball } C_i, \tag{21}$$

$$s_{ij} = d_e(c_i, c_j) - \tfrac{1}{2}(\operatorname{diam}(C_i)) - \tfrac{1}{2}(\operatorname{diam}(C_j)). \tag{22}$$

We define the *structure* of the gaps connecting the covers using the concept of a minimum spanning tree [18], $\mathcal{S}$, as follows. First, we define the minimum *connecting* spanning tree of a cover, $\mathcal{T}(C^\bullet)$, to be the spanning tree of $\mathcal{C} - 1$ line segments that fully connects all the balls, $C_i$, along connecting lines between the cover centers, $c_i$, within the cover, $C^\bullet$, using the minimum distance, see Fig. 7. The $\mathcal{C} - 1$ line segments are denoted by $S^k$, which in turn denotes a $s_{ij}$. Then

$$\mathcal{T} = \mathcal{S}, \tag{23}$$

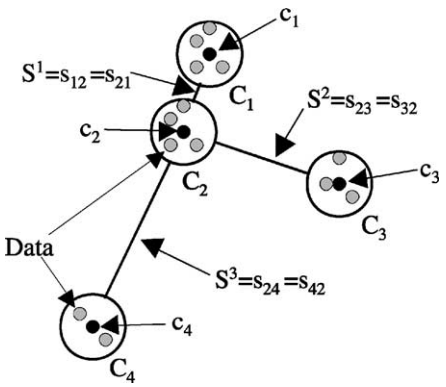



Fig. 7. A cover ($C^\bullet$, $\mathcal{C} = 4$) and spanning tree $\mathcal{S} = \{S^1, S^2, S^3\}$. In this case $\mathcal{S}$ is the minimal connecting spanning tree, $\mathcal{T}(C^\bullet)$.

where

$$\mathcal{S} = \{S^k | k = 1, 2, \ldots, \mathcal{C}-1;\ \ m \neq n;\ \ m, n \in [1, 2, \ldots, \mathcal{C}];\ \ S^k = s_{mn};$$
$$C_m, C_n \in C^\bullet \text{ and all cover centers are connected via } S^k s\}, \tag{24}$$

and

$$\mathcal{S} \text{ minimizes } \sum_{j=1}^{\mathcal{C}-1} S^j. \tag{25}$$

Assuming $D_h(A)$ exists, then there exists a family of open covers, $\mathcal{F}$, that were used in calculating $D_h(A)$, i.e. the open covers used within the calculation of $h^s(A)$ for both the inf and sup in Eq. (19). Choose the open cover $C^\star$ so that

$$C^\star = \lim_{\mathcal{C}\to\infty} C^\bullet \in \mathcal{F} \text{ that minimizes } \sup_{\mathcal{T}(C^\bullet)} S^k. \tag{26}$$

Using the cover $C^\star$, the maximum lacunarity measure is defined as:

$$\mathcal{L}_{\max} = \sup_{S^k \in \mathcal{T}(C^\star)} S^k. \tag{27}$$

An alternative measure, total lacunarity, is:

$$\mathcal{L}_{\text{total}} = \sup \sum_{k=1}^{\mathcal{C}} S^k; \quad S^k \in \mathcal{T}(C^\star). \tag{28}$$

Our concept is to define lacunarity based on either the maximum cover separation or the total cover separation, using the covers that define the HB dimension. By doing this, we create a lacunarity measure that is consistent with the community's current notion of fractal dimension, i.e. HB dimension, while decoupling it from the gliding box method of calculating the fractal dimension. Although not presented here, an analysis of the distribution of the lengths of the spanning tree segments may also be useful.

## 4. Sub-optimal cover lacunarity algorithm: Fuzzy-C method

Since finding the optimal covering for the HB dimension is generally difficult, we use a clustering algorithm, Fuzzy-C means [13], to group data points and calculate an approximate (sub-optimal) cover [6,7]. This Fuzzy-C means method of calculating lacunarity is a practical implementation of the mathematical definition given in Section 3.

First, a cover is obtained by clustering the data via the Fuzzy-C means algorithm. These fuzzy clusters are then turned into crisp sets of covering elements by forcing each point into the covering element, i.e. cluster, that has that point's maximum membership value. The size of each covering element is then determined using a singular value decomposition of its scatter matrix. Finally, the covering element sizes are subtracted from the distances between cover element centers to create the lacunarity estimate. Such a subtraction directly estimates the separation between covering elements themselves, see Fig. 8.

In short, a *good* approximation of the maximum cover separation, $\hat{\mathcal{L}}_{\max}$, or total cover separation, $\hat{\mathcal{L}}_{\text{total}}$, can be obtained by exploiting these cluster characteristics as cover elements via cluster scatter matrix statistics (i.e. the vectors defined by the singular value and its unit directional vector, $\sqrt{\sigma_j^i} w_j^i$). How this is done is important in

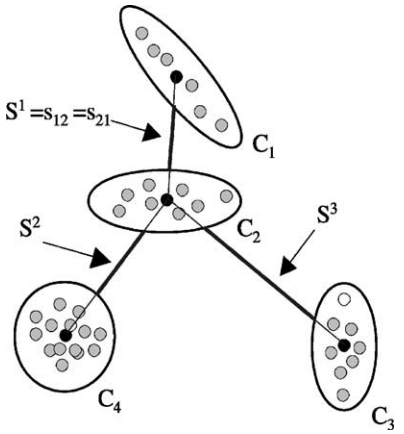



Fig. 8. A minimal spanning tree with ideal cluster separations, i.e. $S^1$, $S^2$, and $S^3$.

obtaining a *good* cover approximation. Moreover, a *good* cover approximation is needed because the sizes of the covering elements obtained via the fuzzy clustering method tend to be hyper-ellipses not hyper-balls. Even so, this restriction is relaxed in the limit as the diameters of the covering elements go to zero, thereby generally making the covering elements simple balls in hyper-space.

Consider the separation calculation introduced above. The singular values, $\sigma_j^i$, of these cluster scatter matrices and their corresponding unit vectors, $w_j^i$, provide the regional spatial characteristics of each covering element, see Fig. 9. We approximate the cluster separation by projecting the cluster characteristics (i.e. the vectors defined by the singular value and its unit directional vector, $\sqrt{\sigma_j^i} w_j^i$) onto the line connecting the cluster centers, $\bar{s}_{ij}$. Next, we find the length of the remaining distance between the clusters, $s_{ij}$ to obtain the separation estimation, see Figs. 9 and 10. One can calculate this distance for each cluster pair, $(i, j)$, see Fig. 11.

Once the distance for each cluster pair is known, the minimal (connecting) spanning tree, $\mathcal{T}$, for the Fuzzy-C cover (based on Euclidean distance) can be obtained. This is the minimum distance set of vectors that fully connects the cover regions shown in Fig. 8. We normalize these distances so that the lacunarity calculations remain comparable over differences in data set ranges and scales. This is accomplished by calculating the full data set's scatter matrix and obtaining the maximum singular value for the full data set, $\sigma^\star$. This singular value is then used to normalize the distances and make them unitless, $\hat{\mathcal{T}}$, thereby avoiding unit length dependence within our lacunarity estimation. This independence, the main objective of our method, is needed for applications such as biofilm characterization.

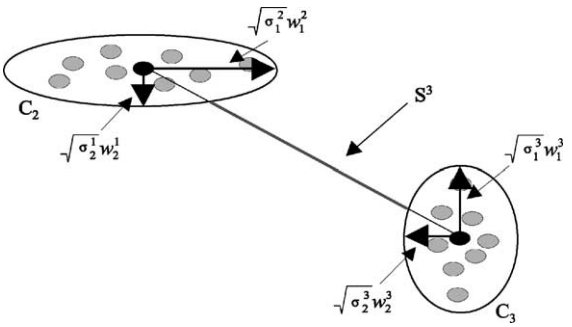



Fig. 9. An optimal distance between two elliptical clusters.

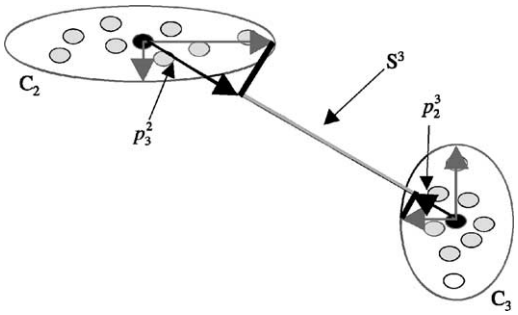



Fig. 10. An approximated distance between two elliptical clusters.

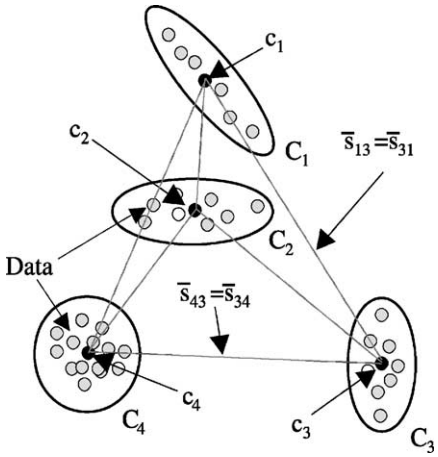



Fig. 11. All possible connecting vectors for use within the minimal spanning tree calculation using elliptical clusters.

It is clear that the maximum cover separation, $\hat{\mathcal{L}}_{\max}$, should stabilize beyond some fixed number of covering elements as long as the lacunarity is above 0; i.e., as the number of clusters increases, at some point the maximum separation will remain the same because further covering elements will only subdivide the already well-separated covered regions. As introduced earlier, one can also consider the total of the cover separation as a measure of lacunarity, $\hat{\mathcal{L}}_{\text{total}}$. This is somewhat more problematic in that to have a lacunarity value, the sum of the gaps on all scales must shrink fast enough for a non-infinite value to occur. However, for finite data this problem is not as important. Results of these lacunarity calculations are given in Section 7.

## 5. The nuts and bolts: the Fuzzy-C lacunarity algorithm

The heart of calculating fractal dimension or lacunarity is finding an optimal cover. A simple solution to this problem can generally be obtained using one of the various clustering methods. In this paper, we have chosen the standard Fuzzy-C means clustering algorithm [13]. Although more advanced methods, such as the Fuzzy-C varieties, preserve the data clusters better, for our goal of covering a set with simple balls in hyper-space, the simple Euclidean-based Fuzzy-C means approach seems appropriate. As a side note, the reason for using Fuzzy-C means

clustering rather than C-means clustering is that Fuzzy-C means clustering has a simple, robust iterative solution and C-means clustering currently does not, see Appendix B.

The Fuzzy-C means algorithm solves the following problem:

$$\min \sum_{i=1}^{\mathcal{C}} \sum_{k=1}^{N} u_{ik}^2 \|x_k - c_i\|^2, \quad \text{W.C. } u_{ik} \in [0,1], \qquad \sum_{i=1}^{\mathcal{C}} u_{ik} = 1 \quad \forall k = 1, 2, \ldots, N, \tag{29}$$

where $\mathcal{C}$ is the number of clusters, $N$ the number of data vectors $\{x_k\}$ being clustered ($x$ is an $L$-tuple), $u_{ik}$ the membership value of the $k$th data vector in the $i$th cluster, and $c_i$ the center (mean) vector of the $i$th cluster. The $\| \bullet \|$ is the Euclidean norm operator. This problem can be solved [13] using the Picard iteration [19], also known as the method of successive approximations [20], see Appendix B.

Once a cover has been chosen, the scatter matrices, $\mathcal{M}$, for each cluster are calculated. To do this, we form crisp clusters through the simple assignment of each point, $x_k$, to the cluster in which it has the maximum $u_{ik}$. (We *throw away* the fuzzy cluster information in this process.) The scatter matrix for a cluster, $C_i$, can be written as [21]:

$$\mathcal{M}^i = \sum_{x_k \in C_i} (x_k - c_i)(x_k - c_i)^T. \tag{30}$$

However, we need the average scatter for our calculation:

$$\hat{\mathcal{M}}^i = \frac{1}{N_{C_i}} \sum_{x_k \in C_i} (x_k - c_i)(x_k - c_i)^T, \tag{31}$$

where

$$N_{C_i} = \text{cardinality of the data points within cluster } \ C_i, \tag{32}$$

$$N = \sum_{i=1}^{\mathcal{C}} N_{C_i}. \tag{33}$$

This matrix can be decomposed into its principal components using the singular decomposition:

$$\hat{\mathcal{M}}^i = W^i \Sigma^i {G^i}^{\mathrm{H}}, \tag{34}$$

$$\hat{\mathcal{M}}^i = W^i \begin{bmatrix} \sigma_1^i & 0 & \cdots & 0 \\ 0 & \sigma_2^i & & 0 \\ & & \vdots & \\ 0 & 0 & \cdots & \sigma_L^i \end{bmatrix} {G^i}^{\mathrm{H}}, \tag{35}$$

$$\hat{\mathcal{M}}^i = \sum_{l=1}^{L} \sigma_l^i w_l^i {g_l^i}^{\mathrm{H}}, \tag{36}$$

where ${G^i}^{\mathrm{H}}$ denotes the Hermitian of $G^i$, $\sigma_i$ are the singular values, and $W^i$ and $G^i$ are matrices which contain the orthonormal vectors describing the direction of the singular values. Each of these components is then projected onto the vector, $\bar{s}_{ij}$, separating the clusters $C_i$ and $C_j$. The maximum projected vectors, $p_j^i$ and $p_i^j$, from each cluster are used to find the cluster separations (Fig. 10):

$$\bar{s}_{ij} = c_i - c_j, \tag{37}$$

$$p_j^i = \max_{l=1,2,\ldots,L} \frac{\sqrt{\sigma_l^i}\, w_l^i \bullet \bar{s}_{ij}}{\bar{s}_{ij} \bullet \bar{s}_{ij}}, \tag{38}$$

$$p_i^j = \max_{l=1,2,\ldots,L} \frac{\sqrt{\sigma_l^j}\, w_l^j \bullet \bar{s}_{ij}}{\bar{s}_{ij} \bullet \bar{s}_{ij}}, \tag{39}$$

$$s_{ij} = (1 - p_j^i - p_i^j)\|\bar{s}_{ij}\|, \tag{40}$$

where $\bullet$ denotes the dot product. (The units of $\sigma_i$ are squared in the above equations, requiring the square root of $\sigma_i$ in these definitions to keep the units straight.) We use these cluster separations to find the minimum spanning tree, $\mathcal{T}$. There are two major methods for finding a minimal spanning tree: Kruskal's algorithm and Prim's algorithm [18]. We use Prim's algorithm because it is easier to implement on a computer, see Appendix C.

The lacunarity measure should clearly be unitless. We suggest normalizing by dividing the distances obtained in the optimal spanning tree, $\mathcal{T}$, by the maximum average scatter within the full data set. This is calculated using the average scatter matrix for the full data set:

$$c_\star = \frac{1}{N}\sum_{i=1}^{N} x_k, \qquad \hat{\mathcal{M}}^\star = \frac{1}{N}\sum_{i=1}^{N}(x_k - c_\star)(x_k - c_\star)^T = W^\star \sum^\star G^{\star \mathrm{H}}, \tag{41}$$

$$\hat{\mathcal{M}}^\star = W^\star \begin{bmatrix} \sigma_1^\star & 0 & \cdots & 0 \\ 0 & \sigma_2^\star & & 0 \\ & & \vdots & \\ 0 & 0 & \cdots & \sigma_L^\star \end{bmatrix} G^{\star \mathrm{H}} = \sum_{l=1}^{L} \sigma_l^\star w_l^\star g_l^{\star \mathrm{H}}, \tag{42}$$

$$\sigma^\star = \max_{l=1,2,\ldots,L} \sigma_l^\star, \tag{43}$$

$$\hat{\mathcal{T}} = \frac{1}{\sqrt{\sigma^\star}}\mathcal{T} = \left\{ \hat{S}^i = \frac{1}{\sqrt{\sigma^\star}} S^i | i \in [1, 2, \ldots, \mathcal{C} - 1] \right\}. \tag{44}$$

Once the optimal spanning tree has been normalized, we make two unitless lacunarity definitions that correspond to our definitions in Section 3. The estimate of maximum lacunarity, see Eq. (27), is defined as:

$$\hat{\mathcal{L}}_{\max} = \max_{\hat{S}^i \in \hat{\mathcal{T}}} \hat{S}^i. \tag{45}$$

Likewise, the total lacunarity estimate, see Eq. (28), is defined as the sum of the finite number of separations:

$$\hat{\mathcal{L}}_{\text{total}} = \sum_{i=1}^{\mathcal{C}-1} \hat{S}^i, \quad \text{where } \hat{S}^i \in \hat{\mathcal{T}}. \tag{46}$$

## 6. Comparison with analytic results

In this section we compare an analytic lacunarity value that is calculated using the optimal cover-based method to an estimated lacunarity value that is obtained via our Fuzzy-C means based cover lacunarity estimator. The analytical expressions for finding the total spanning tree lacunarity, $\bar{\mathcal{L}}_{\text{total}}(l)$, given an optimal hyper-box cover,

can be found by exploiting the structural knowledge of a symmetric $L$-dimension Cantor set. The presentation that follows uses the same structure as was used in Section 2 where the $L$-dimension Cantor sets were constructed.

We start by first defining the number of groups, $N_G$, within each successive scale for an $L$-dimension Cantor set written as:

$$N_G = \left[ K - \left\lfloor \frac{K}{2} \right\rfloor \right]^L, \tag{47}$$

where $K$ is the scaling factor of the Cantor set and $\lfloor \bullet \rfloor$ represents a truncation operator, i.e. $\lfloor 3.9 \rfloor = 3$. Using the basic number of groupings in connection with the basic hyper-box structure of the symmetric Cantor set, the sum of the minimum spanning tree for the first scale, using optimal hyper-box covering elements, is:

$$\mathcal{S}(1) = \frac{N_G - 1}{K}. \tag{48}$$

The numerator, $N_G - 1$, represents the number of spanning segments required to connect the covering elements, while the denominator, $K$, represents the scaling factor for each copy of the set on the first self-similar scale. This scaling factor defines the maximum gap within all spans for all scales, which is by definition $\mathcal{L}_{\max}$:

$$\bar{\mathcal{L}}_{\max} = \frac{1}{K}. \tag{49}$$

As additional spans are added for each descending scale, the span of $\mathcal{S}(1)$ is divided by the scaling factor and multiplied by the number of copies made. This allows definition of the span for any given scale, $l$:

$$\mathcal{S}(l) = \frac{N_G - 1}{K} \left( \frac{N_G}{K} \right)^{l-1} \quad \text{for } l \geq 1. \tag{50}$$

Using these definitions, the total spanning tree for any scale of interest is the sum over all of the prior scales as well as the current scale:

$$\bar{\mathcal{L}}_{\text{total}}(l) = \frac{N_G - 1}{K} \sum_{i=1}^{l} \left( \frac{N_G}{K} \right)^{i-1}. \tag{51}$$

The closed form solution for this sum [22] can be written as:

$$\bar{\mathcal{L}}_{\text{total}}(l) = \frac{N_G - 1}{N_G - K} \left[ \left( \frac{N_G}{K} \right)^{l} - 1 \right]. \tag{52}$$

As a final note, the total number of covering elements used in the optimal covering of $\bar{\mathcal{L}}_{\text{total}}(l)$ is $\mathcal{C}(l) = N_G^l$. Using the above expressions, a direct comparison between the analytic result, $\bar{\mathcal{L}}_{\text{total}}(\mathcal{C}(l))$, and the result of the estimator, $\hat{\mathcal{L}}_{\text{total}}(\mathcal{C}(l))$, can be made, see Fig. 12. It is clear from the figure that the estimator's results are very good. Errors are, in part, due to a mismatch in the optimal covering element shape, i.e. hyper-boxes vs. hyper-ellipses, and to the mismatch in the exact number of covering elements used in the estimation.

A similar comparison of the maximum lacunarity analytic result, $\bar{\mathcal{L}}_{\max}(\mathcal{C}(l))$, with the estimator's result, $\hat{\mathcal{L}}_{\max}(\mathcal{C}(l))$, is shown in Fig. 13. In this case, the estimator's result asymptotically approaches the analytic result as the number of covering elements increases. This occurs because the error introduced by the shape mismatch of the hyper-ellipses with the optimal shape, a hyper-box in this case, decreases as the hyper-ellipses decrease in size.

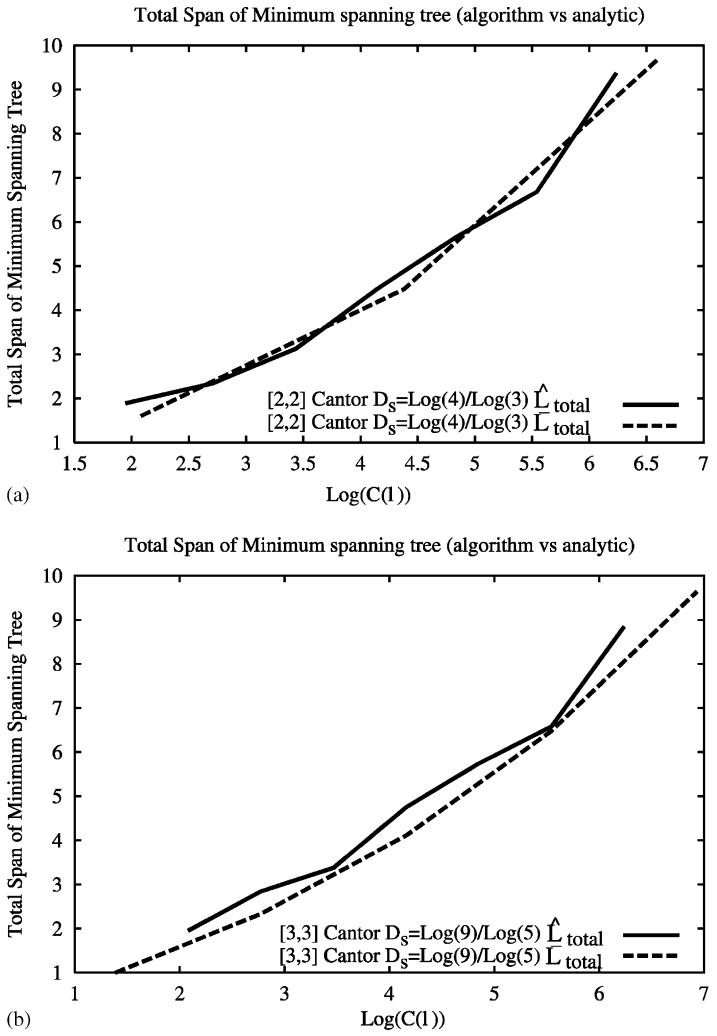



Fig. 12. $\mathcal{L}_{\text{total}}$ over changing number of covering elements for both the analytic optimal hyper-box cover and the estimator's cover for two Cantor sets presented in Fig. 4.

## 7. General results

Our new lacunarity estimation algorithms are accurate with respect to our newly defined measure, as we showed in the previous section. The question remaining is whether or not they achieve our goal of consistently estimating the *gaps* within a data set. In this section, we test the algorithms on the sets defined in Table 1, as well as a series of two- and three-dimensional Cantor sets, and demonstrate that these new lacunarity measures do indeed achieve our goal. These examples also show that the new measures are both scalable and independent of a data set's numerical size and mass. Even with these achievements, the lacunarity measures presented here do contain limitations, discussed in Section 8. Nevertheless, the lacunarity measures presented here work well for ramified data sets, e.g. Cantor sets.

In describing the need for a new measure of lacunarity, see Section 2, we presented a basic series of test sets that demonstrates many of the problems within the $\mathcal{L}_{\text{GB}}$ measure, see Table 1 and Fig. 3. These problems are resolved

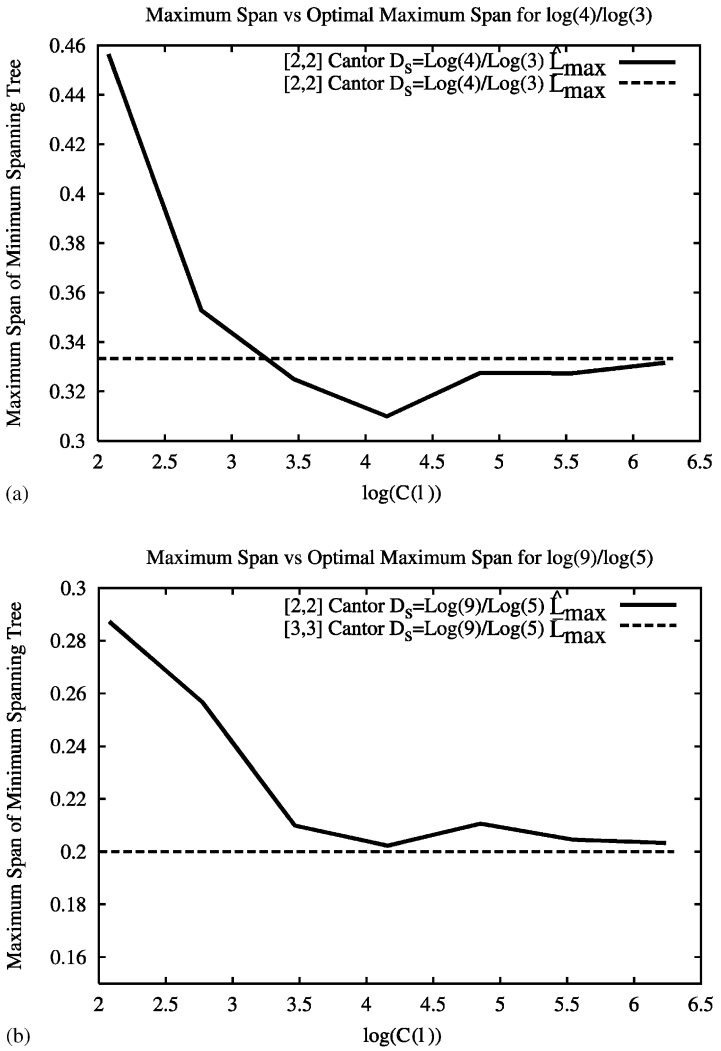



Fig. 13. $\mathcal{L}_{\max}$ over changing number of covering elements for both the analytic optimal hyper-box cover and the estimator's Fuzzy-C means cover for two Cantor sets presented in Fig. 4.

with the $\mathcal{L}_{\max}$ measure, see Fig. 14. Notice that each of these sets has near identical $\hat{\mathcal{L}}_{\max}$ results. There is no apparent variation due to mass change or data shifting within the analysis space, in contrast to $\mathcal{L}_{\mathrm{GB}}$ (Fig. 3). Fig. 15 shows the marked contrast in sensitivity to mass for $\mathcal{L}_{\mathrm{GB}}$ vs. $\hat{\mathcal{L}}_{\mathrm{total}}$. That is, $\hat{\mathcal{L}}_{\mathrm{total}}$ provides a consistent measure for varying resolution of the same data set, where $\mathcal{L}_{\mathrm{GB}}$ fails. In a large way, it is this sensitivity to mass change that limits the usefulness of glide box analysis for many applications, such as biofilm quantification. However, with our new measures, this problem has been resolved.

As a second example, consider the effect on estimated lacunarity value of changing the lacunarity while maintaining the fractal dimension. The $\hat{\mathcal{L}}_{\max}$ and $\hat{\mathcal{L}}_{\mathrm{total}}$ lacunarity estimates for the Cantor sets developed in Section 2 and shown in Fig. 4 are presented in Fig. 16.

For Cantor set families where the number of copies only increases in one dimension, $\hat{\mathcal{L}}_{\max}$ converges to the same value as the number of covers increases. For examples, see Cantor set families $D_s = \log(4)/\log(3)$ where

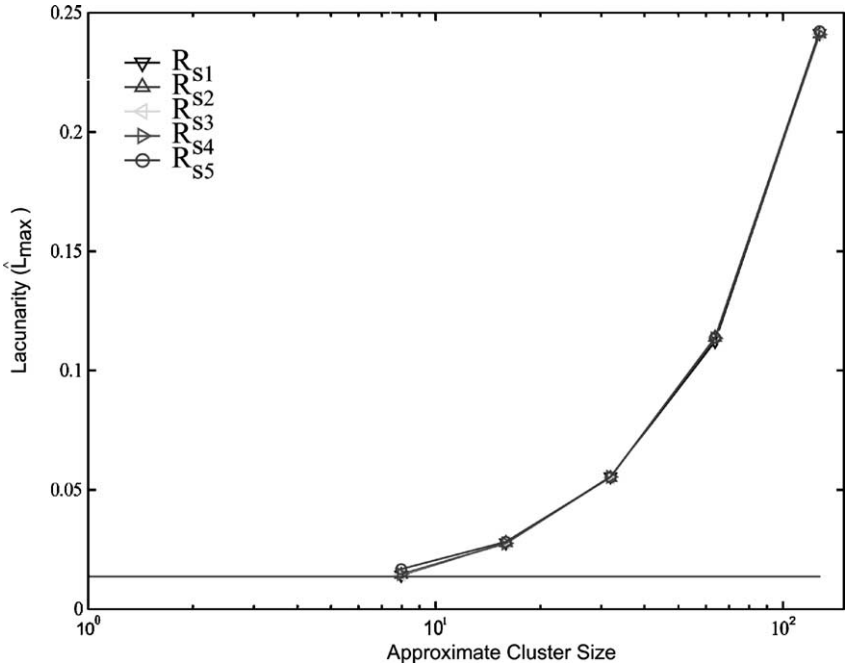



Fig. 14. $\hat{\mathcal{L}}_{\max}$ lacunarity estimations for gliding box algorithm test sets defined in Table 1.

$N_{c,d} = [2, 2], [2, 3], [3, 2]$ and $D_s = \log(9)/\log(5)$ where $N_{c,d} = [3, 3], [3, 4], [4, 3]$. This is expected since only the *gaps* in the one dimension were changed, leaving the dimension with larger *gaps* still contained within the minimum spanning tree. Of course, as the *gaps* in both dimensions are changed, the estimate $\hat{\mathcal{L}}_{\max}$ changes, as shown in Fig. 16a. Another result is that Cantor sets with similar construction in different families (such as $D_s = \log(4)/\log(3)$, $N_{c,d} = [3, 3]$ and $D_s = \log(9)/\log(5)$, $N_{c,d} = [3, 3]$) have more similar $\hat{\mathcal{L}}_{\max}$ than do individual members of a Cantor set family with dissimilar construction (such as $D_s = \log(4)/\log(3)$, $N_{c,d} = [3, 3]$

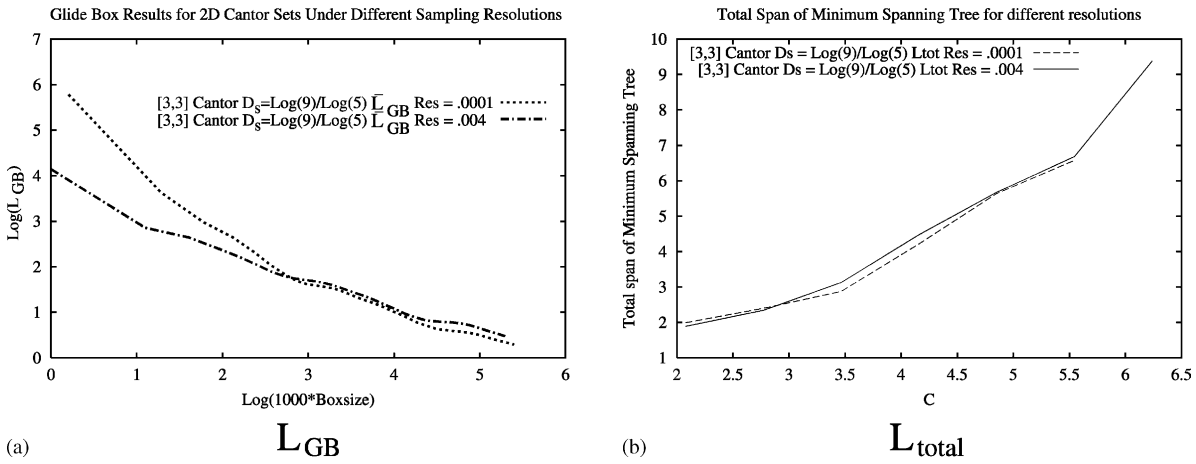



Fig. 15. Comparison of $\mathcal{L}_{GB}$ and $\hat{\mathcal{L}}_{total}$ estimates for two sample resolutions, 0.004 and 0.0001, of the [3, 3] Cantor set with similarity dimension $\log(9)/\log(5)$.

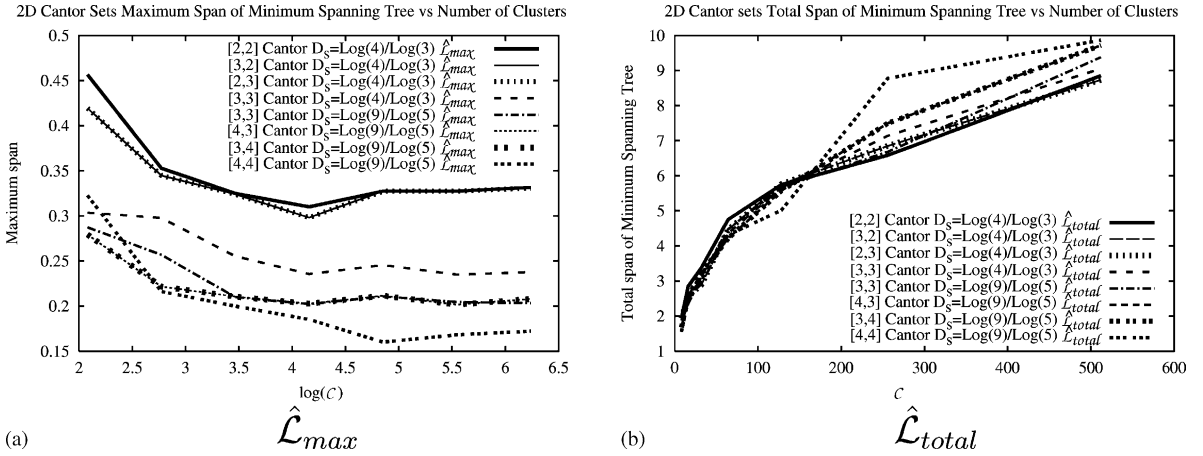



Fig. 16. Lacunarity estimates for two-dimensional Cantor sets shown in Fig. 4a–h.

and $D_s = \log(4)/\log(3)$, $N_{c,d} = [2, 2]$), see Fig. 4. We, therefore, conclude that the fractal dimension and lacunarity, $\hat{\mathcal{L}}_{\max}$, together more fully describe these data sets.

The $\hat{\mathcal{L}}_{\text{total}}$ estimate is highly sensitive to changes within the spanning tree, i.e. as *gap* sizes are changed the estimate changes, see Fig. 16b. Due to this sensitivity, $\hat{\mathcal{L}}_{\text{total}}$ provides additional structural description for varying scales within the data set. It is important to note that both estimates are insensitive to rotational changes of the *gaps*, e.g. $N_{c,d} = [3, 2]$ and $[2, 3]$, see Fig. 4c and d.

To show the $n$-dimensional scaling ability of $\hat{\mathcal{L}}_{\max}$ and $\hat{\mathcal{L}}_{\text{total}}$, we present results for a family of three-dimensional Cantor sets with similarity dimension of $\log(8)/\log(3)$ in Fig. 17. (A visualization of the basic [2, 2, 2] Cantor set was shown in Fig. 5.) The lacunarity estimates for the three-dimensional Cantor sets follow the same pattern as the estimates for the two-dimensional sets. Finally, we provide an example of a sub-optimal cover and its minimum spanning tree for a two-dimensional [2, 2] Cantor set with similarity dimension $\log(4)/\log(3)$ with 128 covering elements in Fig. 18.

The new lacunarity measures we propose are akin to the average point-to-point distance measure of lacunarity discussed by Mandelbrot [1] and Taguchi [23]; however, our measures are valid for sets consisting of infinite points

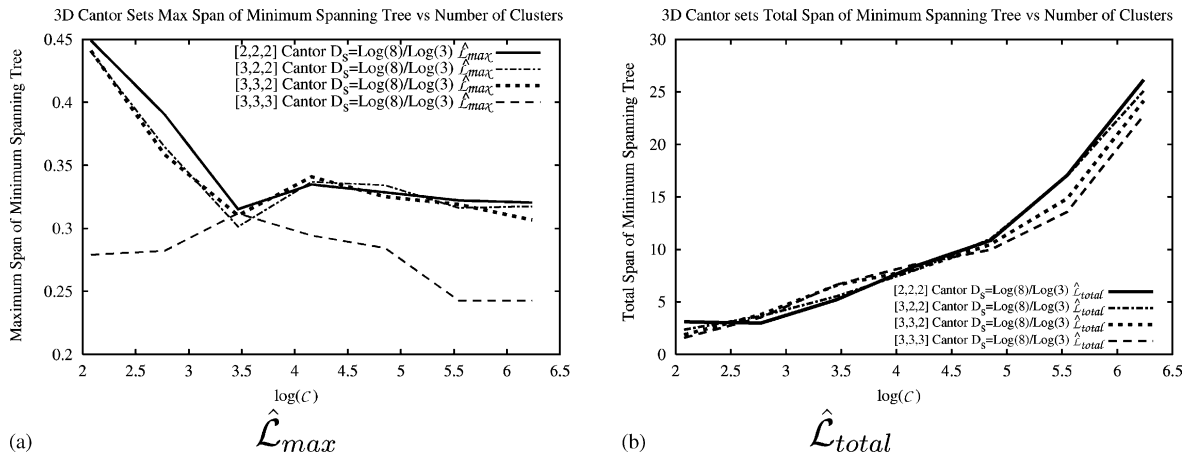



Fig. 17. Lacunarity estimates for three-dimensional Cantor sets with similarity dimension of $\log(8)/\log(3)$.

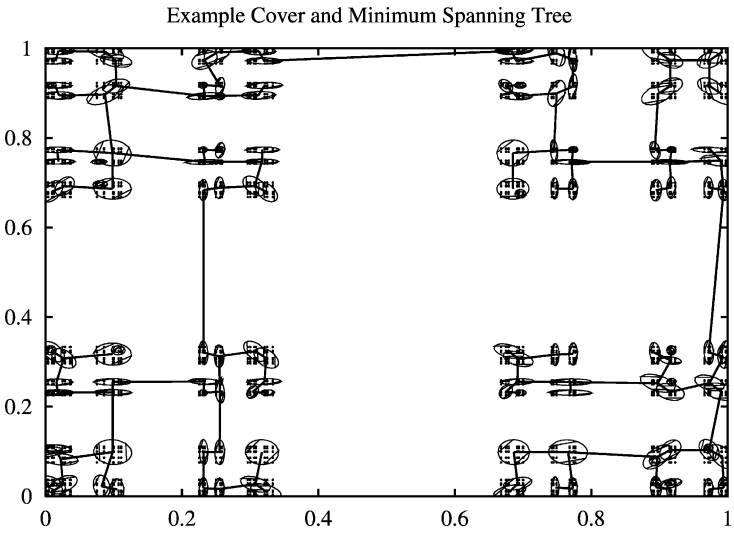



Fig. 18. Sub-optimal cover with its minimum spanning tree for a two-dimensional [2, 2] Cantor set with the similarity dimension log(4)/log(3) and a 128-element cover. The uncovered set is shown in Fig. 4a.

and intervals. A side benefit of using the same cover for calculating both the fractal dimension [7] and lacunarity is the reduction in computation through reuse of the covers in both methods.

## 8. Future directions

One of the drawbacks of the lacunarity measures presented in this paper is that they are for ramified data sets. As the data points become more dense, the covering elements start to pack tightly together over well-connected spaces and the minimum spanning trees no longer measure the gaps in the data sets, see Fig. 19. In fact, the $\mathcal{L}_{\max}$ and $\mathcal{L}_{\text{total}}$

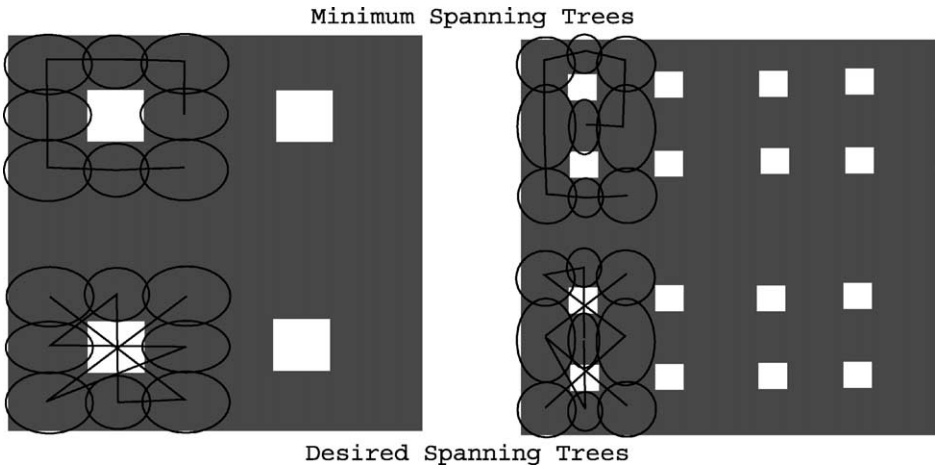



Fig. 19. Two sets that violate the ramified data set assumption with (upper) the spanning tree generated by existing methods and (lower) a new spanning tree that solves this violation.

lacunarity measures go to zero as the ramified assumption is violated. Even so, these methods can be modified to use a different type of spanning tree that still does measure the desired gaps within the data sets. An example of such a tree is shown in Fig. 19. This improved algorithm for dense sets is currently being developed and will be the basis for a future paper.

**Acknowledgements**

We thank Dr. David Gorsich for his careful review of the initial manuscript. In addition, we thank Mrs. Myrtle Siefken for her comments and proof reading of the paper before and during its submittal process. We also thank Dr. Daphne L. Stoner for her insight into the use of this technology within the biological sciences, and her presence as a key member of our applications research team. Finally, we thank the reviewers for their enlightening comments which strengthened this paper. This work was supported through the INEEL Laboratory Directed Research and Development Program under DOE Idaho Operations Office Contract DE-AC07-99ID13727.

## Appendix A. List of variables

| | |
|---|---|
| $a_k^n$ | $k$th interval within the $n$th scale Cantor generation set |
| $A$ | data set or image to be analyzed |
| $b_k^n$ | left endpoint for the interval $a_k^n$ |
| $C_i$ | cover of the $i$th cluster |
| $\mathcal{C}, \mathcal{C}_d$ | number of covering elements for a given size; $d$, of covering element |
| $C^\star$ | the union of $C_i$ for all $i$ in the optimal set of covers that minimize the spanning tree |
| $d$ | size of a covering element |
| $d_{\mathrm{e}}(x, y)$ | Euclidean distance function between $x$ and $y$ |
| $\mathrm{diam}(C)$ | diameter of the covering element $C$ |
| $D_b(A)$ | box dimension definition |
| $D_h(A)$ | Hausdorff–Besicovitch (HB) dimension definition |
| $D_s(A)$ | similarity dimension |
| $g_l^i$ | $l$th row vector of $G^i$ |
| $G^i$ | directional unitary matrix on the right in the singular value decomposition |
| $G^n$ | Cantor set gaps for the $n$th level generation set |
| $h^s$ | $s$-dimensional Hausdroff measure |
| $h^s(A)$ | Hausdroff measure of $A$ |
| $i$, $j$, $k$, $l$, $n$, and $s$ | used as counting indices throughout |
| $I$ | iteration through the Picard algorithm |
| $K$ | scaling factor used in generating Cantor sets |
| $L$ | number of Cantor set tuples within the generalized Cantor set (dimension); dimension of the image or signal to be analyzed ($A$) |
| $\hat{\mathcal{M}}^i$ | scatter matrix for the $i$th cluster |
| $N$ | number of data points with the image of interest, $A$ |
| $N_{\mathrm{c}}$ | number of copies replicated on each successive level within a Cantor set |
| $N_{\mathrm{G}}$ | number of groups within an analysis scale |
| $N_i$ | number of hard clustered points within the $i$th cluster |

| | |
|---|---|
| $p_i^j$ | maximum length of an ellipse axis, or the $j$th cluster projection on the a segment connecting the centers of the $j$th and $i$th clusters |
| $\mathcal{Q}^i$ | matrix of singular values |
| $r$ | unit box size, i.e. the length of one box side |
| $\Re^p$ | $p$-dimensional real vector space |
| $s_{mn}$ | spanning distance between the $m$th and $n$th clusters |
| $S_k$ | $k$th segment of the minimum spanning tree |
| $\mathcal{S}$ | span of a cover, i.e. $\sum^k S_k$ |
| $\mathcal{T}$ | minimum spanning tree of segments connecting covers |
| $u_{ik}$ | membership value for the $k$th point in the $i$th cluster |
| $\mathbf{v}_k$ | vector describing the center of the $k$th cluster |
| $V_n$ | $n$th level generation set used in construction of a generalized Cantor set |
| $\mathcal{V}_i$ | the $i$th tuple of $\mathcal{V}^d$ |
| $\mathcal{V}^d$ | generalized $d$th-tuple Cantor set |
| $w_l^i$ | $l$th column vector of $W^i$ |
| $W^i$ | directional unitary matrix on the left in the singular value decomposition |
| $x_i$ | a component of a vector $x$ |
| $\mathbf{x}_i$ | vector for the $i$th data point |
| $\epsilon$ | maximum change in membership allowed for conversion of the clustering methods |
| $\sigma_l^i$ | $l$th singular value of $\mathcal{Q}^i$ |

## Appendix B. Sub-optimal cover: Fuzzy-C means Picard iteration

(1) Choose the number of clusters $\mathcal{C} \geq 1$. Select $\epsilon > 0$ (this is the ending condition). Next, randomly initialize the centers of the clusters, $c_i$.
(2) Solve for the memberships of each vector:

$$u_{ik} = \frac{1}{\sum_{j=1}^{\mathcal{C}} [\|x_k - c_i\| / \|x_k - c_j\|]^2}, \quad \text{where } k = 1, 2, \ldots, N, \quad i = 1, 2, \ldots, \mathcal{C}. \tag{B.1}$$

(3) Solve for the centers for each cluster:

$$c_i = \frac{\sum_{k=1}^{N} u_{ik}^2 x_k}{\sum_{j=1}^{N} u_{ij}^2}, \quad \text{where } i = 1, 2, \ldots, \mathcal{C}. \tag{B.2}$$

(4) Repeat step 2.
(5) Repeat step 3.
(6) If $\Delta u_{ik} > \epsilon$, then loop back to step 4. Otherwise, stop.

Note that by using the Euclidean norm, the algorithm produces a minimum distance classification of the data, which is what we desire in this work.

## Appendix C. Minimum spanning tree: the Prim algorithm

The algorithm given here is not designed for speed, only ease of comprehension.
Prim algorithm for minimal spanning trees:

(1) Create some bookkeeping sets as follows (show the form and the initial values):

$$\text{used} = \{i | i \in [1, 2, \ldots, \mathcal{C}]\} = \{\},\qquad \text{unused} = \{i | i \in [1, 2, \ldots, \mathcal{C}]\},$$
$$\text{links} = \{(m, n)_i | m \neq n;\ \ m, n \in [1, 2, \ldots, \mathcal{C}];\ \ i \in [1, 2, \ldots, \mathcal{C} - 1]\} = \{\},$$
$$\mathcal{S} = \{S^i | S^i = s_{mn};\ \ m \neq n;\ \ m, n \in [1, 2, \ldots, \mathcal{C}];\ \ i \in [1, 2, \ldots, \mathcal{C} - 1]\} = \{\}.$$

(2) Find the shortest connecting cluster segment and record the link:

$$S^1 = s_{mn} = \min_{m,n \in \text{unused}} s_{mn},\qquad \text{link} = \text{link} \bigcup \{(m, n)_1\},\qquad \text{used} = \text{used} \bigcup \{m, n\},$$
$$\text{unused} = \text{unused} \setminus \{m, n\},\qquad \mathcal{S} = \mathcal{S} \bigcup S^1.$$

(If there is more than one, choose any one.)

(3) Find the remaining shortest attaching links:

For $i = 2$ to $\mathcal{C} - 1$

$$S^i = s_{mn} = \min_{m \in \text{used}\ n \in \text{unused}} s_{mn},\qquad \text{link} = \text{link} \bigcup \{(m, n)_i\},\qquad \text{used} = \text{used} \bigcup \{n\},$$
$$\text{unused} = \text{unused} \setminus \{n\},\qquad \mathcal{S} = \mathcal{S} \bigcup S^i.$$

(4) $\mathcal{T} = \mathcal{S}$.

Note the use of $\setminus$ denotes set deletion.

## References


[1] B.B. Mandelbrot, The Fractal Geometry of Nature: Updated and Augmented, Freeman, New York, 1983.
[2] P.S. Amy, D.L. Haldeman, in: F.J. Brockman, C.J. Murray (Eds.), The Microbiology of the Terrestrial Deep Subsurface: Microbiological Heterogeneity in the Terrestrial Subsurface and Approaches for its Description, Lewis Publishers, New York, 1997.
[3] N.A.C. Cressie, Statistics for Spatial Data, Revised edition, Wiley, New York, 1993.
[4] D. Gorsich, C.R. Tolle, G. Gerhart, Wavelet and fractal analysis of ground vehicle images, in: Proceedings of the Eighth Annual Ground Vehicle Survivability Symposium, 1997.
[5] D. Gorsich, C.R. Tolle, R. Karlsen, G. Gerhart, Wavelet and fractal analysis of ground vehicle images, in: Proceedings of the SPIE Symposium, 1996.
[6] C.R. Tolle, D. Gorsich, Sub-optimal covers for measuring fractal dimension, in: Proceedings of the Rocky Mountain NASA Space Grant Consortium Conference, University of Utah, 1996.
[7] C.R. Tolle, T.R. McJunkin, D.J. Gorsich, Sub-optimal MCV cover based method for measuring fractal dimension, IEEE Trans. Pattern Anal. Mach. Intell. 25 (1) (2003) 32–41.
[8] D. Gorsich, C.R. Tolle, R. Karlsen, G. Gerhart, Wavelet and fractal analysis of ground vehicle signatures, in: Proceedings of the Seventh Annual Ground Vehicle Survivability Symposium, 1996.
[9] Z. Lewandowski, D. Webb, M. Hamilton, G. Harkin, Quantifying biofilm structure, Wat. Sci. Tech. 39 (7) (1999) 71.
[10] T.G. Smith Jr., G.D. Lange, W.B. Marks, Fractal methods and results in cellular morphology—dimensions, lacunarity, and multifractals, J. Neurosci. 69 (1996) 123–136.
[11] S. Möller, D.R. Korber, G.M. Wolffaardt, S. Molin, D.E. Caldwell, Impact of nutrient composition on a degradative biofilm community, Appl. Environ. Microbiol. 63 (1997) 2432–2438.
[12] D.S. Ebert (Ed.), Texturing and Modeling: A Procedural Approach, Academic Press, Cambridge, MA, 1994.
[13] R.L. Cannon, J.V. Dave, J.C. Bezdek, Efficient implementation of Fuzzy C-means clustering algorithms, IEEE Trans. Pattern Anal. Mach. Intell. PAMI-8 (2) (1986) 248.
[14] R.E. Plotnick, R.H. Gradner, W.W. Hargrove, K. Prestegaard, M. Perlmutter, Lacunarity analysis: a general technique for the analysis of spatial patterns, Phys. Rev. E 53 (5) (1996) 5461.
[15] C. Allain, M. Cloitre, Characterizing the lacunarity of random and deterministic fractals sets, Phys. Rev. A 44 (6) (1991) 3552.
[16] P. Meakin, Fractals, Scaling and Growth Far from Equilibrium, Cambridge University Press, New York, 1998.

[17] H.-O. Peitgen, H. Jürgens, D. Saupe, Chaos and Fractals: New Frontiers of Science, Springer, New York, 1992.
[18] A. Tucker, Applied Combination, 3rd ed., Wiley, New York, 1995.
[19] P.V. O'Neil, Advanced Engineering Mathematics, Wadsworth, Belmont, CA, 1987.
[20] P. Hartman, Ordinary Differential Equations, Wiley, New York, 1964.
[21] R.O. Duda, P.E. Hart, Pattern Classification and Scene Analysis, Wiley, New York, 1973.
[22] J.J. Tuma, Handbook of Numerical Calculations in Engineering, McGraw-Hill, New York, 1989.
[23] Y. Taguchi, Lacunarity and universality, J. Phys. A (20) (1987) 6611.