

Quantitative Analysis of Biofilm Images Using Fractal Dimensions

A Thesis
Presented for the
Master of Science
Degree
The University of Memphis

Zhou Ji
August 2000

STATEMENT OF PERMISSION TO USE

In presenting this thesis in partial fulfillment of the requirements for a Master's degree at The University of Memphis, I agree that the Library shall make it available to borrowers under rules of the Library. Brief quotations from this thesis are allowable without special permission, provided that accurate acknowledgement of the source is made.

Permission for extensive quotation from or reproduction of this thesis may be granted by my major professor, or in his absence, by the Head of Interlibrary Services when, in the opinion of either, the proposed use of the material is for scholarly purposes. Any copying or use of the material in this thesis for financial gain shall not be allowed without my written permission.

Signature_____

Date_____

Acknowledgements

I would like to acknowledge the great help from my major advisor, Dr. Giri Narasimhan during the work of this thesis. His positive influence from either academic or personal aspect is much beyond this paper and will endure in the rest of my life. I also want to thank all the support from the Computer Science Division, the Department of Mathematical Science, and the University of Memphis.

Abstract

Ji, Zhou. MS. The University of Memphis. August, 2000. Quantitative Analysis of Biofilm Images Using Fractal Dimensions. Major Professor: Giri Narasimhan, Ph.D.

As one of the important ways to quantify biofilm images, fractal dimensions of pixel-based images are studied. Several algorithms and their implementations are discussed in detail through the introduction of software BIP. An image generator called KochGen is developed to make standard fractal images with known fractal dimensions. It can generate diverse images in TIFF or BMP format. A large set of such standard images is generated to test fractal dimension analysis and used to evaluate the 11 algorithms implemented in BIP.

Table of Contents

1. Introduction	1
1.1 Biofilm research	1
1.2 Fractal and fractal dimension	3
1.3 Demonstration of Julia Sets.....	5
2. Fractal Dimensions of Pixel Image and Biofilm Image Processing.....	7
2.1 Fractal dimensions of pixel image	7
2.2 BIP and its algorithms	8
3. Generating Images with Known Fractal Dimension	17
3.1 Purpose of generating standard images	17
3.2 Fractal generator KochGen	17
3.3 Discussion of implementation.....	21
4. Computational Experiments.....	23
4.1 Experiments on biofilm images.....	23
4.2 Experiments on generated images.....	23
5. Simulation of Biofilm Growth.....	36
6. Conclusion.....	39
Works Cited.....	41

1. Introduction

1.1 Biofilm Research

In most natural, clinical and industrial settings, bacteria often grow attached to surfaces in communities; this form of growth is known as biofilm. Biofilm-associated organisms are able to adapt to environmental changes by altering their gene expression and general performance, of which increased resistance to antibiotics is an example (Costerton et al. 1999). One of the ways microbial communities adjust to environmental change is by changing the structural organization of the biofilm. However, the underlying mechanisms behind initiation of biofilm formation and development of structural organization are only just beginning to be observed (Costerton et al. 1999).

Biofilms occur in a large variety of physical situations - often in an aquatic environment. These occurrences are very important. Biofilms can lead to an increase in friction and heat transfer resistance. They can cause material deterioration through microbial corrosion, and they can contaminate artificial organs and catheters. They are also an important factor to be dealt with in wastewater treatment. The side and bottom of most water navigation vessels are covered by biofilms. Dental plaque is yet another example of biofilms. The understanding of biofilm therefore is an object of a large and increasing amount of research activity, both in the form of biological experiments and in the form of numerical modeling (Wimpenny 1997, Picioreanu 1998) and analysis.

One aspect of this research has focused on the structure of biofilms. An important goal of either experimentation or modeling is to find out what factors

influence the structure of biofilms and how. Some believe that the morphology or structure of biofilms is mainly connected to the transport of nutrients to the surface of the biofilm as well as to the interior of the colonies on the biofilm. An alternative view is that some kind of inter-cellular communication takes place that controls the structure of the biofilm. (Meyer 1999). Quantitative description of biofilm structure thus becomes the fundamental problem (Lewandowski et al. 1999, Yang et al. 2000). Various attempts of analyses include computing parameters such as thickness, roughness, density, specific surface area, mean pore radius, textural entropy, area porosity, fractal dimension and maximum diffusion distance. The last four variables were shown to reach steady state as the biofilm thickness reached steady state (Lewandowski et al 1999).

Fractal dimension is the topic of this thesis. The objects analyzed were thousands of biofilm images taken by Arne Heydorn and Søren Molin at the Danish

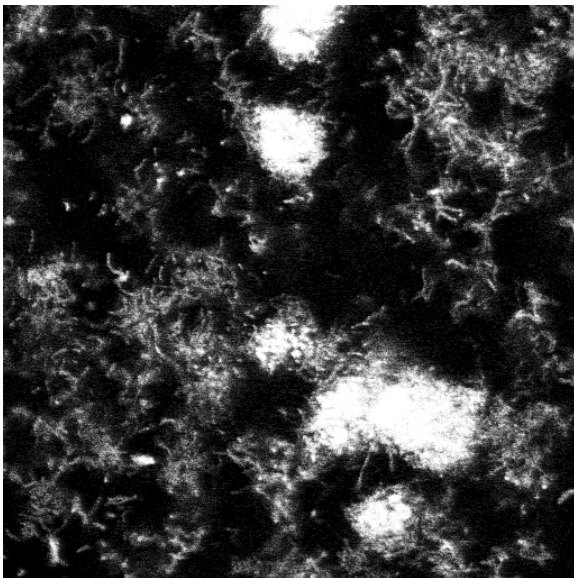


Figure 1. Photograph of Biofilm

Technical University. Four strains of a bacterium known as *Pseudomonas aeruginosa* were analyzed. Each strain was grown in two separate channels and from each channel, images were acquired at different time points (24h, 48h, 96h, 168h, 240h after inoculation). Fig. 1 shows a typical image. The goal of this thesis is to study and establish any possible relationship between the

fractal dimension of images and the strains of bacteria and other experimental parameters. Also, because of the large number of images, it is important to automate the quantification and analysis.

1.2 Fractal and fractal dimension

The concept of a **fractal** was developed by the Franco-Polish mathematician Benoit B. Mandelbrot. He used the term to refer to structures that could not be described by the usual geometric shapes such as lines, surface, or solids.

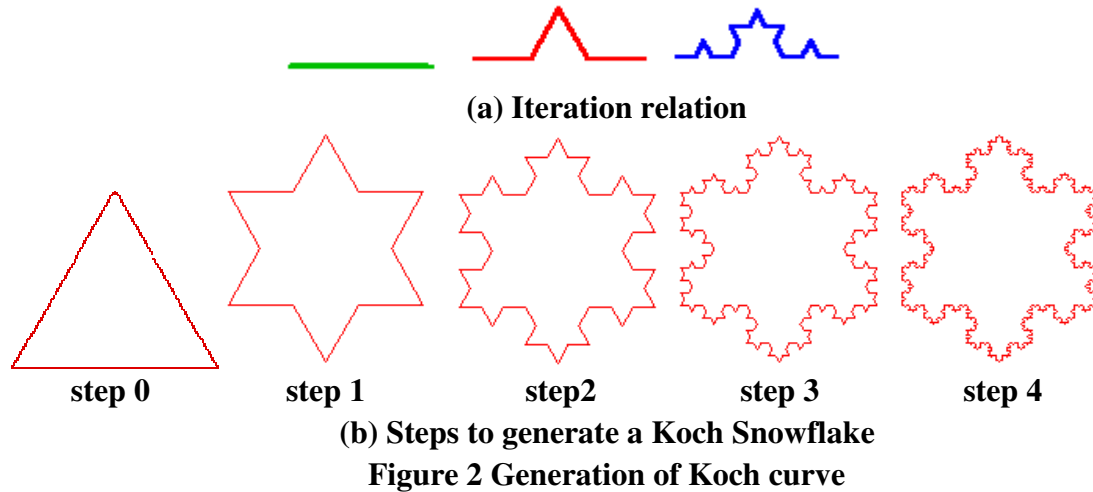
Around 1983, a research group in dynamical systems was founded at Bremen University. In 1984, results from their own computer graphics laboratory were made public and caused a great sensation – what they exhibited were beautiful, colored computer graphics reminiscent of artistic paintings. Since then these images of fractals have appeared almost everywhere. It led to a paradigm change from the scientific viewpoint and resulted in a new theory, the so-called chaos theory, which has shattered the scientific worldview. New techniques are changing the traditional methods of work of mathematics and led to the field of experimental mathematics (Becker et al. 1989).

One example of a fractal image is that of the famous Koch snowflake (see Fig. 2). While the curve may seem ragged, it is of interest. The Koch snowflake is neither a line nor a surface. If you consider the Koch snowflake after an infinite number of iterations, unlike say a circle, its length cannot be determined. The more we magnify – that is, the more closely we look, the longer the boundary becomes. In other words, it is infinitely long and has zero width. Two closely related properties are characteristics of all fractals: 1) self-similarity, that is, in each tiny piece we observe

the form of the entire shape; 2) irregularity, that is, there are no smooth boundaries – lengths or areas cannot be determined. Once this concept had been drawn to the attention of researchers, they sought many examples of it in nature. The coastline of an island is a fractal. It is possible to read off the length of a coastline from a map with a given scale. But if we used a map with a different scale, that can change the result. The smaller the scale of the map, the longer the coastline seems to be. By the same principle, many natural boundaries are fractal. The consequence of this discovery for biology was explained previously (Becker et al. 1989).

The line is a 1-dimensional structure and a surface is a 2-dimensional structure. To quantify the complexity of a fractal, the concept of **fractal dimension** (FD) was introduced. The FD is defined so that FD of a line is always 1 and that of a plane surface is always 2. The FD of a fractal curve is some real number between 1 and 2. Similarly a fractal surface has fractal dimension between 2 and 3. If we plot the measured length L of a fractal curve versus its stride length u on a log-log scale, it can be observed that the slope is constant. This is true for a variety of borders, both natural and man-made (e.g. coastline) (Russ 1992). Such a plot is called Richardson plot. If the slope of this plot is denoted by m , the fractal dimension can be defined as $D = 1 - m$. To understand it more precisely, we consider the Koch curve, which is generated by the recursive relation shown in Fig. 2. The Koch curve is the result after an infinite number of iterations (the first few steps are shown in Fig. 2(b)). In each step of the recursion, the middle third of each line segment is omitted and replaced by two segments forming a tent-like angle (Peitgen et al. 1992). In each iteration a line segment is actually divided into $N = 4$ equal pieces, each scaled down by a factor

$\sigma=1/3$. The formula of fractal dimension becomes $D = \log N / \log(1/\sigma) = \log 4/\log 3 = 1.26186$.



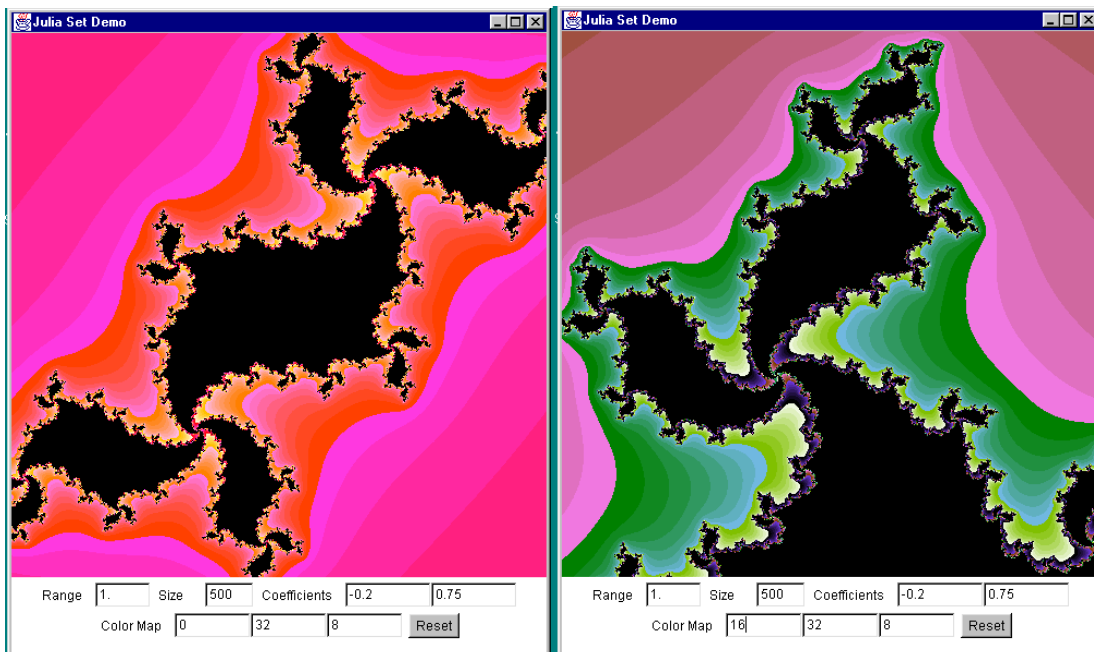
In general, N and r could be numbers other than 4 and 3. More generally, FD can be defined for recursively generated curves as $\log N / \log(1/\sigma)$, where σ is the size of the smallest unit in the definition and N is the number of the smallest units. The above formula provides as a way of generating images with given fractal dimension as discussed in section 4. The algorithms to calculate FD for pixel-based images are also based on the same general formula.

1.3 Demonstration of Julia Sets

A program is developed in Java to visually demonstrate the basic properties of fractals, such as self-similarity and irregularity. We use Julia Sets to illustrate it. The iterative function implemented currently is of the form of $f(z) = z^2 + (c_1 + i c_2)$, where z is the complex number corresponding to a point on the complex plane. Extension to other functions is straightforward. The coefficients c_1 and c_2 can be set from the user interface to generate various patterns. The size (in pixels) of the resulting image, and the coordinate range can also be set.

Color map is set through three coefficients, each controlling one of the RGB components. The red component changes periodically from 0 to 255 when depth changes an interval that is 64 divided by the first coefficient. If the coefficient is zero, red component is fully included (255). The other two coefficients work similarly on green and blue components. The depth at each point z is defined as the level of recursion for the iterative function $f(z)$ to reach the value 100. If it is larger than 64, black color is always used. It works very well to generate beautiful combination of colors. As pointed out by Becker et al (1989), Julia sets and other computer-generated fractals are interesting both for research and for their aesthetic appeal.

The mouse can be used to magnify a selected part of the display. The image scale can be reduced to one millionth of the starting scale, while still showing similar structures. Fig. 3(a) is the display with default initial settings. Fig. 3(b) is a display with a different color map and a much smaller scale.



(a) Default initial setting

(b) A much smaller scale

Figure 3. Demonstration of Julia Sets

2. Fractal Dimension of Pixel Image and Biofilm Image Processing

From the geometric point of view, the fractal dimension of a curve or a surface can be used as measure of its “roughness”. Consequently, it makes sense to study biofilms using this measure. For some applications, it may be sufficient to determine an approximate value of FD by visually comparing the roughness of a real object with that of generated objects having known fractal dimension, (Russ 1992). However, determination of the dimension using a Richardson-type plot is more precise and desirable since it also provides confirmation of the self-similarity of the object’s outline, established by the linearity of the plot.

2.1 Fractal dimensions of pixel images

For a pixel-based image, some distinct problems arise in determining the fractal dimension. Firstly, objects or features (such as the boundary) need to be detected from the image – and the algorithms and parameters used to do so would affect the result in one way or the other. Secondly, a pixel is the lower limit of the scale in which the image is considered. While a true fractal implies an infinitely recursive structure, the slope of the Richardson plot for real objects could be different in a different range of scale. The pixel image only makes a small range of the plot available. Thirdly, the boundary must be considered either as a sequence of pixels or as boundaries between groups of pixels. The former involves choice of 8-connectivity or 4-connectivity. In other words, pixels that share only one corner vertex may or may not be considered adjacent. The latter, i.e., boundary between pixels, is closer to the

original boundary in the sense of no width. However, most algorithms use the sequence of pixels as the description of the boundary.

2.2 BIP and its algorithms

A software package **Biofilm Image Processing** (BIP) was developed to compute fractal dimension of biofilm images. Of course it can be used on images from other areas too when fractal dimension is of interest. While BIP is not an individual work of this author¹, it is the basic tool used in the analyses presented in this thesis.

There are no accepted definitions or algorithms for computing FD, although many have been proposed in the literature. BIP implements 11 different algorithms to calculate fractal dimension from pixel-based images. Nine of them were mainly based on those discussed and implemented by Warfel ([10]). While Warfel's implementations are in the form of macros of a popular image processing software NIH Image, BIP is a standalone software written in Visual C++. It is noteworthy that for computing FD, the specific implementations of an algorithm are less significant, and not likely to be exactly that same in any two. The last two algorithms are new and were designed by BIP's major contributor, Qichang Li.

The processing in BIP includes two stages:

- 1) Preprocessing (shown in Fig. 4):
 - (a) make the original grayscale or color image a binary image;
 - (b) delete small objects from the image;

¹ The major developer of BIP is Qichang Li, another student advised by Dr. Giri Narasimhan. Zhou Ji is honorably listed as the second author.

(c) fill small holes;

(d) find the one-pixel border of the connected objects (the bacterial colonies in this image).

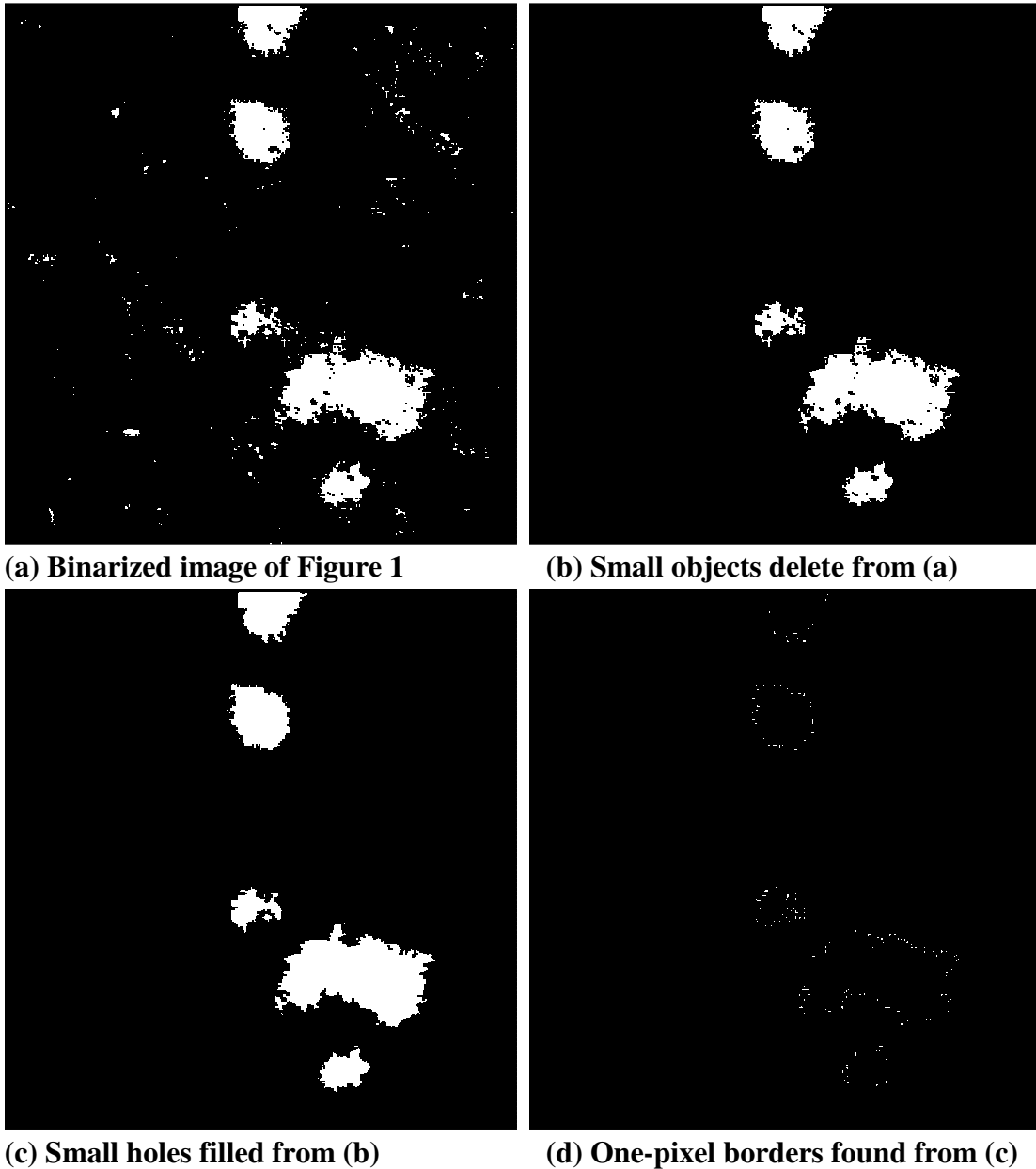


Figure 4. Preprocessing Stage

2) Calculating fractal dimension (with one of the eleven FD algorithms):

(e) calculate two variables iteratively;

(f) plot on log-log scale;

- (g) calculate slope by least square regression;
- (h) determine fractal dimension.

Before applying a FD algorithm, the actual pictures of biofilms are preprocessed with the four necessary steps (a) through (d), as discussed in the previous section: Steps (b) and (c) are applied to eliminate noises that are unlikely to represent inherent geometric properties of the shapes in the pictures.

After preprocessing, one of the eleven algorithms can be used to calculate the fractal dimension of the image. Individual objects (bacteria colonies) in the image are not considered separately because it is assumed that the fractal dimension is related with the global condition of the whole picture. The algorithms are hereafter briefly explained.

1) Dilation Method

This method is also popularly known as the Minkowski Sausage Method. It “dilates” the one-pixel outline of the objects iteratively. Pixels with neighbor(s) on the boundary are added to the boundary area in each step. The logarithm of twice the dilation count (iteration steps) is plotted against the logarithm of the resulting area (count of pixels) divided by twice the dilation count. For fractal images, the plot is a linear relation. The slope m is calculated by least square regression. The fractal dimension D is calculated using the formula $D = 1 - m$. We presented a general description of the algorithm; there could be differences in specific implementation. For example, the dilation used by Warfel is based on the neighbor rule of one pixel. In other words, as long as a pixel has one neighbor in the boundary area, it is added in the next step. BIP on the other hand only adds pixel with neighbor at coordinate

directions, resulting from the method of choosing neighbor in circle and default radius 1.

The red area in Fig. 5 is the resulting area of the outline shown in Fig. 4(d) after several steps of dilation. BIP also has an animation option implemented to show the algorithm as it iterates.

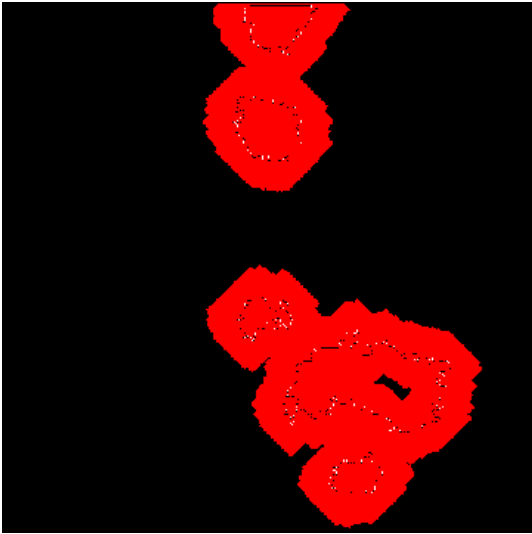


Figure 5. Dilated outline

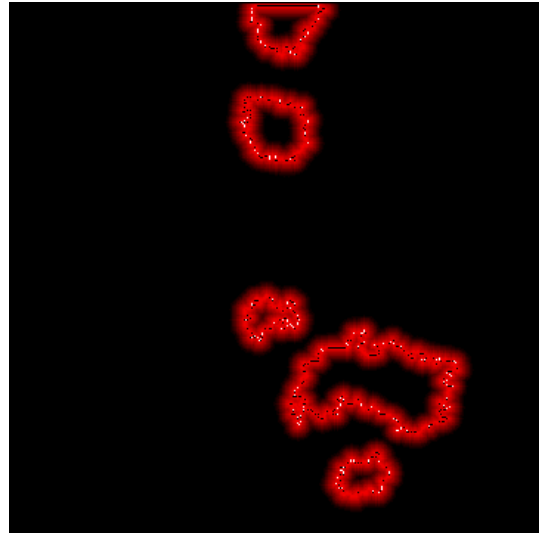


Figure 6. EDM of the outline

2) EDM (Euclidean Distance Mapping) Method

This method is an improvement over the dilation technique in that it avoids some effects of different neighbor rules and the directional bias inherent in the dilation technique. The algorithm sums the Euclidean Distance Maps (EDM) of the object and its inverse. For increasing grayscale levels, the resulting thresholded areas are then plotted on a log-log scale. The fractal dimension is calculated from the slope m using the formula $D = 2 - m$. The EDM of the inverse of the object describes the inside distance to the outline. So the local brightness of the sum image is actually defined by the distance to the outline from both outside and inside. Fig. 6 is the displayed EDM of Fig 4(d) in BIP.

3) *Box-counting method*

This algorithm counts the number of boxes of an increasing size that are needed to cover the one-pixel object boundary. The box size and the number of boxes necessary to cover the boundary are then plotted on a log-log plot and the fractal dimension is determined from the slope m as $D = -m$. Fig. 7 shows the boxes covering the boundary.

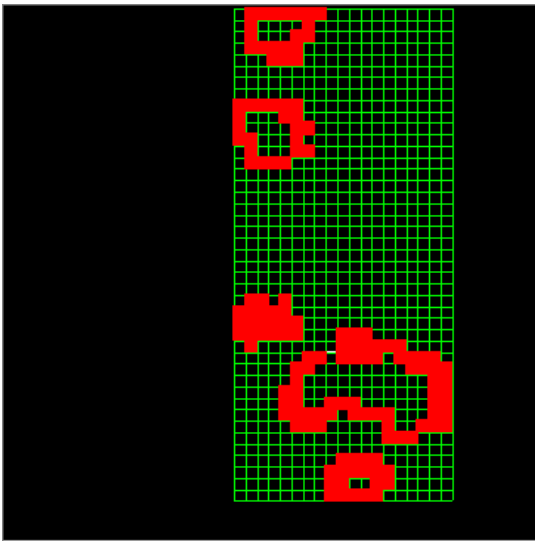


Figure 7. Box-counting method

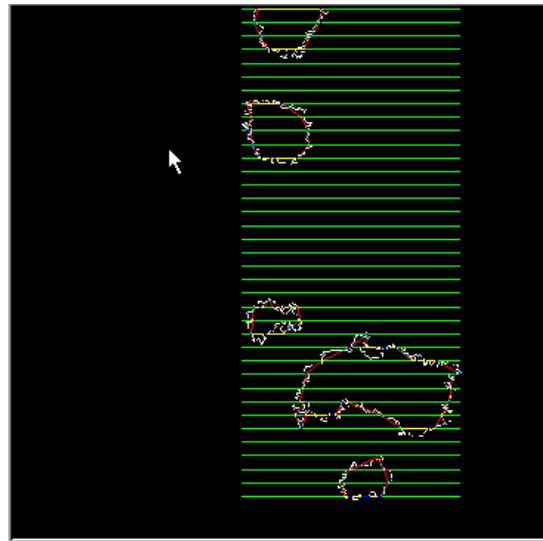


Figure 8. Parallel-lines method

4) *Fast Method*

This involves obtaining the coordinates of centers of all the pixels on the particular boundary in a consecutive list and calculating the straight-line distance between pairs of consecutive points with increasing number of interval points. The fractal dimension is calculated from the slope m of the log of interval length versus the log of the perimeter of the resulting polygon. $D = 1 - m$.

5) *Fast (Hybrid) Method*

This method obtains the coordinates of all the points on the particle boundary in a consecutive list and calculates the straight-line distance between pairs of

coordinates with increasing interval distance. The difference from the previous one is to increase interval distance instead of the number of interval points. Both this method and the Fast method seem very close to the original idea of the Richardson's plot.

6) Parallel-Lines Method

The method uses parallel lines drawn horizontally over the image at increasing lengths of separation. The intersections of these lines and the outlines of objects are used to draw polygons. The log of the separation length between parallel lines and the log of the polygon perimeter are plotted to determine the objects boundary fractal as $D = 1 - m$, where m is the slope. Fig. 8 is the display in BIP using the Parallel-lines Method.

7) Cumulative-Intersection Method

This method and the next two are different from the previous ones. We will refer to them as the area-based methods. The subject of the algorithms is the area instead of the boundary of the shapes. In the case of biofilm images, we still consider the fractal dimension of the outline of the bacteria colonies. So the object mentioned in the three area-based algorithms is the one-pixel outline obtained from the original image.

The Cumulative-Intersection method involves placing a circle centered on the centroid of the object with a radius of r . The number of object pixels (border pixel for the biofilm image) is then counted. The radius is increased and the process repeated until r is equal to the radius of gyration (calculated using area as physical mass). The regression routine uses a value of 1 instead of zero when no pixels intersect the circle

to avoid errors with the natural log function, since this is very likely the case when a small radius is used on a near circular border. The fractal dimension is calculated from the slope m of the log-log plot of r versus the resulting number of cumulative intersecting pixels, using the formula $D = m$. Fig. 9 is the screenshot from BIP using this method.

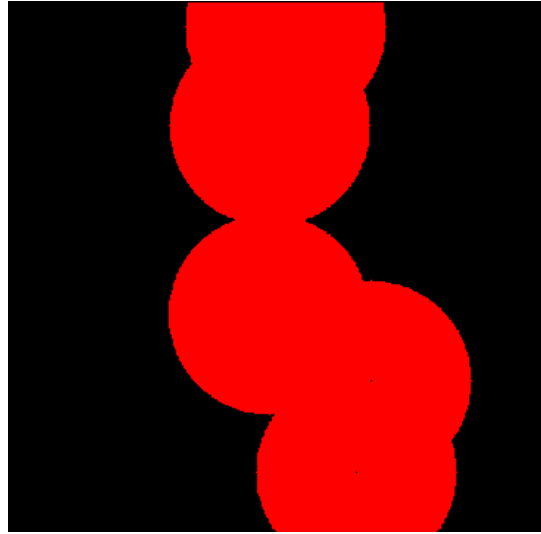


Figure 9. Cumulative-Intersection method

8) *Mass Radius (Long) Method*

This method places circles of radius r centered on all points within the object or the object's radius of gyration. The average of the object areas within each of these circles is denoted by $M(r)$. The value of r is increased and the process repeated. The slope of the log-log plot of $M(r)$ versus r is the mass fractal dimension. Fig. 10 is the display of this method in progress.

9) *Mass Radius (Short) Method*

This method places circles of radius r centered on a user-defined number of random points within the object or the object's radius of gyration. It differs from the previous one only in that not all the points are considered to improve the performance. The slope m of the log-log plot of the average object areas within each of these circles versus radius r is the fractal dimension. Fig. 11 is the display in BIP with the default parameter. The similarity with the Fig. 10 suggests that the results from the two methods be close to each other.

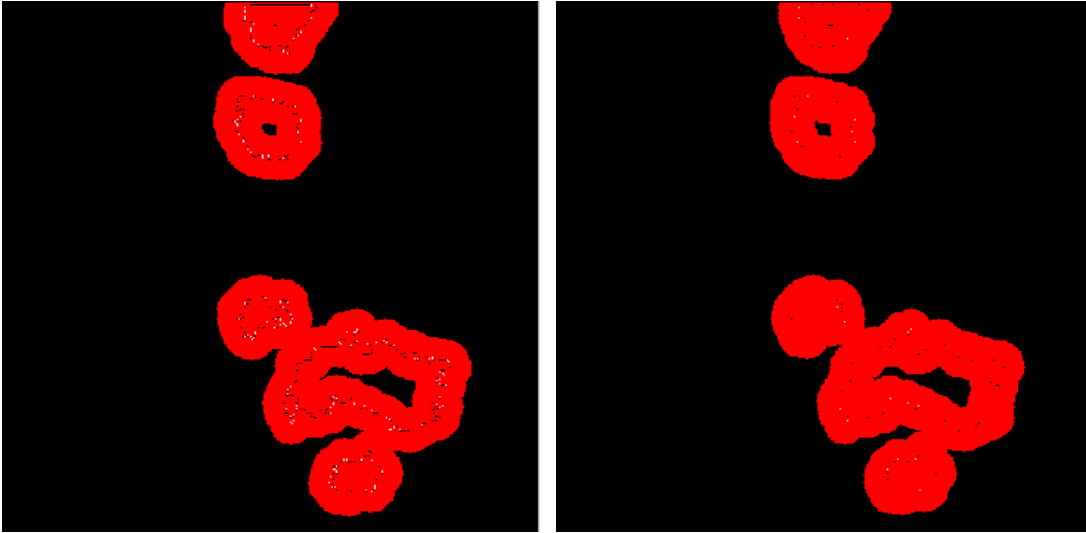


Figure 10. Mass Radius (long) method **Figure 11. Mass Radius (short) method**

10) Corner (Count) Method

A corner is defined as the first point that is farther than a given height in the perpendicular direction from the segment connecting a reference point and a point following in the consecutive point list on the outline. The corner is used as the reference point to find the next corner. If the slope of the log-log plot of the number of corners versus the corner height is m , the fractal dimension is calculated as $D = -m$.

11) Corner (Perimeter) Method

Using the above definition of corner, the perimeter can be calculated as the sum of the distance between consecutive corner points. Plot the log-log plot of this perimeter versus corner height. The fractal dimension is obtained from the slope m using the formula $D = 1 - m$.

Besides the algorithms described above, BIP has additional features to improve the flexibility and efficiency of determining fractal dimension from pixel-based images. Included among them are:

1) Parameter setting: While default parameters are provided, user can set many control parameters, like the threshold used to binarize, the size of small object or small hole to be removed. Corresponding parameters for a specific algorithm show up when that algorithm is chosen.

2) Project mode: The user can choose to process individual image with manually set parameter or to batch process a large quantity of images automatically, which is important when literally tens of thousands of images need to be analyzed.

3) Display of plot: The log-log plot is displayed with the resulting fractal dimension. The deviation from linear relation or linear relation over certain scale range can be seen visually.

BIP accepts TIFF or BMP as input image file format. As mentioned earlier, the computation process of different algorithms can be shown with animation.

3. Generating of Images with Known Fractal Dimension

3.1 Purpose of generating standard images

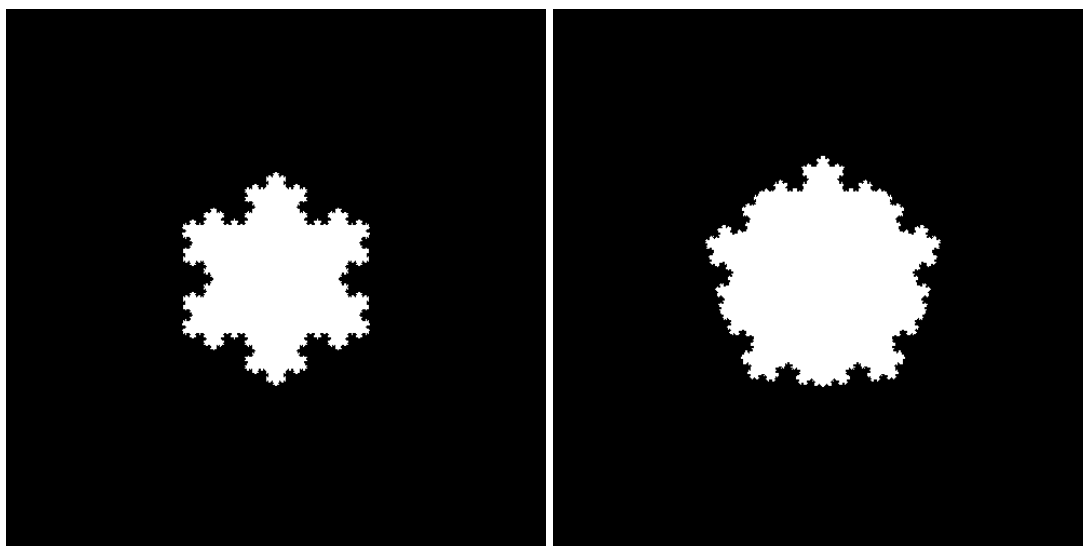
With BIP or with other tools, it is possible to calculate fractal dimensions from pixel-based images and use the result to establish relation between the calculated fractal dimension and other variables in question, as bacteria strains, time, or nutrient concentration in the case of biofilms. But whether the algorithm or the implementation captures the properties of fractals or not, and whether the original images are fractals, are questions that remain unanswered.

One way to test any existing or new methods to compute FD is to use pixel-based images generated using fixed and simple recursive rules and with known fractal dimensions. For this purpose, a piece of software is developed to generate images of shapes with required FD. It is named KochGen because the shapes are based on the classic fractal Koch curve. It is designed to be able to generate images with enough variety so that properties other than the fractal dimension will not affect the testing result. In spite of the limitations of image resolution, the shape should have known fractal dimension over any scale range. In the other words, they should be fractal in the strict sense of the term. It is also desirable that the generated image be, to some extent, similar to the real image analyzed.

3.2 Fractal generator KochGen

The shapes in the images are generated by KochGen with recursive relation as described in Section 1.2 and Fig. 2. In order to achieve diversity, it provides several

options to choose. To generate closed fractal boundary, regular polygons are used as the starting shape, which we will set as the initial polygon. The number of sides in the initial polygon can be set by the user. Theoretically any number above 3 can be used. Since the length of a side become very small if the number is too large, we used a triangle, a square, a pentagon and a hexagon. The current version allows number of sides from 3 to 10. Figs. 12(a) and (b) are generated from an initial triangle and an initial pentagon, respectively, both with the fractal dimension 1.26186.



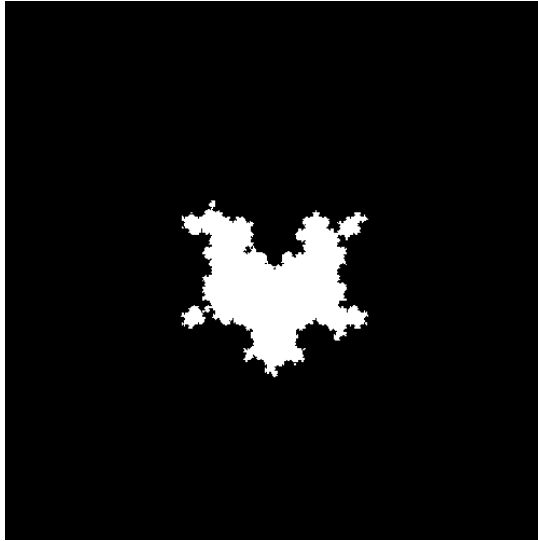
(a) Initial number of sides = 3

(b) Initial number of sides = 5

Figure 12. Snowflake, FD=1.26186

Another important choice is between the snowflake shape shown and the so-called random Koch curve. The former is generated by always making the replacement with the tent-like angle be outside of the polygon. The latter gives the angle 50% probability to be on the other side the original segment (Peitgen et al. 1992). Fig. 13 is a corresponding random Koch curve of Figure 12(a), which is a general snowflake, both starting with an initial triangle and having same FD.

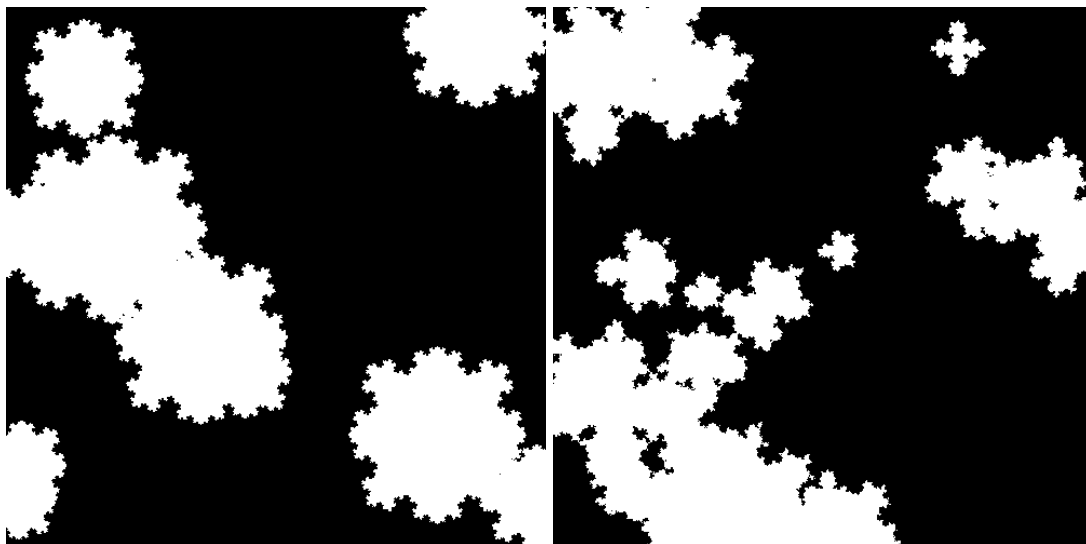
KochGen also supports generating a group of fractals in the image instead of a single shape. The individual object in the group has a random size and is placed at



**Figure 13. Random curve, $FD=1.26186$,
with initial number of sides = 3**

random positions in the image. In the image with a default size of 500×500 pixels, the size of an individual object (defined as the radius of the circumscribed circle of the initial polygon) is between 0 and 100. A size of 100 was used when a single object was generated. The number of objects within the image is randomly chosen between 0 and 20. Both choices are

selected with uniform probability. In the current version, the initial polygons in the group have the same number of edges. Fig. 14(a) shows an image of fractal group (snowflakes based on initial square with fractal dimension 1.26186). Fig. 14(b) shows a group of random curves (same initial polygon and fractal dimension as Fig. 14(a)). It looks rather like a binarized biofilm image, as shown in Fig. 4(a).

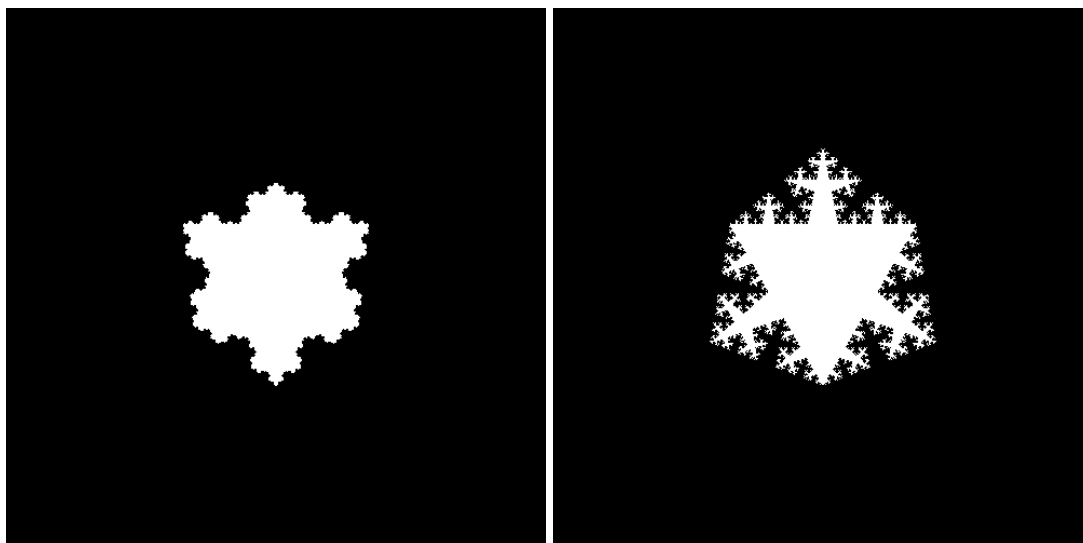


(a) Snowflake

(b) Random curve

Figure 14. Group of shapes, $FD=1.26186$, initial number of sides = 4

The most important parameter, the fractal dimension, is set through the scaling factor σ , which is defined as the ratio of the new segment to the original one. In the standard example shown in Figure 2(b), σ is $1/3$. The number N is always 4 here. As discussed in Section 1.2, the formula for FD is $D = \log N / \log(1/\sigma) = \log 4 / \log(1/\sigma)$. Figs. 15(a) and (b) are two images that are different from Fig. 12(a) only in the value of the fractal dimension.



(a) FD=1.17327

(b) FD=1.59803,

Figure 15 Snowflake with initial number of sides = 3

KochGen generates quadratic Koch islands too. The recursive relation replaces a segment with the seven smaller segments perpendicular to the neighboring segments, as shown in Fig. 16 (Peitgen et al. 1992).

Fig. 17 shows a single quadratic Koch island. In KochGen, it is generalized to starting with various initial polygons other than a square, and random replacement direction. It can also be used to generate a group of islands, but the fractal dimension cannot be changed. In the current version, since $N = 8$, $\sigma = 0.25$, the fractal dimension

is always $D = \log N / \log(1/\sigma) = \log 8 / \log 4 = 1.5$. The scaling factor setting in the user interface is ignored.

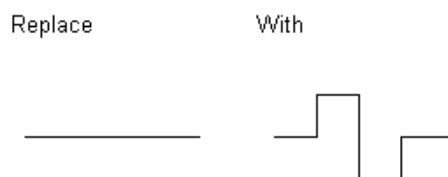


Figure 16. Recursive relation for quadratic Koch island

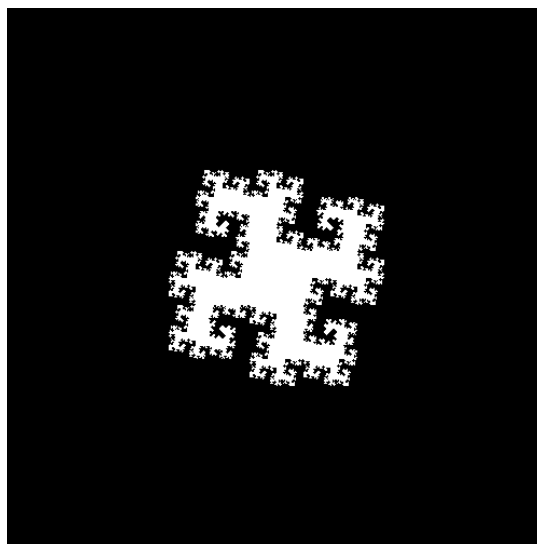


Figure 17. Quadratic Koch island, FD=1.5

Using combination of these variables, this software can make images with more diverse appearance.

3.3 Discussion on Implementation

Output format of the image files can be TIFF or BMP. The program is developed in Java. Java Advanced Image (JAI) package is used, and thus is very convenient to extend to most widely used image file formats.

Some implementation details need to be discussed more. The boundary of each object is implemented with a PathIterator in Java 2D, which is a consecutive list of points. When the distance between two consecutive points is smaller than one pixel in the image, the recursion need not continue or else unexpected results could occur at

that point. In KochGen recursion is terminated if the segment is shorter than the size of pixel.

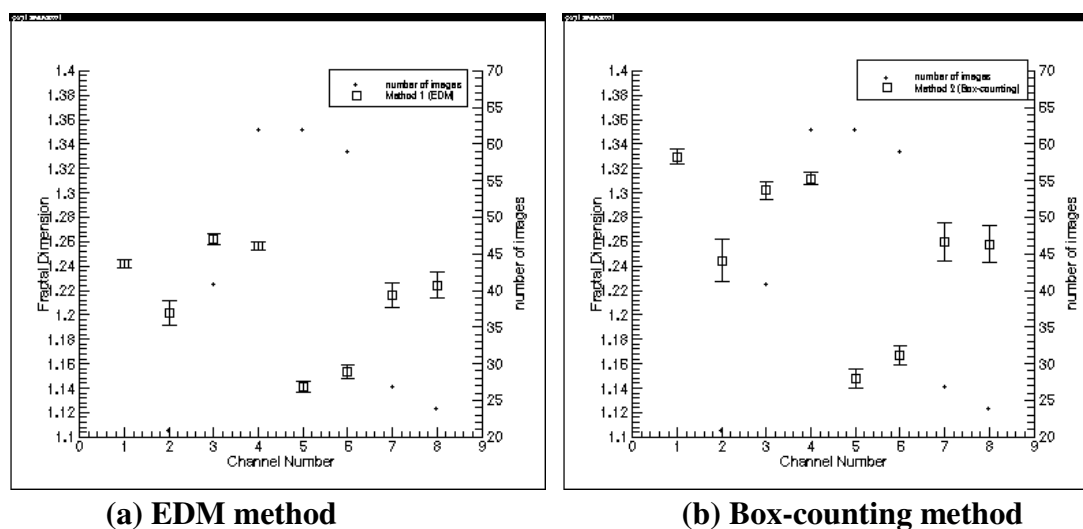
Theoretically the fractal dimension of a curve could be any real number between 1 and 2. The corresponding scaling factor σ is respectively 0.25 and 0.5 for the recursive relation of the tent-like angle. In the actual implementation, when σ is over 0.4 the generating process becomes very slow. Depending on the initial number of sides or the size, the scaling factor of $\sigma = 0.37$ works well for most cases.

KochGen supports choice of background color and object color. It can demonstrate the recursion process step by step for a standard Koch snowflake. For convenience, batch generation is also implemented in order to make a set of images to test software packages like BIP. When using batch generation, the initial edge number and scaling factor in the user interface are ignored. All the images for initial edge number 3, 4, 5, 6 are generated. Each group includes images of five different scaling factors 0.25, 0.3, 1/3, 0.35, 0.37, except in the case of quadratic Koch island, which has only one possible fractal dimension. Since a random process is involved, 10 images are generated for each setting of the parameters.

4. Computational Experiments

4.1 Experiments on biofilm images

Computational experiments were carried out on about 11,000 biofilm images of *P. aeruginosa*. These pictures were taken at five time points and included four different strains. Figs. 18(a) and (b) are the results obtained by two of the eleven algorithms on 311 images, which were chosen as having sufficiently large microbial colonies in them from about 600 images. Not every algorithm can reveal the obvious difference shown in these results.



(a) EDM method

(b) Box-counting method

Figure 18. Results for different strains of bacteria

4.2 Experiments on generated images

Using the KochGen, a set of standard images with known fractal dimension were generated. Totally 730 images are summarized as following.

Single snowflake: 21 images (6 scaling factor 0.25, 0.3, 1/3, 0.35, 0.37, 0.42 based on triangle; 5 scaling factor 0.25, 0.3, 1/3, 0.35, 0.37 on square, pentagon, hexagon respectively.)

Single random curve: 200 images (10 random images for each of the 5 scaling factor 0.25, 0.3, 1/3, 0.35, 0.37 on four different initial polygons.)

Group of snowflakes: 200 images (same parameters as above.)

Group of random curves: 200 images (same parameters as above.)

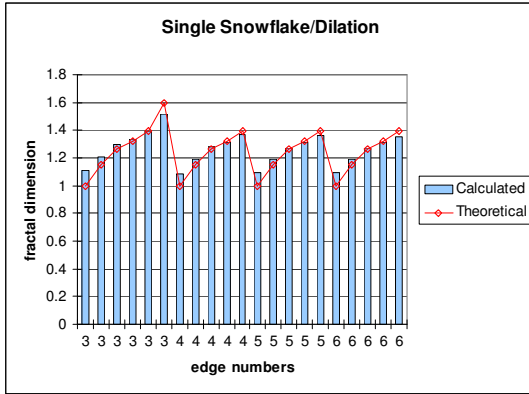
Quadratic Koch island: 124 images (4 single snowflake for different initial polygons; 100 random images for each of the three categories single random curve/group of snowflakes/group of random curves on four different initial polygons.)

The fractal dimensions with the corresponding scaling factor is listed the following table.

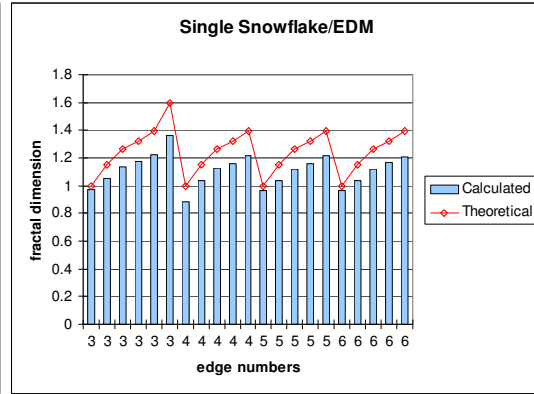
Scaling factor σ	0.25	0.3	1/3	0.35	0.37	0.42	0.5
Fractal Dimension D	1	1.1514	1.2618	1.3205	1.3943	1.59803	2

All the eleven algorithms in BIP are tested with these images. We will not differentiate between the algorithm and its implementation in BIP when discussing the results.

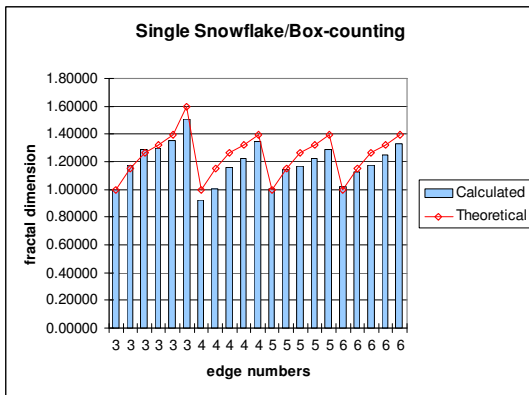
The results shown in Figs. 19(a)~(k) are from the 21 single snowflakes. Each figure is for one algorithm. The red curve in the graphs is the theoretical value of the fractal dimension. The blue bars show the calculated valued from BIP.



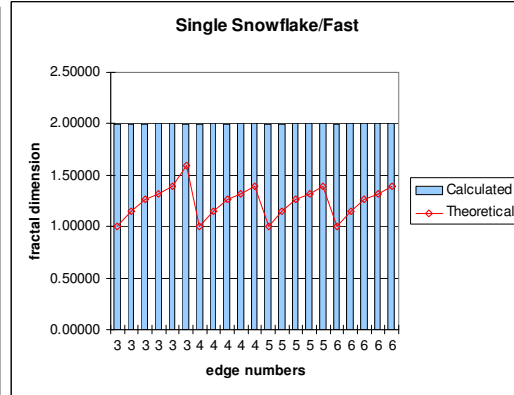
(a) Dilation method



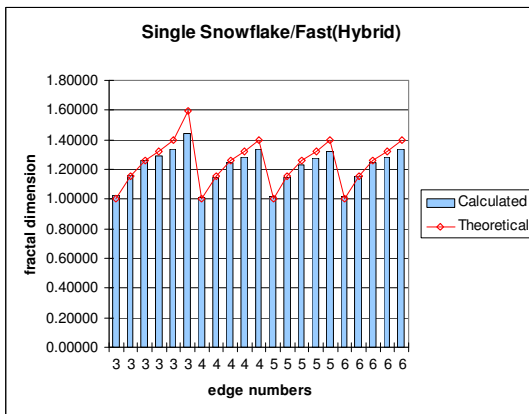
(b) EDM method



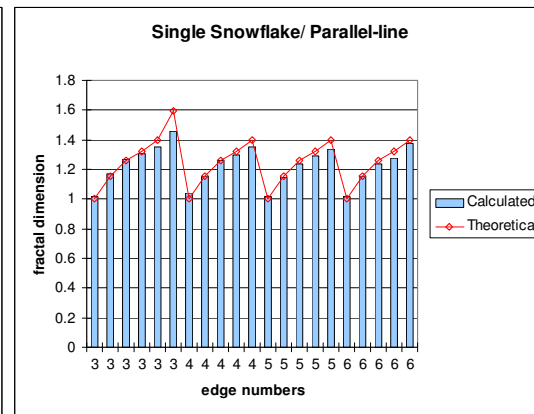
(c) Box-counting method



(d) Fast method



(e) Fast (hybrid) method



(f) Parallel-lines method

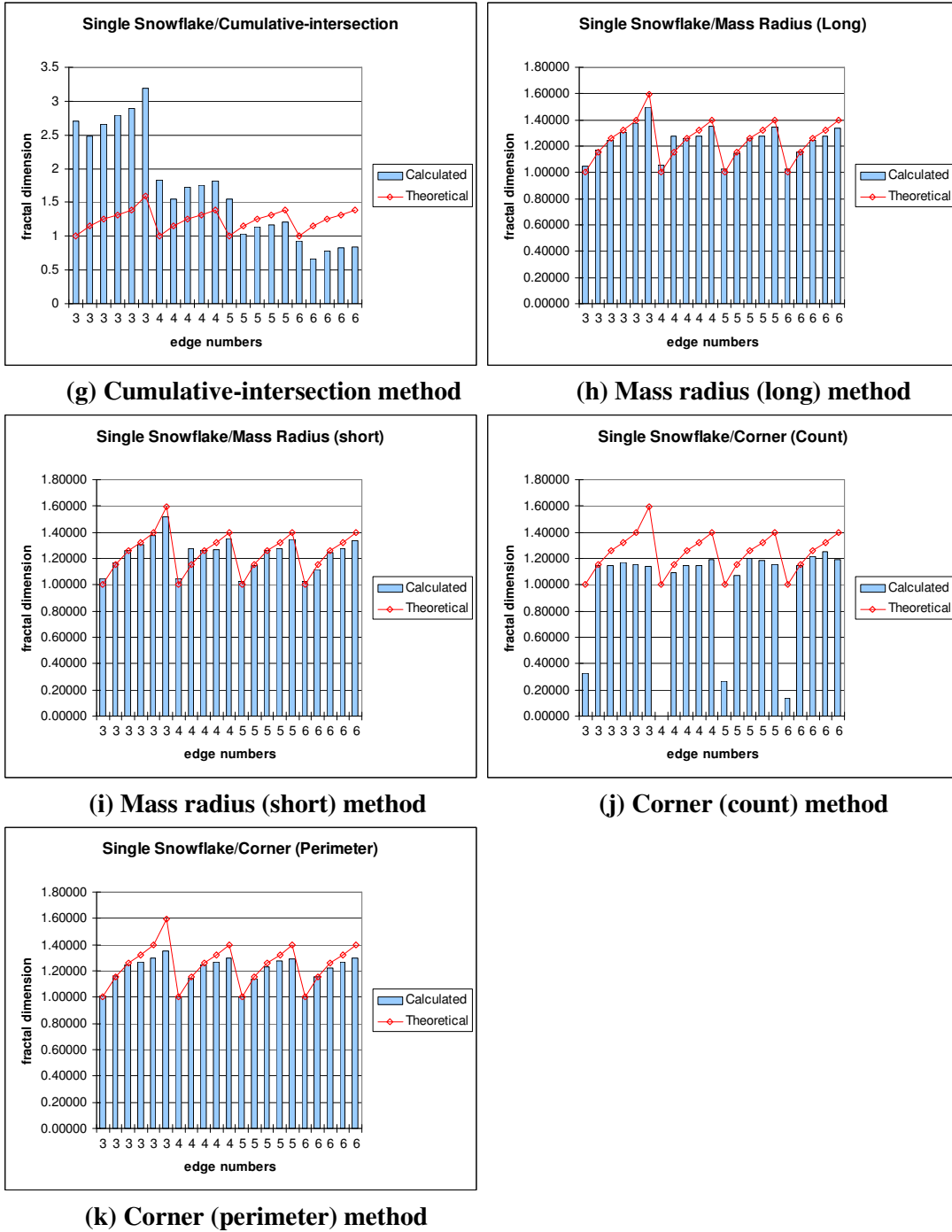
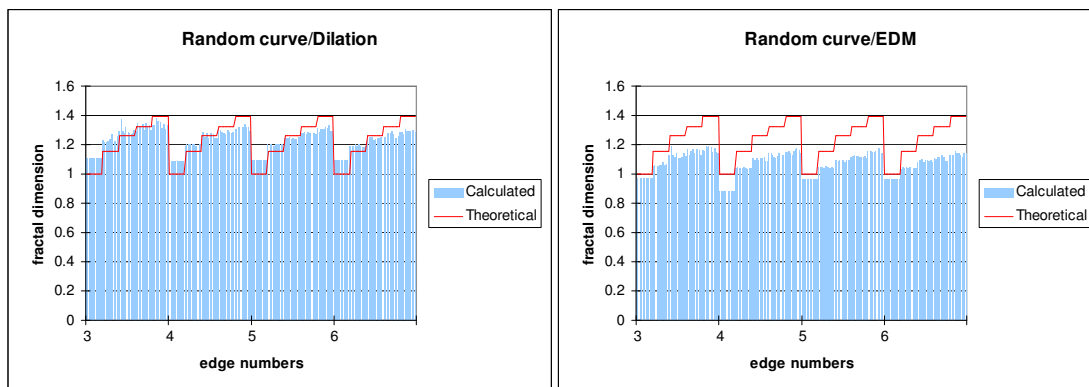


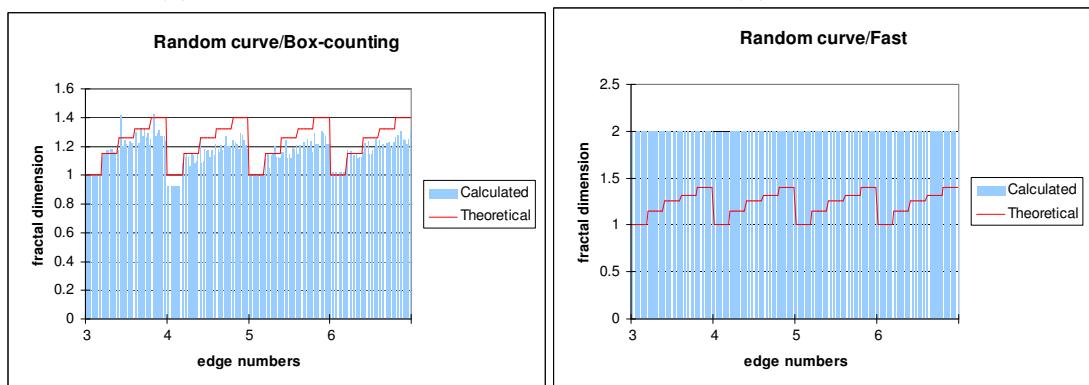
Figure 19. Results on single snowflakes

The results shown in Figs. 20(a)~(k) are from the 200 random curves. Each figure is using a single algorithm. The curve in the graphs is the theoretical value of the fractal dimension. Since there are too many values, they are indicated as a shadowed area.



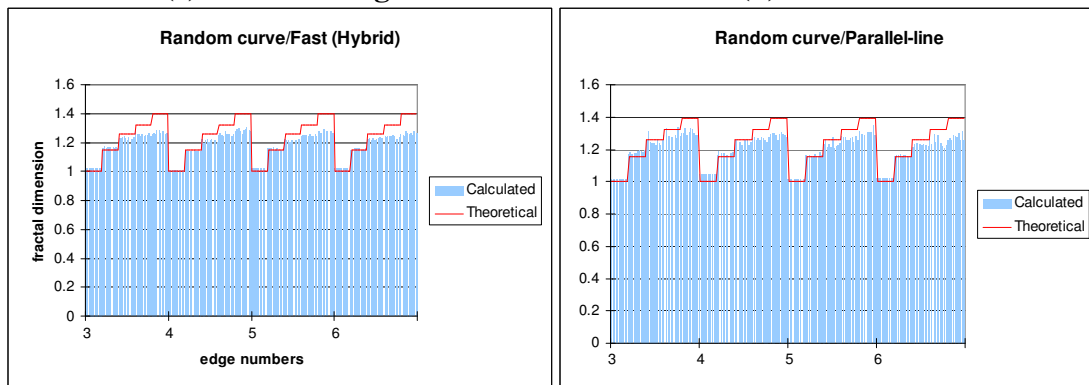
(a) Dilation method

(b) EDM method



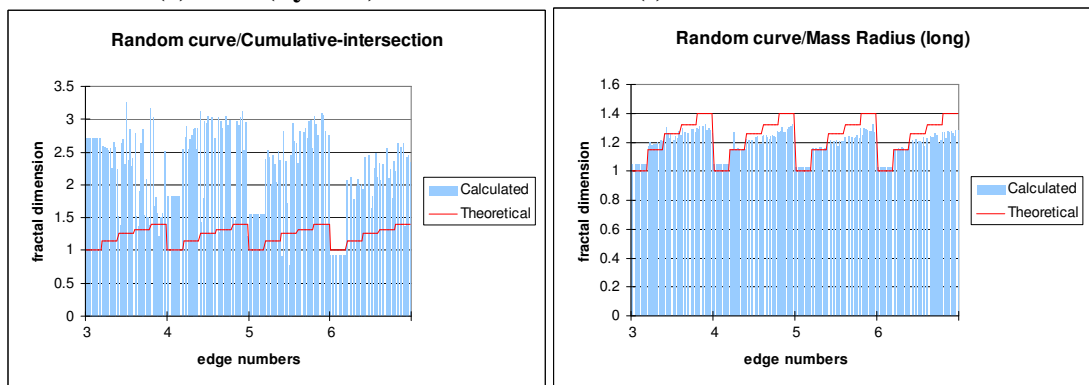
(c) Box-counting method

(d) Fast method



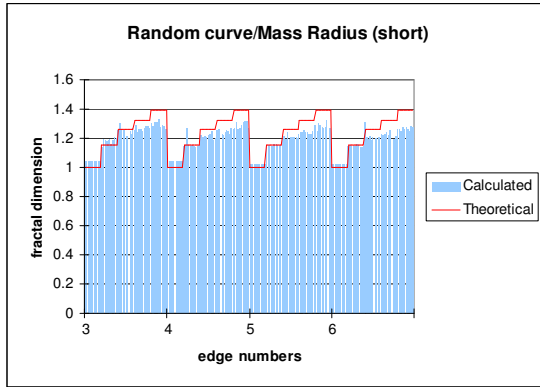
(e) Fast (hybrid) method

(f) Parallel-lines method

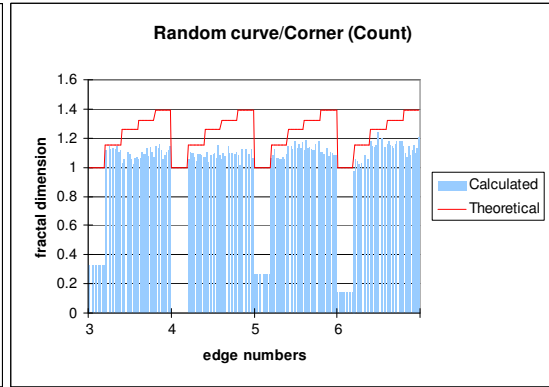


(g) Cumulative-intersection method

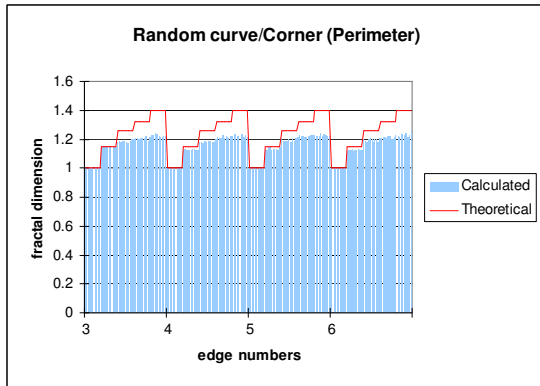
(h) Mass radius (long) method



(i) Mass radius (short) method



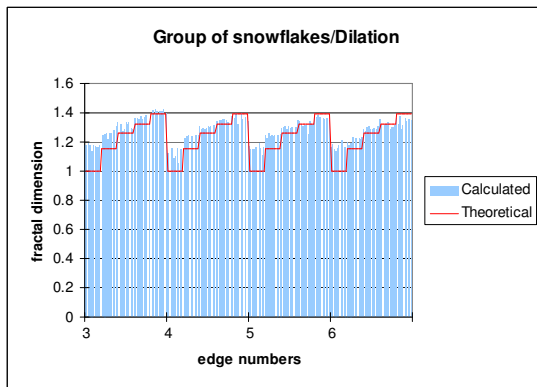
(j) Corner (count) method



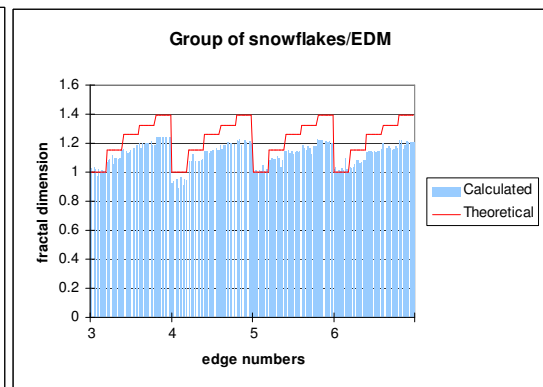
(k) Corner (perimeter) method

Figure 20. Results on random curves

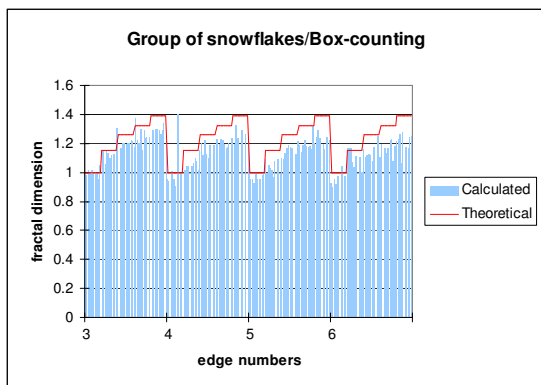
The results shown in Figs. 21(a)~(k) are from the analysis of 200 groups of snowflakes. Same convention is used as for random curves.



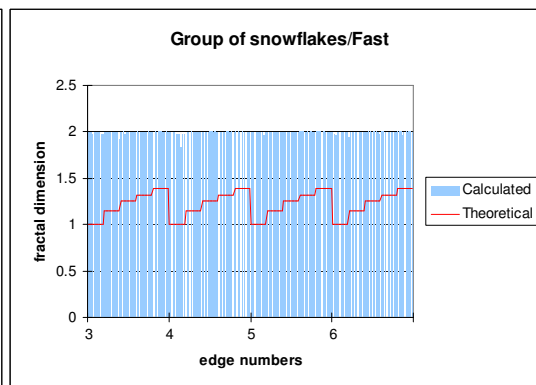
(a) Dilation method



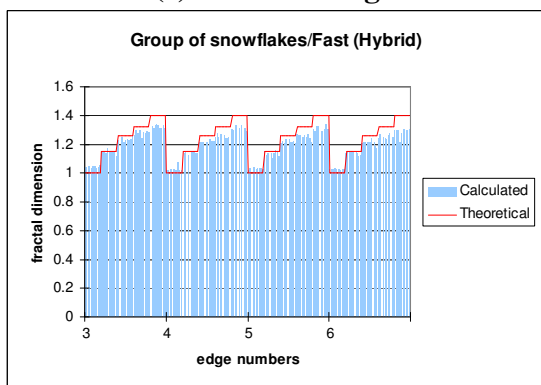
(b) EDM method



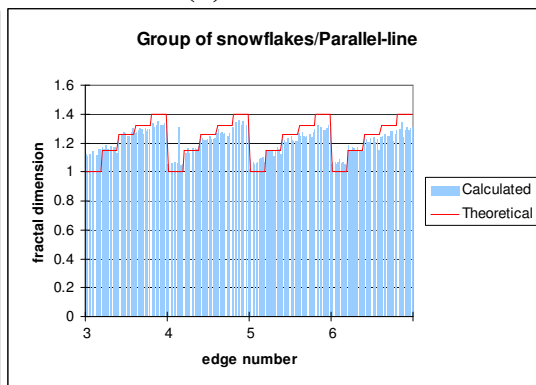
(c) Box-counting method



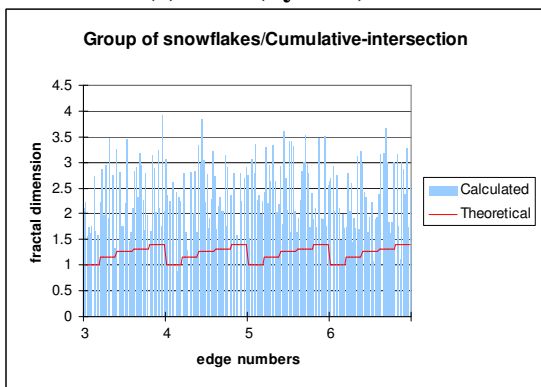
(d) Fast method



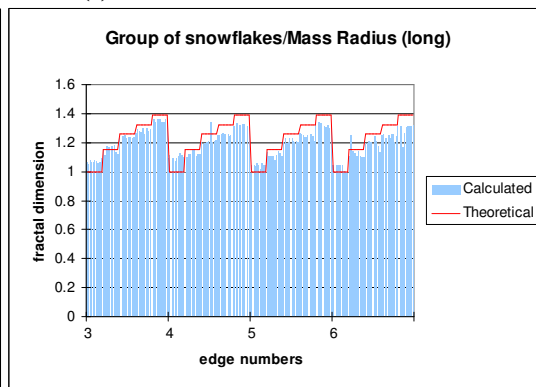
(e) Fast (hybrid) method



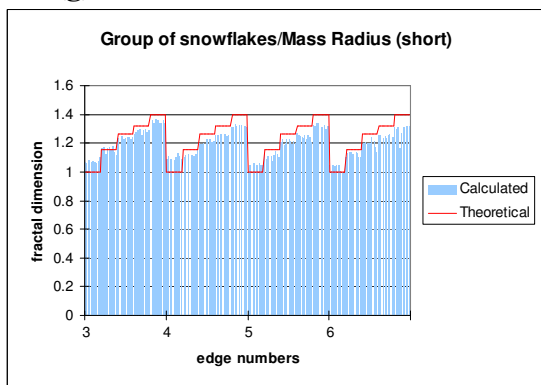
(f) Parallel-lines method



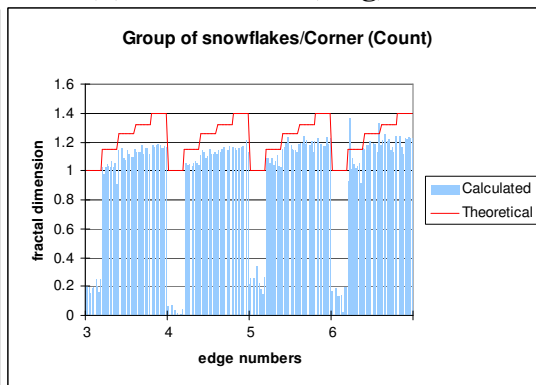
(g) Cumulative-intersection method



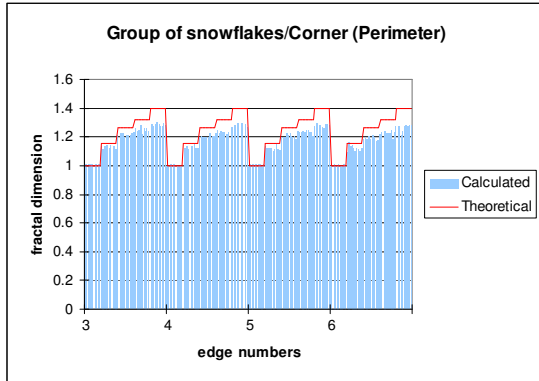
(h) Mass radius (long) method



(i) Mass radius (short) method



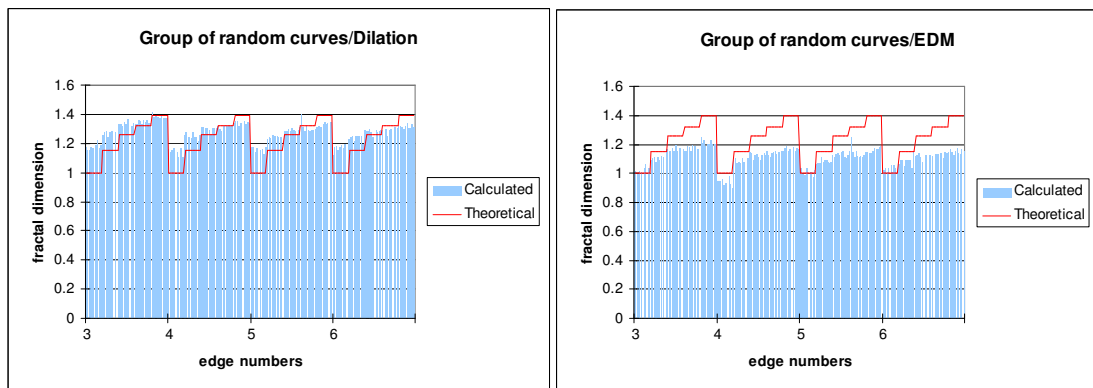
(j) Corner (count) method



(k) Corner (perimeter) method

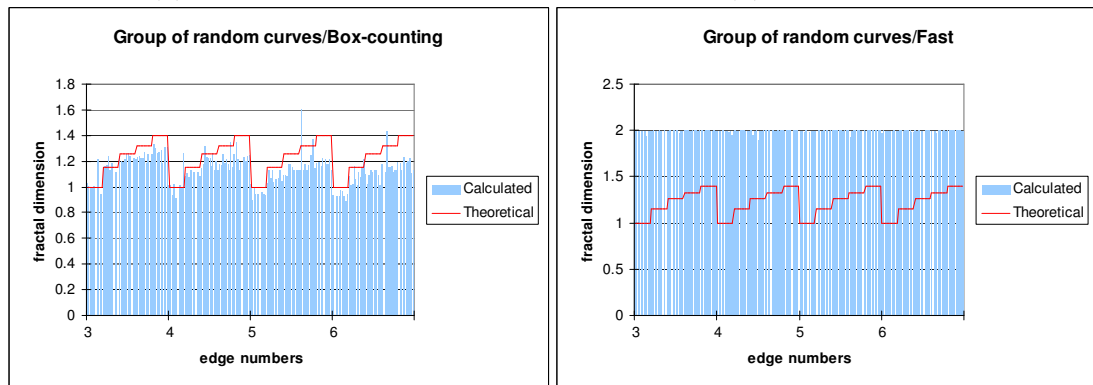
Figure 21. Results on groups of snowflakes

The results shown in Figs. 22(a)~(k) are from the 200 images containing groups of random curves.



(a) Dilation method

(b) EDM method



(c) Box-counting method

(d) Fast method

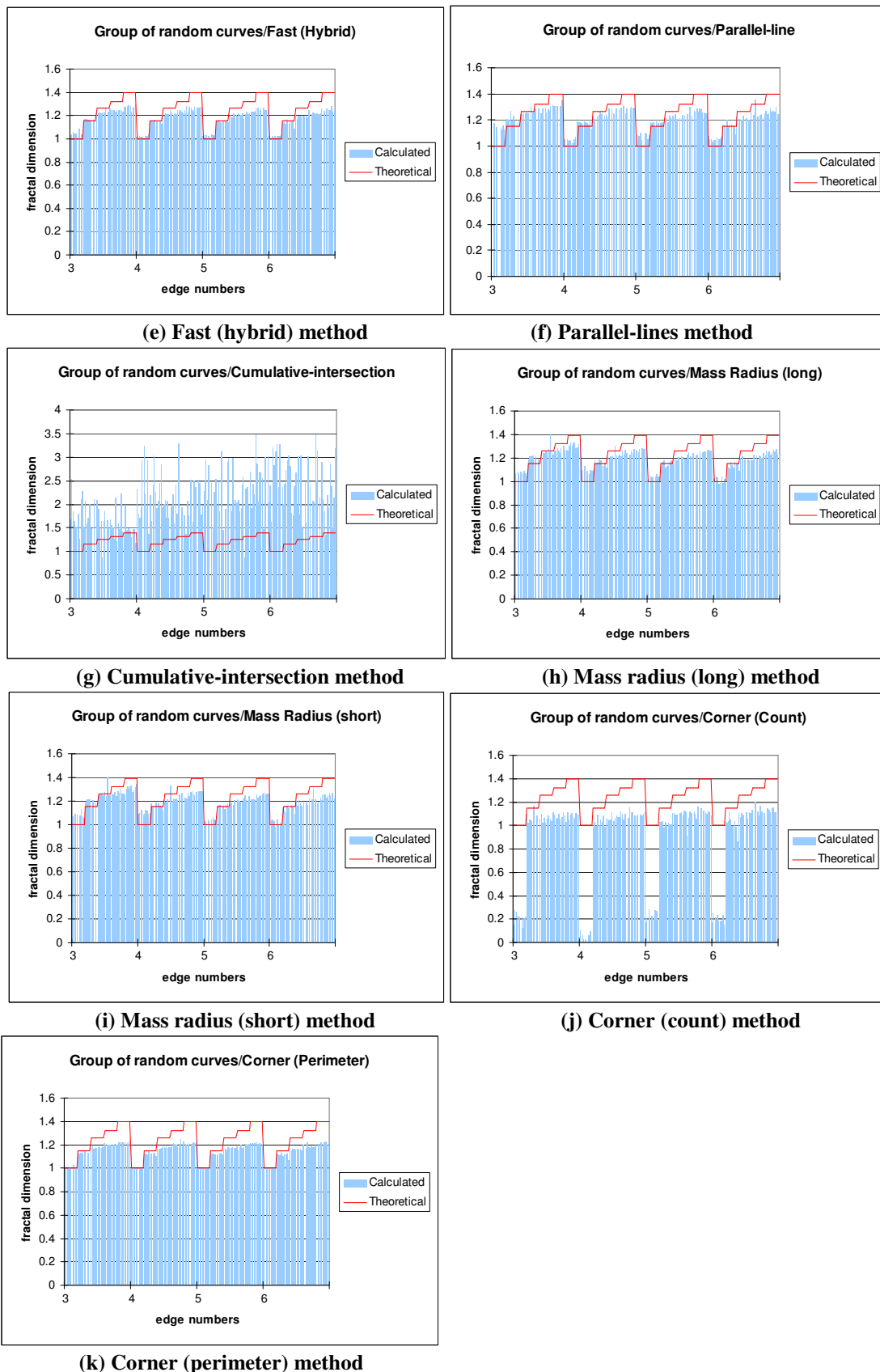
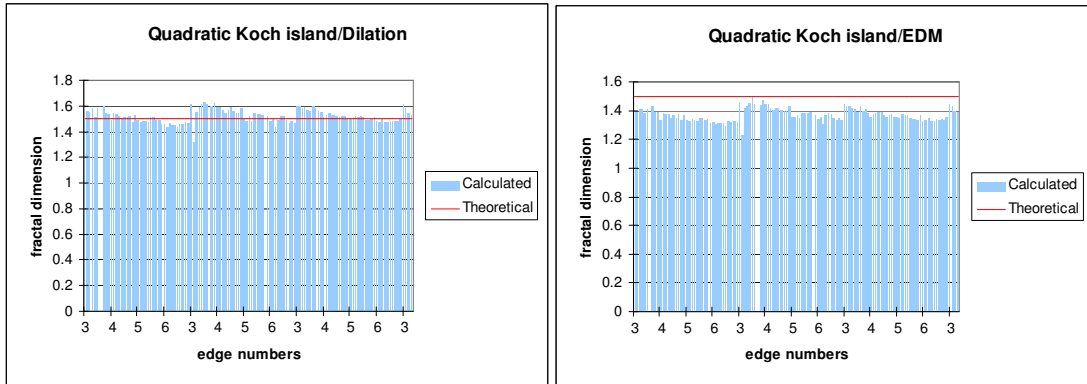


Figure 22. Results on groups of random curves

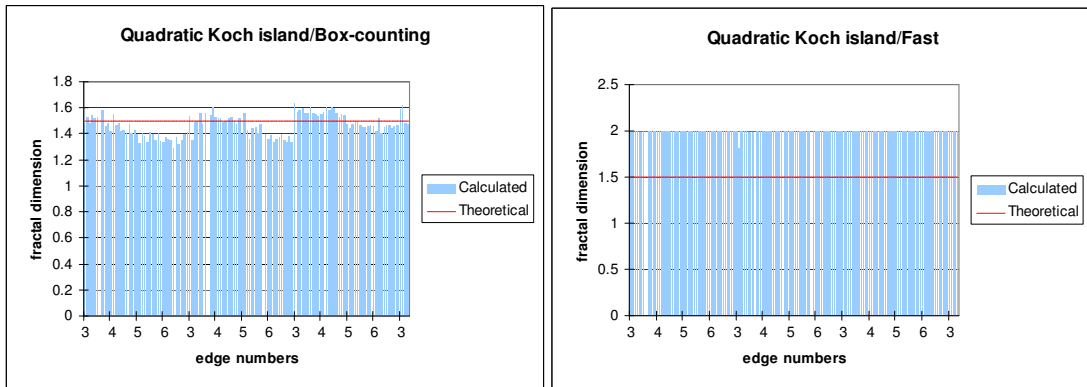
The results shown in Figs. 23(a)~(k) are from the 124 quadratic Koch island.

The theoretical fractal dimensions for all of them are 1.5.



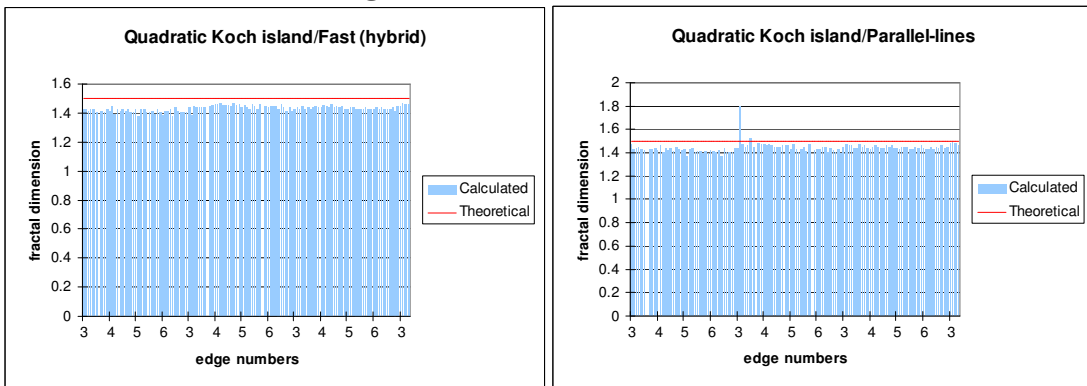
(a) Dilation method

(b) EDM method



(c) Box-counting method

(d) Fast method



(e) Fast (hybrid) method

(f) Parallel-lines method

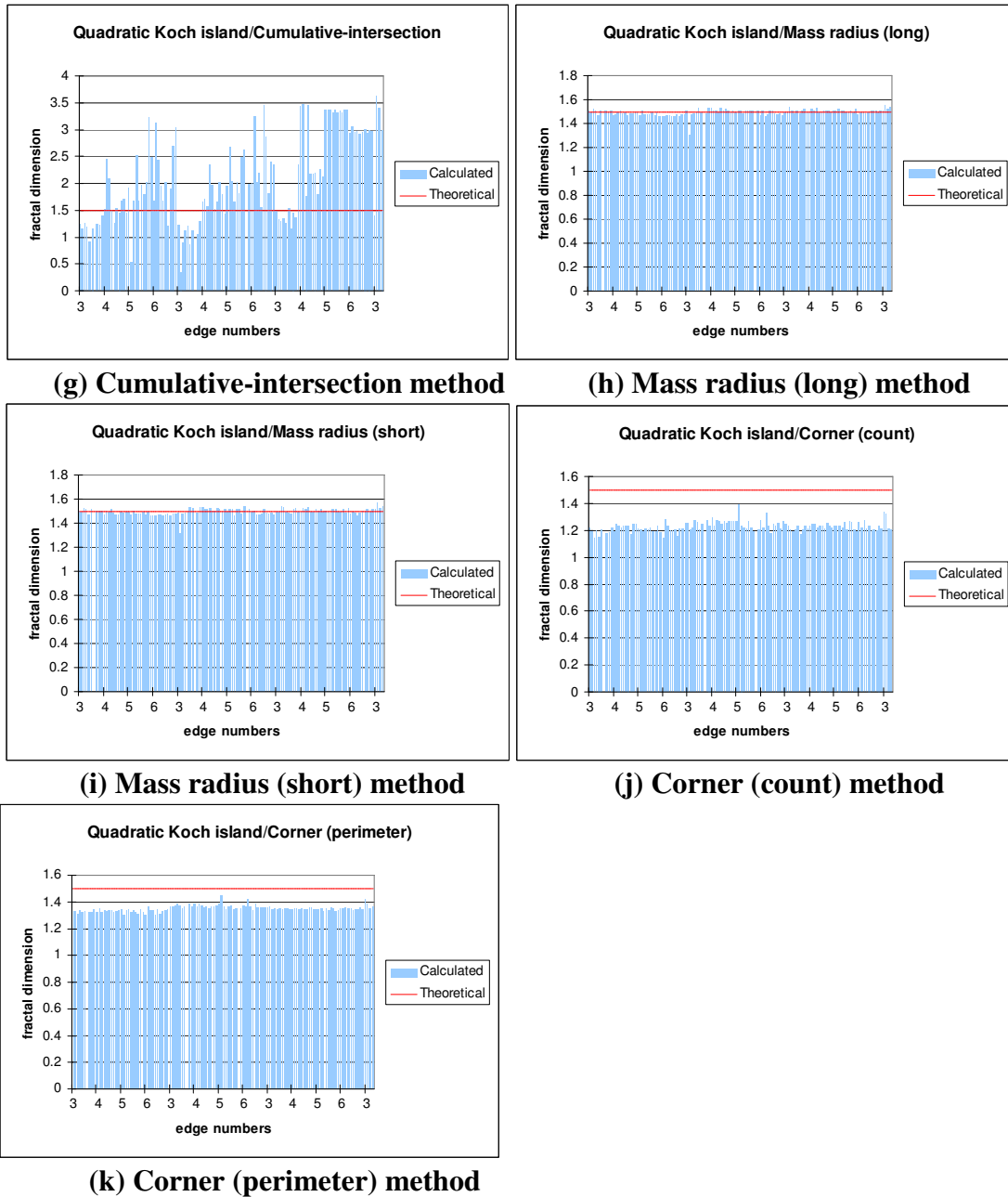


Figure 23. Results on quadratic Koch island

The Dilation method only predicts the relative magnitude of the fractal dimension correctly. It overestimates for low fractal dimension images, especially for those that are supposed to be one. For higher fractal dimension images, it underestimates. In cases where the initial polygons have more sides, the

underestimations are a little more obvious; this is not unexpected because these can be thought of as being less “rough”.

The EDM method catches the relative magnitude too, but it underestimate more for images with high FD, especially for the groups of random curves.

The same is true of the Box-counting method. However, the error looks smaller than the EDM method, but is not as consistent for several images with the same parameters.

The Fast (hybrid) method is good except that it does not evaluate higher fractal dimension high enough. Parallel-lines method is similar at the high fractal dimension end, but overestimates 1-dimensional images for the group cases.

The two mass radius methods are generally good, but they also have the problem of underestimation with high fractal dimension cases. They work very well for quadratic Koch island for which Dilation, EDM, and Box-counting show variation depending on the number of sides of initial polygons, and Fast (hybrid) and Parallel-lines underestimate. Another problem is that the Mass radius (long) is slower than other methods, so the similar results from Mass radius (short) suggests that it is a good alternative of the other method.

The Corner (perimeter) method does not work very well at the high fractal dimension either, but is close to the Fast (hybrid) and Parallel-lines methods. Its outstanding merit is that it is consistently accurate for 1-dimensional images.

The Fast, Cumulative-intersection and Corner (count) methods basically fail. Most results by Fast method are very close to 2, almost constant. The Cumulative-intersection method obtained result not obviously correlated with fractal or other

parameters and is out of any reasonable range. The Corner (count) method obtained values much smaller than 1 for those with theoretical 1-dimensional shape, and results for all other images have no obvious difference from each other.

The common problem in these 11 algorithms is that they cannot predict the high fractal dimensions well enough. But this could be due to the low resolution compared to the sizes of the shapes in the image – smaller structure is not represented enough. Whether it is true or not can be revealed from further study on images with more pixels.

The Mass radius methods are the best although they do not overcome the common problem either. Except for the three bad methods (it is possible that the problem comes from implementation), all the other six methods have similar general performance.

5. Simulation of Biofilm Growth

Another important direction of biofilm research is to model the growth of biofilm. The ultimate goal is also to analyze what and how the various factors affect the structure of biofilms. This thesis has taken a step in this direction as well.

A piece of software based on the work of Meyer (1999) was developed with Visual C++ to simulate the growth of biofilms. A 2-dimensional cellular automata model was used.

It can demonstrate the growth process starting with inoculation of bacteria. There is one or zero cell (bacterium) in each automata cell array, which contains nutrient state of the bacteria if a bacterium is present, and available nutrient units. In each step of iteration, the cell array is updated in three steps:

- 1) nutrient consumption;
- 2) cell (bacteria) division;
- 3) nutrient diffusion.

Corresponding rules for these steps constitute the transition rule of the cellular automata.

Various control parameters, including the initial nutrient concentration and the initial number of bacteria can be set up interactively. The display methods include growing the biofilm in time steps, starting and stopping growing on a mouse click, or growing until a given time step. The number of bacteria and the current time step are displayed on the status bar. The local nutrient concentration is shown with different brightness.

If smaller cells are used in the simulation, the display could be fed to BIP as images to analyze their fractal dimension. From the observation, there is an obvious difference of boundary roughness for different nutrient concentrations.

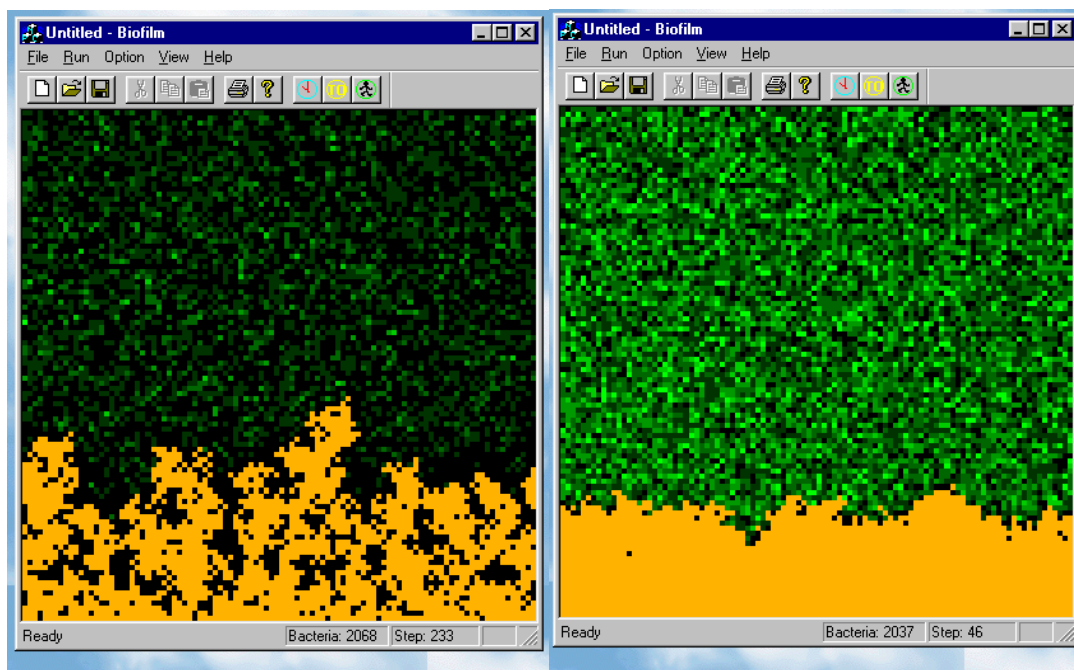
Fig. 24(a) shows the display when initial nutrient is 6000 units and initial number of bacteria is 50. Fig. 24(b) is the display when initial nutrient is 20,000 and initial number of bacteria is 25. The numbers of iterations (time steps) are 233 and 46, while the current numbers of bacteria are close to each other, 2068 and 2037, respectively. More nutrient concentration generates smoother structures. Even when smaller initial number of bacteria has opposite influence, the difference is obvious. Meyer's implementation in MATLAB has the same conclusion.

The links between mathematical models of biofilm development and well-controlled, reproducible biofilm experiments still need to be established. Although this model is very simplified from the biological point of view, Meyer (1999) did an informative discussion relating the simulation to the corresponding condition of actual biofilm.

From the computer science point of view, it is also interesting since it is a typical cellular automata - the global behavior is controlled by (or resulted from) some simple local rules. Peitegn et al. (1992) discussed generating fractals with cellular automata, so this software is related to other topics in this thesis from another direction.

Direct improvement of this simulation could be a continuous diffusion model (Picioreanu et al. 1998) or a 3-dimensional model. The 3-dimensional extension has

the problem of how to visualize and how to compare with biofilm experiment, which usually takes 2-dimensional images.



(a) Initial bacteria: 50, nutrient: 6000 (b) Initial bacteria: 25, nutrient: 20000

Figure 24. Simulation of biofilm growth

6. Conclusion

(1) The fractal dimension calculated with BIP is clearly distinguished for different bacterial strains. But no correlation was found between time of growth and the fractal dimension. Quantifying experimental images by programs like BIP or COMSTAT (Heydorn 2000), accompanied by numerical model generated data, is an important way to improve the understanding of microbial biofilm development.

(2) A powerful tool called KochGen was developed to evaluate algorithms calculating fractal dimensions of pixel images and their implementation. Results from the algorithm are supposed to conform to the known fractal dimension of these standard images. The method proved to be effective. It is valuable for current and future study of fractal dimension from pixel-based images.

(3) Eleven algorithms implemented in BIP were tested thoroughly. Mass radius methods obtained the best results. At higher fractal dimension, none of the algorithms are satisfactory.

(4) Many factors influencing the computational result of fractal dimension of pixel-based image are still not clear. For example, fractal dimension that are only valid in a certain scale range may be lost by the low resolution. The inaccuracy of the algorithms in BIP at high fractal dimension may be related to this.

(5) Simulation using a cellular automata model suggests that nutrient concentration and initial density of bacteria influences the structure of resulting biofilms.

(6) There are many interesting topics deserving further exploration in this area. For example, how to analyze the image with more than one strain of bacteria,

quantitatively deciding the deviation of those log-log plots from exact linear relation or finding out the ranges where linear relation holds, obtaining pictures taken at different times but at a fixed spot, extending the analysis to 3-dimensional case with pictures taken at layers distancing with each other at the order of magnitude of a pixel, and studying the influence of image resolution, etc., are all worth more effort.

Works Cited

- [1] Becker, K.-H., Dörfler, M., Dynamical System and Fractals – Computer graphics Experiments in Pascal, Cambridge University Press, 1989.
- [2] Costerton, J. W., Stewart, O. S. & Greenberg, E. P., Bacteria Biofilms: a Common Cause of Persistence Infection, Science 284, 1318-1322, 1999.
- [3] Heydorn, Arne, Alex Toftgaard Nielsen, Morten Hentzer, Claus Sternberg, Michael Givskov, Bjarne Kjær Ersbøll, and Søren Molin, Effects of Species Identity and Nutrient Concentration on Pseudomonas spp. Biofilm Development: Automated Image Analysis for Analysis for Quantification of Biofilm Structure, Biofilm 2000 Conference, Big Sky, Montana, 2000.
- [4] Lewandowski, Z., D. Webb, M. Hamilton, and G. Harkin, Quantifying Biofilm Structure, Wat. Sci. Tech. Vo. 39, No.7, pp. 71-76, 1999.
- [5] Meyer, Carsten, Simulation of Biofilm, Pre-project at Institute of Physics and Institute of Microbiology, Technical University of Denmark, 1999.
- [6] Peitgen, H.-O., H. Jurgens, D. Saupe and E. M. Maletsky, Fractal for the Classroom, Part Two Complex Systems and Mandelbrot Set, Springer-Verlag, 1992.
- [7] Picioreanu, C., M. C. M. van Loosdrecht, and J. J. Heijnen, A New Combined Differential-Discrete Cellular Automaton Approach for Biofilm Modeling: Application for Growth in Gel Beads, Biotechnology and Bioengineering, Vol. 57, No. 6, March 20, 1998.
- [8] Russ, J. C., Computer Assisted Microscopy, Plenum Press, 1992.

- [9] Sheu, John, Mike Turvey and Luke Shulenburger, Fantastic Fractal, <http://library.advanced.org/12740/>, {on-line} June 25, 2000.
- [10] Warfel, Matthew, Fractal Analysis Macro, <http://www.cee.cornell.edu/~mdw/fractech.html>, {online} June 25, 2000.
- [11] Wimpenny, J. W. T., Colasanti, R., A Unifying Hypothesis for the Structure of Microbial Biofilms Based on Cellular Automaton Models, FEMS Microbiology Ecology 22 (1997) 1-16.
- [12] Yang, X., H. Beyenal, G. Harkin, and Z. Lewandowski, Quantifying Biofilm Structure using Image Analysis, Journal of Microbiological Methods 39 (2000) 109-119.
- [13] Zahid, W. M., and Ganzarczyk, J. J., Fractal Dimension of the RBC Biofilm Structure, Wat. Sci. Tech., Vol. 29, No. 10-11, pp. 271-279, 1994.

To the Graduate Council:

I am submitting herewith a thesis written by Zhou Ji entitled "Quantitative Analysis of Biofilm Images Using Fractal Dimensions." I have examined the final copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science with a major in Computer Science.

Giri Narasimhan, Ph.D.
Major Professor

We have read this dissertation and
recommend its acceptance:

Dipankar Dasgupta, Ph.D.

David Lin, Ph.D.

Accepted for the Council:

Linda L. Brinkley, Ph.D.
Vice Provost for Research
& Dean of the Graduate School