



Quantifying structural heterogeneity in biofilms
by Wei Huang

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in
Computer Science

Montana State University

© Copyright by Wei Huang (1996)

Abstract:

In biofilm engineering research, quantifying the biofilm physical structure is important in understanding structure/function relations. To achieve this objective, we use digital image analysis techniques to analyze digital biofilm images collected with confocal scanning laser microscope (CSLM). A software package on Unix platform with source code written in C++ and Motif was developed. It has image segmentation tools and tools for extracting four quantitative structural parameters from the input biofilm images. The parameters are "2-D porosity," "average cluster size," "fractal dimension," and "texture entropy." The description of a structural feature of the image by each parameter agrees with human visual observation. An image reconstruction program is also written using an annealing algorithm. The program can be used to check whether this set of parameters captures all essential structural information of the original image.

QUANTIFYING STRUCTURAL HETEROGENEITY IN BIOFILMS

by

Wei Huang

A thesis submitted in partial fulfillment
of the requirements for the degree

of

Master of Science

in

Computer Science

MONTANA STATE UNIVERSITY-BOZEMAN
Bozeman, Montana

April 1996

N378
H85999

ii

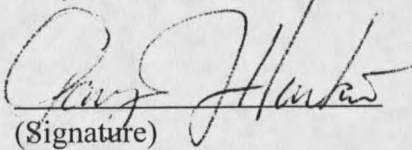
APPROVAL

of a thesis submitted by

Wei Huang

This thesis has been read by each member of the thesis committee and has been found to be satisfactory regarding content, English usage, format, citations, bibliographic style, and consistency, and is ready for submission to the College of Graduate Studies.

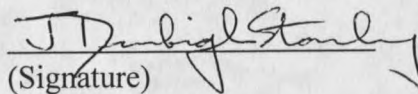
Gary Harkin, Chair


(Signature)

4/17/96
Date

Approved for the Department of Computer Science

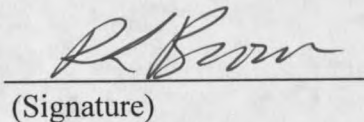
Denbigh Starkey, Head


(Signature)

4/17/96
Date

Approved for the College of Graduate Studies

Robert Brown, Dean


(Signature)

5/4/96
Date

STATEMENT OF PERMISSION TO USE

In presenting this thesis in partial fulfillment of the requirements for a master's degree at Montana State University-Bozeman, I agree that the Library shall make it available to borrowers under rules of the Library.

If I have indicated my intention to copyright this thesis by including a copyright notice page, copying is allowable only for scholarly purposes, consistent with "fair use" as prescribed in the U.S. Copyright Law. Requests for permission for extended quotation from or reproduction of this thesis in whole or in parts may be granted only by the copyright holder.

Signature

Wei Fenang

Date

4/17/96

ACKNOWLEDGMENTS

I would like to thank the Center for Biofilm Engineering, especially Prof. Zbigniew Lewandowski, and my graduate committee, especially Prof. Gary Harkin, for their support, guidance, and inspiration.

TABLE OF CONTENTS

	Page
1. INTRODUCTION	1
2. IMAGE COLLECTION AND IMAGE SEGMENTATION	4
2.1 Image Collection	4
2.2 Image Segmentation	4
3. MARKOV RANDOM FIELD MODEL.....	6
3.1 Basic Concepts	6
3.2 Meanings of Model Parameters	8
3.3 Compute Model Parameters from an Image	9
3.4 Construct Images from Model Parameters	9
3.5 Results and Discussion	11
4. FOUR STRUCTURAL PARAMETERS	13
4.1 Two-dimensional Porosity	13
4.2 Average Cluster Size	15
4.2.1 Euclidean Distance Mapping	16
4.2.2 Compute Average Cluster Size	19
4.3 Texture Entropy	21
4.4 Fractal Dimension	22
5. IMAGE RECONSTRUCTION	30
6. SOFTWARE DESIGN	32
6.1 Combination of Motif and C++	32
6.2 Software Design	33
7. SUMMARY AND FUTURE WORK	36
REFERENCES	37

LIST OF FIGURES

Figure	Page
1. Algorithm for Image Annealing with a Set of Markov Parameters	10
2. Markov Image Reconstruction	11
3. Compute Average Cluster Size for a circular cluster.....	14
4. Compute 2-D porosity of a Thresholded Biofilm Image	16
5. Compute Average Cluster Size of a Thresholded Biofilm Image	20
6. Compute Texture Entropy of a Biofilm Image.....	23
7. Compute Texture Entropy of another Biofilm Image	24
8. Compute Fractal Dimension of a Thresholded Biofilm Image	28
9. Compute Fractal Dimension of another Thresholded Biofilm Image	29
10. Image Reconstruction	31
11. Image Reconstruction	31

ABSTRACT

In biofilm engineering research, quantifying the biofilm physical structure is important in understanding structure/function relations. To achieve this objective, we use digital image analysis techniques to analyze digital biofilm images collected with confocal scanning laser microscope (CSLM). A software package on Unix platform with source code written in C++ and Motif was developed. It has image segmentation tools and tools for extracting four quantitative structural parameters from the input biofilm images. The parameters are "2-D porosity," "average cluster size," "fractal dimension," and "texture entropy." The description of a structural feature of the image by each parameter agrees with human visual observation. An image reconstruction program is also written using an annealing algorithm. The program can be used to check whether this set of parameters captures all essential structural information of the original image.

CHAPTER 1

INTRODUCTION

In recent studies biofilms have often been shown to be heterogenous, i.e., composed of voids, channels, and irregular clusters [1-7], instead of the simplistic uniform model. This raised a need of more study on biofilm physical structure. Biofilm structure study is also important in understanding structure/function relations. Some structure-related questions are: What are the characteristics of biofilm structures? How does this heterogenous structure evolve with time? How do the biofilm physiological processes affect physical structure? How do the physical structure variables relate to other variables such as the local mass transfer parameter and the fluid velocity? To answer these questions, quantifying biofilm structure is a necessary and essential step.

Various efforts have been directed to biofilm structure study. To visualize biofilm physical structure, traditional light microscopy [1], atomic force microscopy [2], and confocal scanning laser microscope (CSLM) [3,6], etc. have been used. Compared with other microscopies, CSLM has several advantages such as nondestructiveness and three-dimensional capability.

Although much visualization work on biofilm has been carried out, quantification of biofilm structure is in a state of insufficiency. The descriptions of biofilm structure in most of the above studies are qualitative. Among a few quantitative studies, Zahid first used fractal dimension to describe biofilms [9], and Zhang studied density, porosity, specific surface area and mean pore radius of biofilm [10]. Both of them used experimental techniques such as micro-

slicing in the quantification of biofilm structure. This has a serious drawback, in that the measurement is not efficient, is destructive, and cannot be combined with other in-situ experiments.

S.W. Hermanowicz's group at Berkeley studied a fractal dimension parameter of CSLM images of biofilms [23]. He computed fractal dimensions of images and investigated its variance with scales, depth, and flow direction. The approach, analyzing CSLM biofilm images with digital image analysis techniques in order to quantify biofilm structure, will also be our approach. Apparently, a single parameter of fractal dimension alone will not suffice for the quantification of biofilm structure. In our work we studied more parameters, each of which captures a unique feature of the biofilm structure, and developed a software package that performs the image analysis.

Our goal for quantifying biofilm structures is to obtain a set of parameters that contain enough structural information such that the reconstructed image, using the set of parameters, is visually similar to the original image.

In the following chapters, after a brief description of image collection and image segmentation steps in Chapter 2, we describe in Chapter 3 a popular image modeling approach that we attempted. However the result shows that it is not very suitable for biofilm study. Then Chapter 4 presents other image processing approaches, each of which extract a certain structural parameter. The description of an image feature by each parameter agrees with human visual examination. Chapter 5 describes a program that can synthesize images with desired parameter values. We can use this kind of program to check whether a set of parameters can capture *all* essential features of an image. Chapter 6 describes some software design issues. Chapter 7

summarizes the work that has been done and provides a few suggestions for future work.

CHAPTER 2

IMAGE COLLECTION AND IMAGE SEGMENTATION

2.1 Image Collection

A transparent biofilm chemical reactor with biofilm on it is mounted on the CSLM. The CSLM is controlled by a software that can digitize and store images at one or more depths in the sample. The image files can be converted to 768×512 8-bit gray-scale TIFF format files.

2.2 Image Segmentation

To differentiate focused biofilm clusters from surrounding voids in a CSLM biofilm image, image segmentation needs to be done. For biofilm pictures, the boundary between cluster and void is indistinct, and the most prominent difference between cluster and void is the difference of brightness. Therefore thresholding seems to be the best technique of segmentation.

Thresholding can be done interactively by instantly viewing thresholded results using various threshold values and choosing the best. The histogram of the image provided in the software can be used to select the best threshold value, if the histogram is bimodal. Usually, for a set of biofilm images in an experiment, CSLM setups are fixed, so threshold value should be the same for all images. In this situation, after interactively choosing the correct threshold, users can create a simple script file to batch-process a set of images. The script repeatedly uses a separate and independent executable file to process image files.

Although the software provides some image segmentation tools, it does not elaborate on

this phase. The purpose of segmentation is to get a binary cluster/void image for the next-step's parameter extraction. Therefore image segmentation tools in other existing software can also be used to achieve this result.

CHAPTER 3

MARKOV RANDOM FIELD MODEL

In order to find a set of parameters that contain essential structural information and that can be used to reconstruct similar images, we studied an image modeling technique. In the image processing field, image modeling involves the construction of models or procedures for the specification of images. These models serve a dual role in that they can describe images that are observed and also can serve to generate synthetic images from the model parameters.

Markov random field (MRF) texture model is one of the well known image texture models and is among the most powerful stochastic texture models.

3.1 Basic Concepts

The brightness level at a point in an image is correlated with the brightness levels of neighboring points unless the image is simply random noise. Markov random field is a precise model describing this correlation.

Let $X(i,j)$ denote the brightness level at a point (i,j) on the $N \times N$ lattice L . Sometimes we simplify the labeling of the $X(i,j)$ to be $X(i)$, $i=1,2,\dots,M$ where $M = N^2$.

Definition 1: Let L be a lattice with G levels. A *coloring* of L denoted X is a function from the points of L to the set $\{0,1,\dots,G-1\}$. The notation $\mathbf{0}$ denotes the function that assigns each point of the lattice to 0.

Definition 2: The point j is said to be a neighbor of the point i if:

$$p(X(i)|X(1),X(2),\dots,X(i-1),X(i+2),\dots,X(m))$$

depends on $X(j)$. (In most applications, we assume the neighbors of a point are its physically close points.)

Definition 3: A *Markov random field* is a joint probability density on the set of all possible colorings $\mathbf{X}$ of the lattice L subject to the following conditions:

1) Positivity: $p(\mathbf{X}) > 0$ for all $\mathbf{X}$.

2) Markovianity: $p(X(i)|\text{all points in the lattice but } i)$

$$= p(X(i)|\text{neighbors of } i)$$

3) Homogeneity: $p(X(i)|\text{neighbors of } i)$ depends only on the configuration of neighbors and is translation invariant with respect to translation with the same neighborhood configuration).

The Hammersley-Clifford theorem and Besag's work [13] provide a formulation relating the probability of a point $X(i,j)$ having gray level k with the parameter determined by its neighbors, as the following shows.

The probability $p(X(i,j)=k|\text{neighbors of } (i,j))$ is binomial with parameter $\theta(T)$ and $G-1$, where G is the number of gray levels. The value of T is given in (1)-(4) for MRF models of various orders. The a and $b(m,n)$ are the parameters of the model and are 0 for all i larger than the order.

$$\theta = \frac{e^T}{1 + e^T}$$

where a first-order model has the form for T :

$$(1) \quad T = a + b(1,1)[X(i-1,j) + X(i+1,j)] + b(1,2)[X(i,j-1) + X(i,j+1)].$$

A second-order model has a T of the form:

$$(2) \quad T = a + b(1,1)[X(i-1,j) + X(i+1,j)] + b(1,2)[X(i,j-1) + X(i,j+1)] \\ + b(2,1)[X(i-1,j-1) + X(i+1,j+1)] + b(2,2)[X(i-1,j+1) + X(i+1,j-1)].$$

A third-order model takes the form:

$$(3) \quad T = a + b(1,1)[X(i-1,j) + X(i+1,j)] + b(1,2)[X(i,j-1) + X(i,j+1)] \\ + b(2,1)[X(i-1,j-1) + X(i+1,j+1)] + b(2,2)[X(i-1,j+1) + X(i+1,j-1)] \\ + b(3,1)[X(i-2,j) + X(i+2,j)] + b(3,2)[X(i,j-2) + X(i,j+2)].$$

A fourth-order model is obtained by adding an additional term of the form

$$(4) \quad b(4,1)[X(i-1,j-2) + X(i-2,j-1) + X(i+1,j+2) + X(i+2,j+1)] \\ + b(4,2)[X(i+1,j-2) + X(i+2,j-1) + X(i-1,j+2) + X(i-2,j+1)]$$

to the form for the third-order T . Additional high-order terms can be obtained by extending the orders in a similar way.

For binary image case that we are interested in, we obtain the conditional probability:

$$p(X=x | neighbors) = \frac{e^{xT}}{1 + e^T}$$

where x is one of the $\{0,1\}$.

3.2 Meanings of Model Parameters

Each parameter of the set $\{a, b(m,n)\}$ does not independently correspond to a certain

image feature. In fact, the set of several parameters together determines an infinite number of images with some common features. The relationships between those common features and parameter values are indirect and so are usually not easy to decide.

We can specify some of these relationships as follows. If $b(1,1)$ and $b(1,2)$ are positive and a is negative, then the bigger $b(1,1)$ and $b(1,2)$, the bigger the clusters in an image. If $b(1,1)$ is not equal to $b(1,2)$, the image is anisotropic.

3.3 Compute Model Parameters from an Image

The method used to estimate the parameters $\{b(i,j)\}$ is the maximum likelihood estimation. Let $p(x|.)$ denote the conditional probability $p(X(i,j)=k|\text{neighbors of } (i,j))$. The usual log likelihood is given by:

$$A = \sum_X \ln(p(X|.))$$

where the summation extends over all points of the lattice.

To maximize A , we must find the optimal solution of a multi-variable function. We used the “downhill simplex method” algorithm from the book “Numerical Recipes in C” [17]. For all of the sample images in Cross’s paper [12], our program gives correct parameter values.

3.4 Construct Images from Model Parameters

Theorem: Let X and Y be two colorings of the Markov random field lattice L . Then

$$\frac{p(Y)}{p(X)} = \prod_{i=1}^M \frac{p(X(i)=y(i)|X(1),X(2),...,X(i-1),Y(i+1),...,Y(N))}{p(X(i)=x(i)|X(1),X(2),...,X(i-1),Y(i+1),...,Y(N))}$$

Other mathematical theorems [12] along with the above theorem guarantee that the application of the algorithm in Fig. 1 will eventually result in a lattice with desired Markov field.

```

while not STABLE do
  begin
    choose sites X(1), X(2) with X(1) <> X(2);
    r := P(Y) / P(X);
    If r >= 1
      then switch X(1), X(2)
      else
        begin
          u := uniform random on [0,1];
          If r > u
            then switch X(1), X(2)
            else retain X
          end
        end
      end
    end
  end
end

```

Figure 1. Algorithm for image annealing with a set of Markov parameters.

The algorithm takes an arbitrary image that has the desired histogram and the set of desired parameters as input and then begins the annealing process. Initial configurations affect the rate at which equilibrium is reached.

We use an image with random pixel values as input. Experiments by other authors and our own showed that the annealing should take at least $10M$ attempted switches between pixels with different colors, where M is the size of image. Usually we use $100M$ attempted switches. The equilibrium is manifested when more attempts change the appearance of the image little. The measured parameter values of the output image are very close to the desired parameters, with difference usually less than 10%.

3.5 Results and Discussion



Figure 2. Markov image reconstruction. (a): an original image, (b): the generated image.

Fig. 2 shows original and generated pictures, using the first-order Markov method. The measured parameters for the original image are: $a = -22.269$, $b(1,1) = 18.809$, and $b(1,2) = 3.167$. The measured parameters for the generated image are: $a = -24.433$, $b(1,1) = 20.199$, and $b(1,2) = 3.597$. We can see that the generated image grasps these properties of the original image: general sizes of clusters and directionality. (But note that horizontal directionalities of the two images have different causes: one is due to the arrangement of clusters and the other is due to the shapes of clusters.)

From the above figure we may speculate that if we use higher order parameters, we may achieve satisfactory results. However, from our tests, we found that higher order parameters behave quite unreliably and unpredictably. For example, given a set of parameters, we generate an image; measuring the image shows that the first order parameters are close to the original parameters, but the higher order ones differ from the original ones completely.

Besides the above drawback, from our tests and from the nature of the Markov method,

we can see the following limitations of the technique for use in the quantification of biofilm structure. (1) Due to the homogeneous nature of the model, if, in the original image, there are a few very small clusters clustering in a subregion, in the generated image those very small clusters will be spread out through the whole image. (2) This method is standalone, that is, we cannot incorporate other parameters into this method. (3) There is no direct physical meaning for any one of the set of parameters. The parameters are dependent with each other.

In conclusion, we think the Markov model is not a good way for quantifying biofilm structures, although it does capture certain features of an image and it has the reconstruction ability.

CHAPTER 4

FOUR STRUCTURAL PARAMETERS

We developed four other parameters to quantify biofilm images. Two of these parameters, porosity and average size, have straightforward morphological meanings. They directly describe certain geometric features. The other two parameters, texture entropy and fractal dimension, are more abstract and describe image features at higher levels.

For each of these parameters, we tested with a few real biofilm images. The description of image features by each parameter agrees with human visual judgment. The software will be used by other engineering researchers in the CBE, and at that time, with extensive experimental data, those parameters will relate to the concrete research problems in biofilms.

4.1 Two-dimensional Porosity

Since biofilms often consist of voids and channels, there should be a parameter describing such porous effect, which porosity does. It is the ratio of void volume to the total volume for an object. Straightforwardly, we define 2-D porosity as the ratio of void area to total area for a cross section plane and use this parameter as an indicator of porousness. The 2-D porosity is closely related to 3-D porosity. If we have a set of images scanned at different depths above the same spot, the average of 2-D porosities of these planes differs from the 3-D porosity only by a constant factor. Fig. 3 shows a biofilm image with the computed 2-D porosity displayed as 0.217.

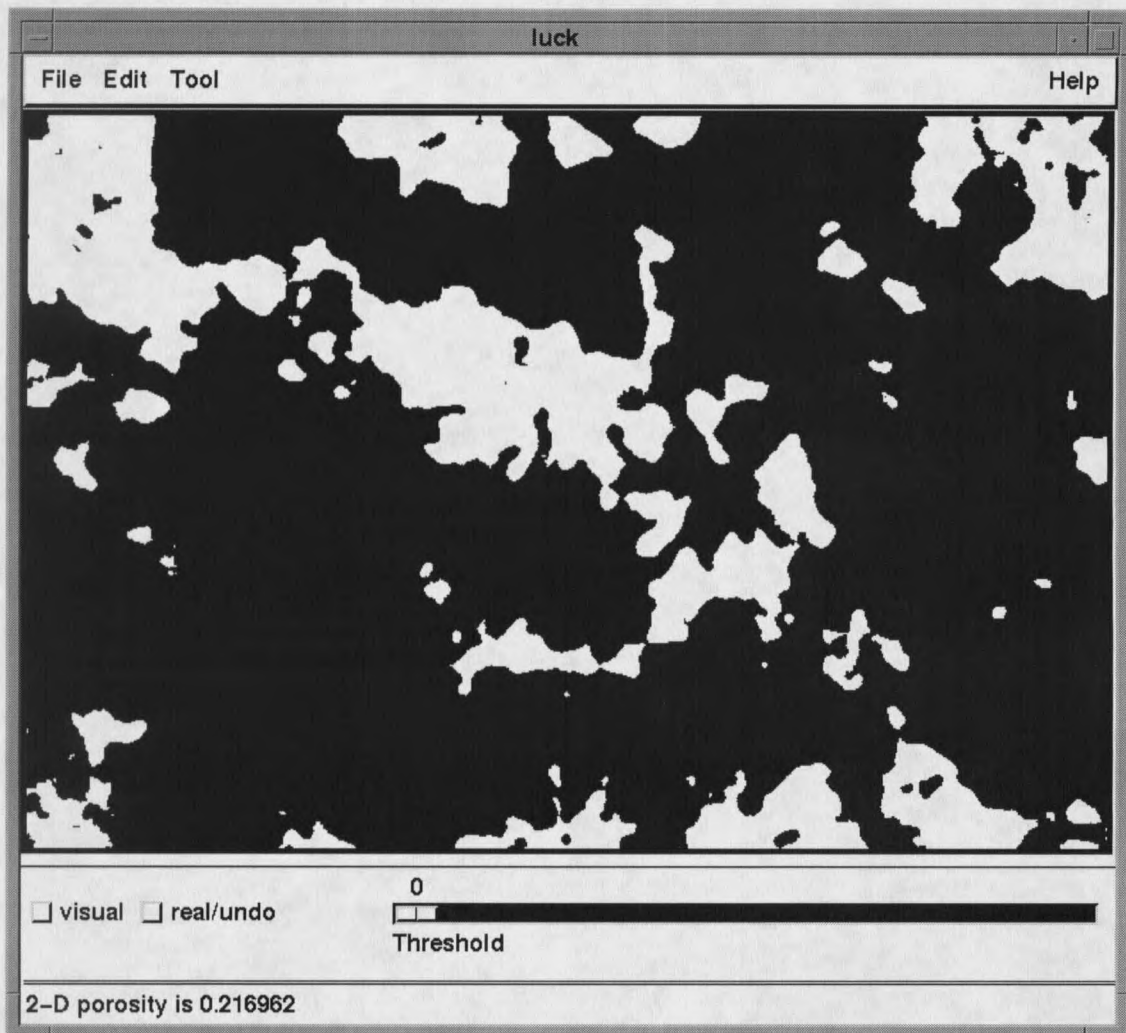


Figure 3. Compute 2-D porosity of a thresholded biofilm image

4.2 Average Cluster Size

As we can see in Fig. 3, biofilms are usually very irregularly shaped. They are often interconnected and have filament-like components. Usual descriptions of sizes such as diameter are based on circle-like shapes and are not appropriate for many biofilms. To obtain a value that describes the average size of biofilm clusters, we defined a unique parameter. The definition was inspired by chemical diffusion processes. For each cluster pixel we compute its shortest distance to cluster/void borders and then average over all pixels to get an average distance. The average distance thus obtained is not only directly related to cluster size, but also suggests in average how much depth the exterior chemicals need to penetrate to reach the interior microorganisms. Therefore, the parameter is ready to be combined with other studies on biofilm such as the mass transfer coefficient study.

To get a quantitative sense of the average cluster size parameter, let us calculate it for an input of a circle with radius R in Fig. 4:

$$average\ size = \frac{1}{\pi R^2} \int_0^R 2\pi r(R-r)dr = \frac{R}{3}$$

It is not surprising that the average size is proportional to the radius. As a digression, the above calculation was also used in the verification of our software.

It seems computationally very complex to compute the average cluster size. The Euclidean Distance Mapping algorithm [15] greatly reduces computational burden of our problem.

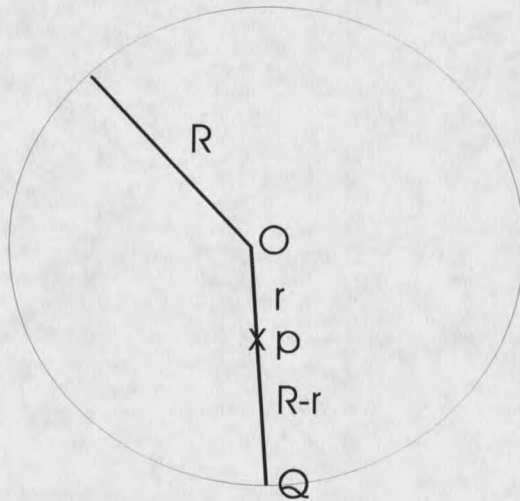


Figure 4. Compute average cluster size for a circular cluster. R is the radius. $R-r$ is the shortest distance to the border from the point P .

4.2.1 Euclidean Distance Mapping

In two dimensional rectangular space, several distance metrics are defined, in which, metric

$$d_e((i,j),(h,k)) = \sqrt{(j-i)^2 + (k-h)^2}$$

is called Euclidean metric d_e . Obviously, we assume that (i,j) and (h,k) are two points on the two dimensional space.

Given a binary image with two set of pixels,

S = set of 1's, the objects

$\underline{S}$ = set of 0's, the background

a distance map $L(S)$ is an image such that for each pixel $(i,j) \in S$, there is a corresponding pixel in $L(S)$ where

$$L(i,j) = \min(d[(i,j), \underline{S}])$$

i.e., each pixel in S has been assigned a label in $L(S)$ that amounts to the distance to the nearest background $\underline{S}$. Obviously we can define a similar map $L(\underline{S})$ for the background. The computational procedure $L: S \rightarrow L(S)$ is called distance mapping.

There have been efficient algorithms for distance maps based on the metrics d_4 and other quasi-Euclidean metrics. The efficient algorithm for true Euclidean distance mapping did not occur until 1980 [15]. With this algorithm the seemingly computationally complex problem can be solved more easily.

The algorithm operates on the picture L , which is a two dimensional array with the elements

$$L(i,j) \quad 0 \leq i \leq M-1, \quad 0 \leq j \leq N-1.$$

Each element is a two-element vector

$$\mathbf{L} = (L_i, L_j), \quad L_i, L_j \text{ being positive integers,}$$

$$\mathbf{L}(i,j) = (L_i(i,j), L_j(i,j)).$$

The size of a vector $\mathbf{L}(i,j)$ is defined by

$$|\mathbf{L}(i,j)| = \sqrt{(L_i^2 + L_j^2)}.$$

The four-point sequential Euclidean distance mapping algorithm (4SED) is defined as

follows.

Initially:

$$L(i,j) = (0,0) \quad \text{if } (i,j) \in S,$$

$$L(i,j) = (Z,Z) \quad \text{if } (i,j) \in \underline{S}$$

where Z is the largest integer that can be stored without inconvenience and where S is the object and $\underline{S}$ is the background in the original $M \times N$ binary picture.

First picture scan:

For $j = 1, 2, \dots, N-1,$

for $i = 0, 1, 2, \dots, M-1,$

$$L(i,j) = \min(L(i,j), L(i,j-1) + (0,1);$$

for $i = 1, 2, \dots, M-1,$

$$L(i,j) = \min(L(i,j), L(i-1,j) + (1,0);$$

for $i = M-2, M-3, \dots, 1, 0,$

$$L(i,j) = \min(L(i,j), L(i+1,j) + (1,0).$$

Second picture scan:

For $j = N-2, N-3, \dots, 1, 0,$

for $i = 0, 1, 2, \dots, M-1,$

$$L(i,j) = \min(L(i,j), L(i,j+1) + (0,1);$$

for $i = 1, 2, \dots, M-1,$

$$L(i,j) = \min(L(i,j), L(i-1,j) + (1,0);$$

for $i = M-2, M-3, \dots, 1, 0,$

$$L(i,j) = \min(L(i,j), L(i+1,j) + (1,0).$$

The algorithm first assigns a maximum label to all background pixels. The real assignments take place in two scans of the complete picture where each picture scan involves for each line j , first a one-step propagation in the j -direction, then a sequential propagation in both i -directions.. Each label $L(i,j)$ is involved in six comparisons.

Detailed analysis of the algorithm shows that it produces a distance map that is error-free except for very sparsely scattered pixels that may be assigned a distance label with an absolute error less than 0.29 pixel units.

4.2.2 Compute Average Cluster Size

In the first step, we need to threshold the input biofilm image into a binary image. In the second step, we apply the above Euclidean distance mapping algorithm and get a distance map for the cluster regions. In the third step, for those regions, we sum up distance values over all pixels and divide it by the total pixel number.

Fig. 5 shows a thresholded biofilm image with computed average cluster size value displayed at the bottom of the window. Biofilm image in Fig. 5 is thresholded and then filtered from the original biofilm image in Fig. 6. (The black parts are clusters.) The unit of the size is one pixel. By knowing the magnification setup of CSLM, we can convert this value to the actual length.

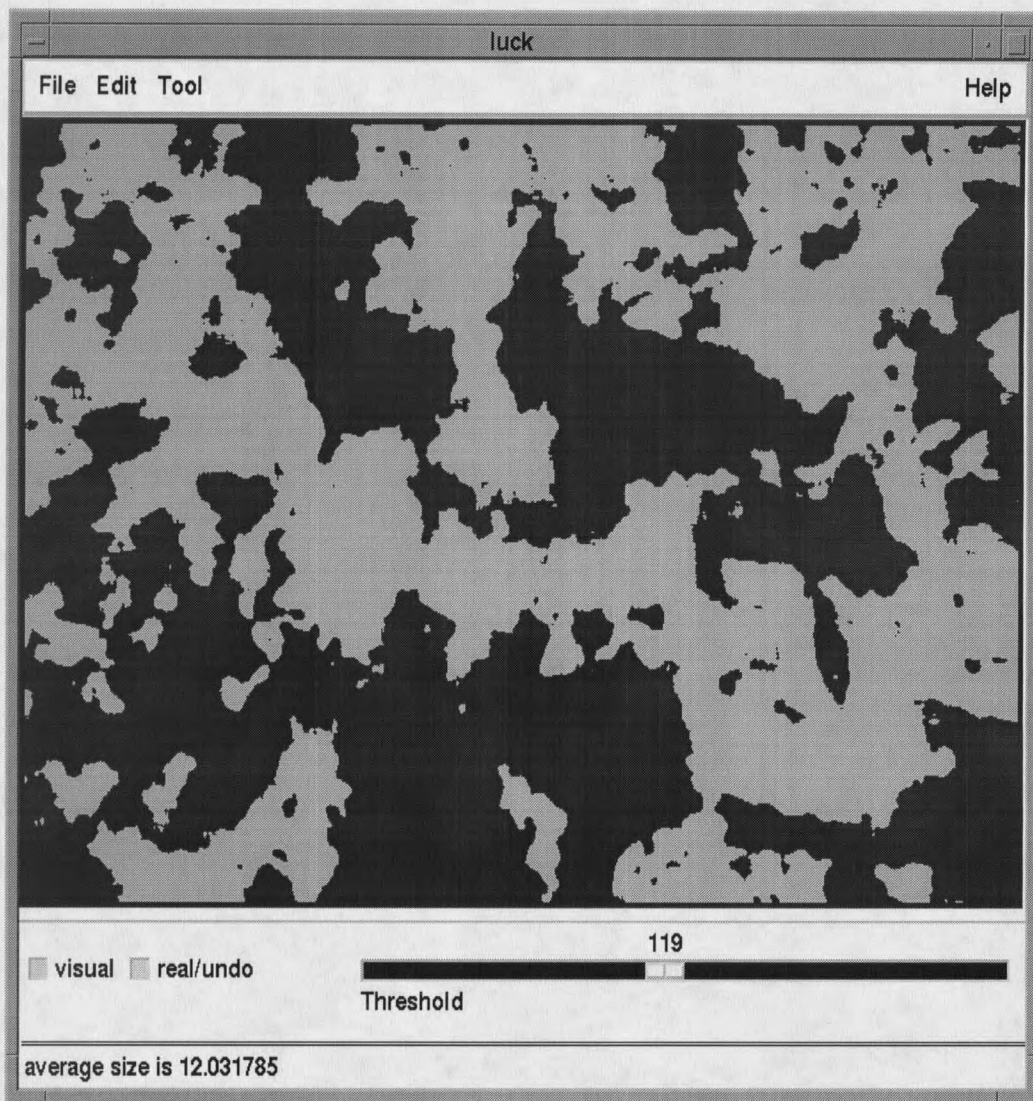


Figure 5. Compute average cluster size of a thresholded biofilm image.

4.3 Texture Entropy

Texture is an important characteristic for the analysis of many types of images. Texture can be described as being fine, coarse, smooth, rippled, irregular, etc. Texture is an innate property of virtually all surfaces -- the grain of wood, the pattern of crops in a field, etc. It contains information about the structural arrangement of surfaces and their relationship to the surrounding environment.

Although it is quite easy for human observers to recognize and describe in empirical terms, texture resists precise definition and analysis by digital computers. Gray-scale co-occurrence matrices are a successful second-order statistical method for quantifying textures [14].

Suppose a gray-scale image $f(x,y)$ have G gray-levels. Let $\mathbf{d} = (\Delta x, \Delta y)$ be a vector in the (x,y) plane. The gray-scale co-occurrence matrix p_d is G by G array, where $p_d(i,j)$ is the probability of the pair of gray levels (i,j) occurring at separation $\mathbf{d}$. A variety of measures can be employed to extract texture information of the original image from the p_d matrix. Texture entropy is one of them.

$$\text{Texture Entropy: } -\sum_i \sum_j p(i,j) \log(p(i,j))$$

$$\text{Correlation: } \frac{\sum_i \sum_j (ij)p(i,j) - \mu_x \mu_y}{\sigma_x \sigma_y}$$

where $p_x(i) = \sum_{j=1}^G p(i,j)$, $p_y(i) = \sum_{i=1}^G p(i,j)$, μ_x , μ_y , σ_x and σ_y are the means and standard

deviations of p_x and p_y .

“Texture entropy” is a measure of how much order exists within an image. For a truly random image, it reaches maximum, and for a uniform single color image, it is the minimum value of zero. It is also a normalized parameter, which means that the parameters from images of different sizes are comparable.

Fig. 6 and Fig. 7 show two biofilm images with different texture entropies. We used the full 256 gray-scale level images as inputs for computation. Certainly, we can also use binary image as input. From the figures we can see that, by describing degrees of order, texture entropy seems to be one of the discriminators of different types of biofilm structures.

4.4 Fractal Dimension

Fractal theory, developed by Mandelbrot [18], provides a completely new approach to the characterization of many natural and engineered systems that have no definite form or regularity. Fractals occupy a borderline between Euclidean geometry and complete randomness. Many natural geometries have fractal property, and as a phenomenological description of geometry, fractal has been applied to many different fields of applications. For example, fractals are used in computer graphics for generating natural-looking objects.

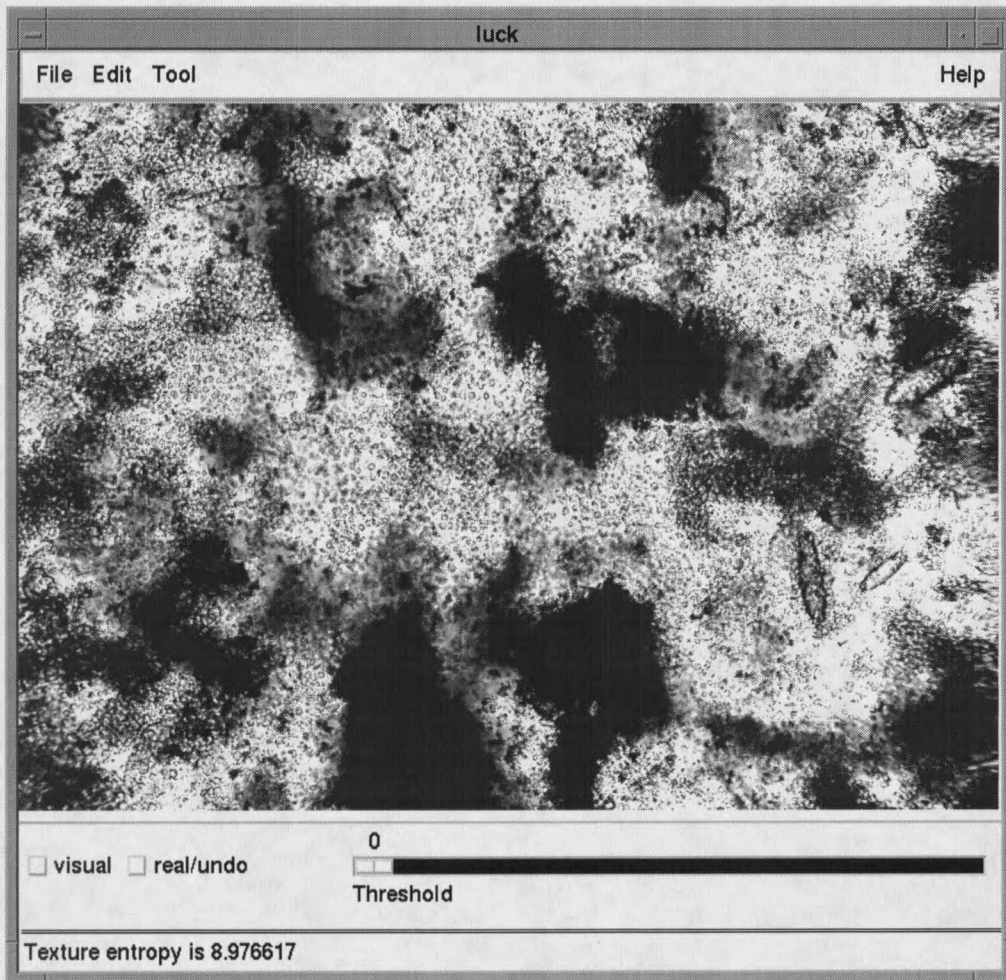


Figure 6. Compute texture entropy of a biofilm image.

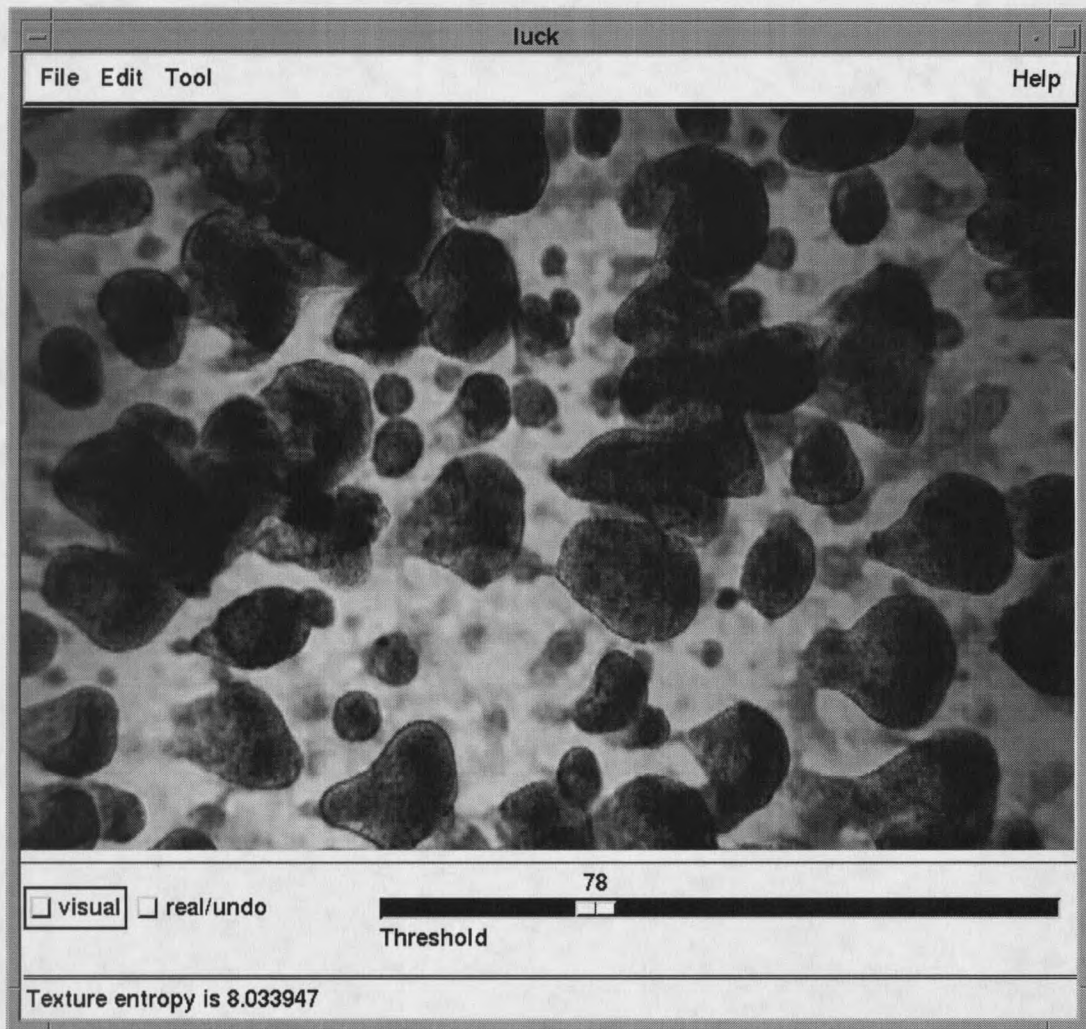


Figure 7. Compute texture entropy of another biofilm image.

The most important numerical parameter in fractal theory is the Hausdorff or fractal dimension. As an extension and generalization of the classical concept of Euclidean dimensions, the fractal dimension preserves the fundamental dimension roles in the measurements where the ordinary dimensions are used as exponents. For example, the volume V of a fractal with fractal dimension D is proportional to its size R raised to power D :

$$V(R) \propto R^D$$

The above equation differs from an ordinary volume-size relationship only in that the power coefficient D is no longer limited to integers. For a fractal dust-line, which is a line composed of separated small line segments, its fractal dimension is between 0 to 1. For a two-dimension or three-dimension fractal sponge-like geometry, its fractal dimension is between 1 to 2 or between 2 to 3, respectively. The more hollow the object, the smaller its fractal dimension.

There is another type of fractal geometry. A rugged line has a fractal dimension between 1 to 2. A rugged surface has a fractal dimension between 2 to 3. The more rugged the line or surface, the bigger its fractal dimension. Besides ruggedness or hollowness, fractal also describes the self-similar property of a geometrical object. For example, the coastlines of Britain looks similar at different magnifications.

There are many different methods to measure fractal dimensions. Different methods usually get slightly different results for the same geometry, but asymptotically approach the same limit for a pure fractal input. Generally, in various measurement methods, fractal dimension almost always associates with the slope of some logarithmic plot, whose one axis is the scaling unit and the other axis is some countable parameter.

In our problem of study, we want to measure the fractal dimension of biofilm cluster surfaces. Measuring fractal dimension of a surface is not an easy task. However, there is a widely-believed proposition speculated by Mandelbrot that intersecting a fractal surface of dimension $2.d$ with a plane will produce an intersection boundary whose dimension is exactly $1.d$. Therefore, we can just measure the fractal dimension of outlines of clusters in a scanned cross-sectional image of biofilm. In other words, an input image is first thresholded into a binary image consisting only of clusters and voids. Then we measure the fractal dimension of those border lines between clusters and voids.

We employ a method called mosaic amalgamation in measuring the fractal dimension of the boundary [18]. The method progressively coarsens the image representation. The original image gives the finest scale approximation to the boundary length, and larger scale approximations which do not make use of all of the available image resolution are used to construct log-log plots of total boundary length vs. measurement scale. At each amalgamation step, the pixels from the previous image are averaged together to form larger pixels. The larger pixels ignore the fine details of the boundary, and so the measured perimeter will decrease. For a fractal object, it is expected that a log-log plot of the perimeter against the size of the pixels used will show the usual straight line, whose slope's absolute value gives the fractal dimension.

It is appropriate here to mention the difference of approaches between ours and that of S. Hermanowicz [11] in measuring biofilm fractal dimension. We measure the fractal dimension of the biofilm cluster's surface, while they actually studied biofilm cluster's aggregation fractal property.

Application of fractal dimension parameter on real biofilm pictures agrees with visual

judgement of the ruggedness of biofilm cluster outlines. The bigger the fractal dimension, the more rugged the outline. Fig. 8 and Fig. 9 show two segmented biofilm images with different fractal dimension values.

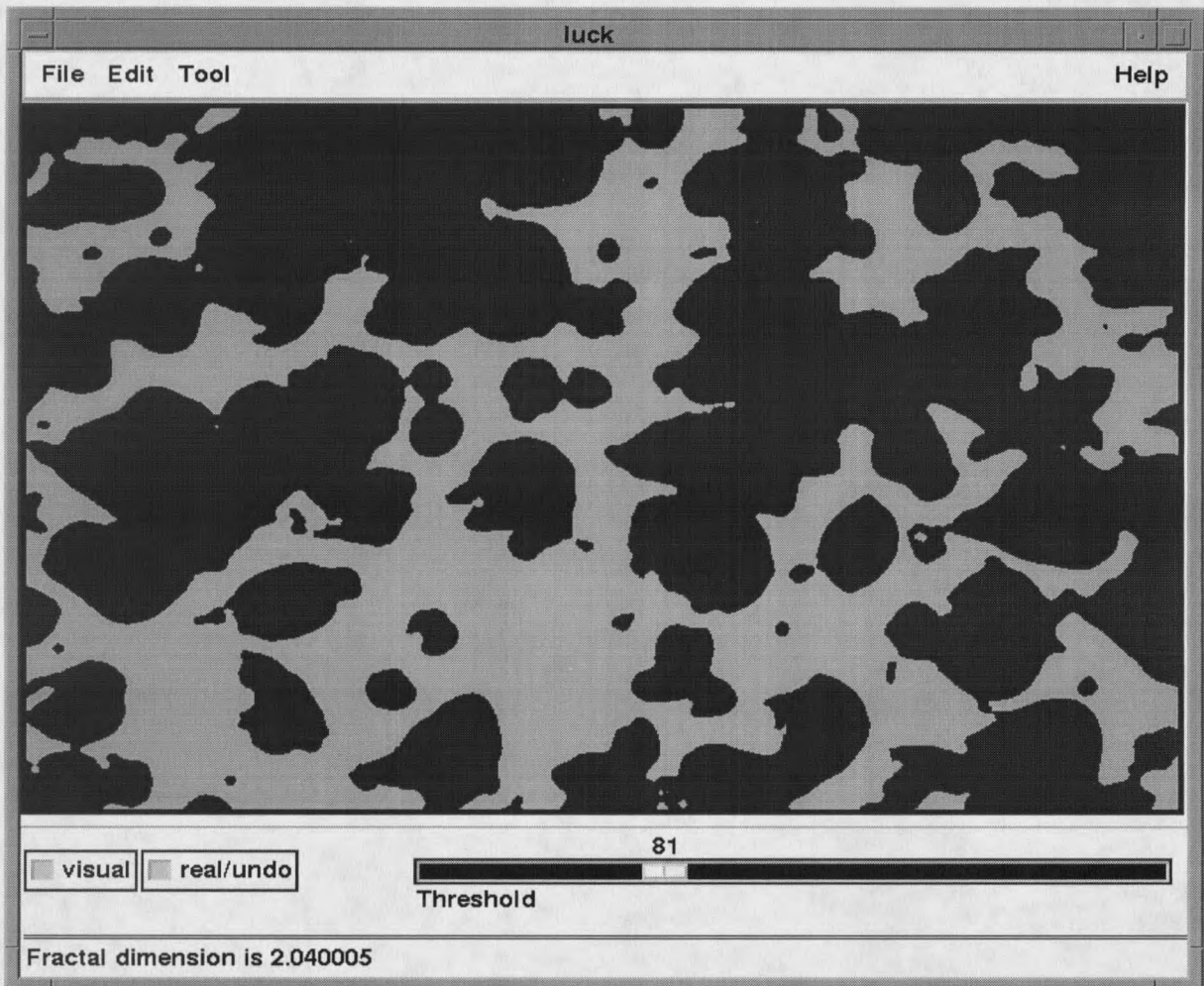


Figure 8. Compute fractal dimension of a thresholded biofilm image.

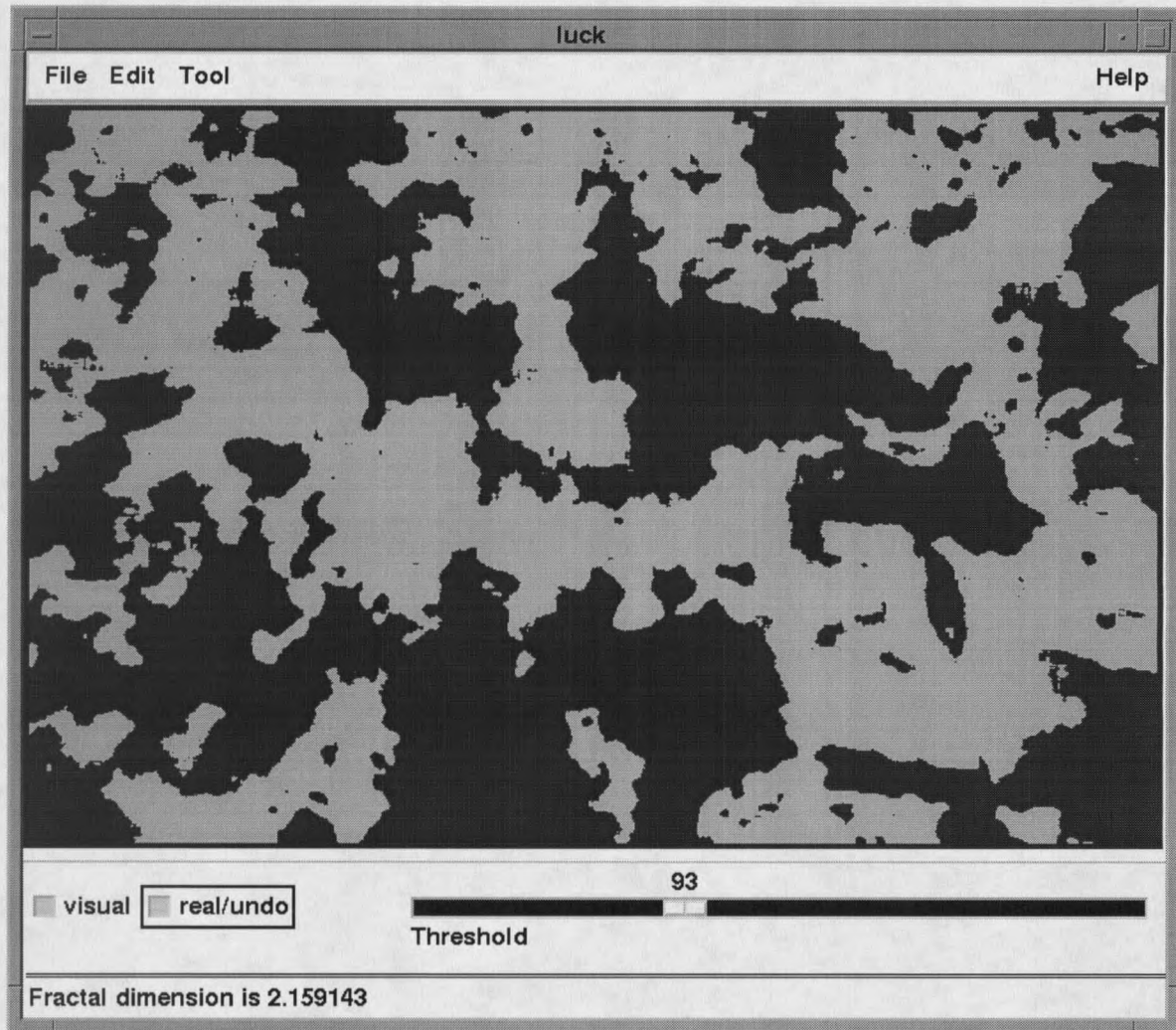


Figure 9. Compute fractal dimension of another thresholded biofilm image.

CHAPTER 5

IMAGE RECONSTRUCTION

In this chapter we describe a program that uses an annealing algorithm to synthesize images from a random-dust input image with desired black/white ratio. The program uses average cluster size and entropy as controls of annealing. The values of the two parameters for generated image agree with the desired values.

The annealing method is as follows. First we use the Markov simulated annealing algorithm with infinite parameter values to form spread small clusters. As shown in Fig. 8 and Fig. 9, for a small black rate, the intermediate image consists of separated small clusters, and for near the 0.5 black rate, the intermediate image consists of interconnected thin clusters. Then we randomly choose two clusters to merge them together. Marking clusters is accomplished by spreading pixels around a randomly selected cluster pixel. This marking algorithm does not necessarily cover all connected regions. Merging is done by eliminating the smaller cluster and augmenting the other cluster along a direction randomly chosen from these four directions: up, down, left, and right.

Fig. 8 and Fig. 9 show the results of this reconstruction program for two inputs. We can see the algorithm is able to reconstruct an image with the parameters reasonably close to the desired ones. We can see that based on the similar values of entropy and average cluster size, the reconstructed image can be visually very dissimilar to the original one. (Here we didn't consider the fractal dimension parameter in image reconstruction, since it only affects borders.) This tells

us that the three parameters, porosity, average cluster size, and entropy, are not containing all essential structural information.

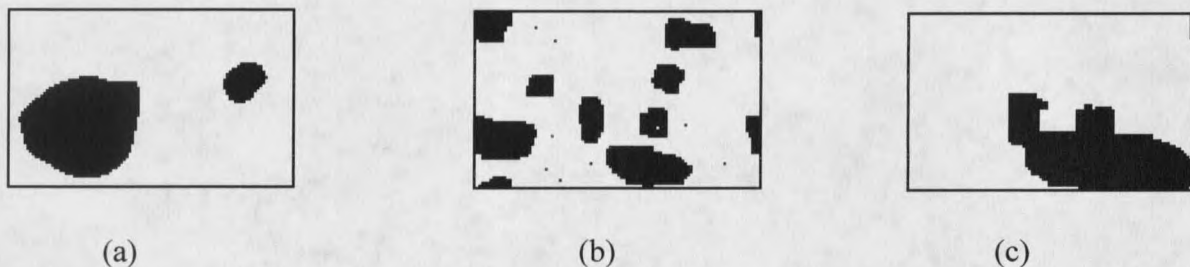


Figure 10. Image reconstruction. (a), (b), and (c) are original, intermediate after Markov simulation, and output images, respectively. Black rate for these images is 0.22. Average cluster sizes for (a), (b), and (c) are 5.84, 2.64, and 5.69, respectively. Texture entropies are 0.605, 0.675, and 0.607, respectively.

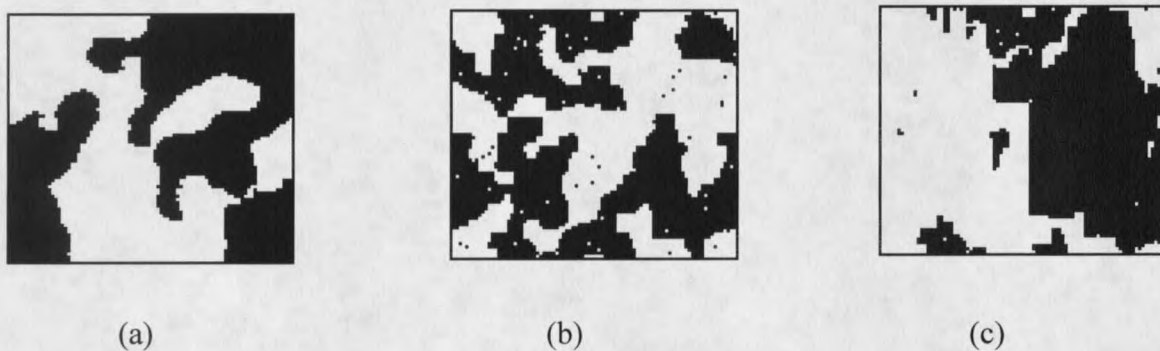


Figure 11. Image reconstruction. (a), (b), and (c) are original, intermediate after Markov simulation, and output images, respectively. Black rate for these images is 0.51. Average cluster sizes for (a), (b), and (c) are 5.70, 2.48, and 5.73, respectively. Texture entropies are 0.863, 0.972, and 0.896, respectively.

CHAPTER 6

SOFTWARE DESIGN

6.1 Combination of Motif and C++

Let's first examine some properties of the tools, Motif and C++. Motif itself is built upon certain object-oriented principles. Some examples are that each widget is either parent or child of another widget, and callback functions are associated with respective widgets.

There can be several ways to combine C++ class concepts with Motif. First, a widget can be wrapped up with a C++ class. All callback functions of that widget are the class's member functions. Second, a C++ class can describe a conceptual object. Such an object is composed of several widgets as well as other members and performs a certain conceptual task. For example, the "Menu" class encapsulates the conceptual menu object. In our software, we used both of the above techniques.

Since Motif existed before C++ became popular, one problem appears in putting Motif callback functions into C++ member functions. The problem is that, in a C++ member function, there is an implicit first argument, "*this*" pointer. Therefore we cannot directly let a callback function be an ordinary member function. A trick to solve this problem is to declare the callback function to be a static member function and pass "*this*" pointer into the callback function through its standard fourth argument. In this way, since the callback is a static member, the implicit "*this*" argument problem is solved. And although the callback is a static member, it is associated with a concrete object because of the "*this*" pointer in its explicit argument.

6.2 Software Design

Because of the research and open-ended nature of the project, the overall architectural design of the software cannot be done at the beginning. In fact, during the earlier period of the project, after working out TIFF input/output utility modules, we wrote one main file for each image processing technique. An independent executable can be obtained by linking a main object file and a TIFF i/o module. Such an executable takes an image as input and processes it with a certain technique.

Overall design began in the middle stage of the project. A GUI design organizes every part of the software project. Our graphical interface is not complex. The whole window is of fixed size and is divided into three sections: menus, canvas, and message. Each of these sections, and also the whole window, is a C++ object. The menu contains cascade buttons such as "File," "Edit," "Tool," and "Help." The canvas is to display the image we are processing. The message section has several thresholding controls for inputting messages and a label widget for displaying output messages.

In "File" menu, there are "Open," "Close," and "Save as" buttons. The callback for the "Open" button invokes a file selection dialog, in which the "Ok" button triggers a callback function that takes the selected image file as input, initializes a global array containing image data, and initializes a pixmap for the canvas to display. The callback for the "Close" button deallocates unneeded memory. If the memory leak is serious, after a few opens and closes, memory will be used up.

The "Edit" menu contains buttons such as "Second window," "Filter," "Paint black," and "Undo." "Second window" is to pop up a second window containing a user-selected rectangular

subregion of the image. In that second window, the same parameter extracting tools can be used for processing the sub-image. This second window is implemented with another Motif shell, which is a pop-up shell. Selecting a subregion is done by using the well-known rubber band technique in the canvas widget. The information of the subregion, such as coordinates, is contained in a global object so that it can be seen by other programs. Other buttons in the “Edit” menu are each associated with less complex callback functions.

The “Tool” menu contains the four buttons extracting the previously mentioned four parameters. The computed result will display on the label widget in the message section of the window. In order for the tool callbacks to be able to process both original image and subimage, the callbacks only use minimum/maximum row/column information of the image data in processing. The “Tool” menu also contains a “Histogram” button, which can pop up a histogram diagram for the image. The implementation also uses a pop-up shell.

The canvas section is represented by a canvas class, whose members include a *xmDrawingAreaWidgetClass* widget for displaying image, graphic context information, pixmap, and callback functions such as “redisplay” and “resize” callbacks.

The thresholding control part of the message section has three controls: a “visual” toggle button, a “real” toggle button, and a threshold scale bar. The “visual” button toggles thresholding on and off, but the thresholding happens only in a visual fashion, that is, it is the colormap that changes, not the real image data. The purpose of this is for speed. When a user drags the button in the scale bar in choosing a threshold value, the thresholded image simultaneously shows on the canvas. The “real” button, when pressed, modifies the image data according to the threshold value, and, when pressed again, undoes thresholding and restore the previous data. When the

“visual” button is off, both the “real” button and the scale bar are ineffective.

The output message part of the message section is implemented as a label widget. A member function of the message class can be called to display a string within the widget.

CHAPTER 7

SUMMARY AND FUTURE WORK

We studied four structural parameters of biofilm image in an open-ended approach. These four parameters each can capture a certain structural feature, and the results agree with human visual judgement. The parameters will be used in the biofilm research to study biofilm heterogeneities and biofilm structure/function relations. The software we developed for extracting the parameters has graphical user interface and several other image processing utilities. The research project is not ended; more parameters need to be studied to form a complete set of parameters that captures all essential structural information of an image such that the reconstructed image with the parameters is similar to the original one.

As we have seen in Chapter 5, the four parameters that we have do not form a complete set of parameters for defining all essential structural information of an image. Since our approach is an open-ended one, instead of close-ended Markov random field approach, more parameters can be studied and added into the description of biofilm structure. As a suggestion we recommend considering parameters such as the distribution of the cluster sizes and some shape parameter.

REFERENCES

- [1] Kellogg S. (1989) Three-dimensional ultrastructure of microbial biofilms, abstr. N18. Abstr. Annu. Meet. Am. Soc. Microbiol. 1989.
- [2] Bremer P., Geesey G, Drake B. (1992) Atomic force microscopy examination of the topography of a hydrated bacterial biofilm on a copper surface. *Current Microbiology*. **24**:223-230.
- [3] Brakenhoff G., van der Voort H., Baarslag M., Mans B., Oud J., Zwart R., and van Driel R. (1988) Visualization and analysis techniques for three-dimensional information acquired by confocal microscopy. *Scanning Microsc.* **2**:1831-1838.
- [4] Caldwell D. And Germida J. (1985) Evaluation of difference imagery for visualizing and quantitating microbial growth. *Can. J. Microbiol.* **31**:35-44.
- [5] Lawrence J., Korber D., and Caldwell D. (1989) Computer enhanced darkfield microscopy for the quantitative analysis of bacterial growth and behavior on surfaces. *J. Microbiol. Methods*. **10**:123-138.
- [6] Caldwell D., Korner D., and Lawrence J. (1992) Confocal laser microscopy and digital image analysis in microbial ecology . *Adv. Microbiol. Ecol.* **12**:1-67.
- [7] Massol-deya A., Whallon J., Hickey R., and Tiedje J. (1995) Channel structures in aerobic biofilms of fixed-film reactors treating contaminated groundwater. *Appl. Environ. Microbiol.* **61**:769-777.
- [8] deBeer D, Stoodley P., Roe F., and Lewandowski Z. (1994) Effects of biofilm structures on oxygen distribution and mass transfer. *Biotechnol. Bioeng.* **43**:1131-1138.
- [9] Zahid W., Ganczarczyk J. (1994) A technique for a characterization of RBC biofilm surface. *Water Res.* **28**:2229-2231.
- [10] Zhang T. and Bishop P. (1994) Density, porosity and pore structure of biofilms. *Water Res.* **28**:2267-2277.
- [11] Hermanowicz S., Schindler U., and Wilderer P. (1995) Fractal structure of biofilms. (Communication)

- [12] Cross G. and Jain A. (1983) Markov Random Field Models. IEEE Trans. Pattern Anal. Machine Intell., Vol.5, No.1, 25-39.
- [13] Besag J., (1974) Spatial interaction and the statistical analysis of lattice systems, J. Royal Statist. Soc., series B, Vol. 36, 192-326.
- [14] Elfadel I., et al (1994) Gibbs random fields, co-occurrences, and texture modeling. IEEE Trans. Patt. Anal. Mach. Intell., Vol.16, No.1, 24-37.
- [15] Per-Erik Danielsson (1980) Euclidean Distance Mapping. Computer Graphics and Image Processing. **14**:227-248.
- [16] Gray A, Kay J, and Titterington D. (1994) An empirical study of the simulation of various models used for images. IEEE Trans. Patt. Anal. Mach. Intell., Vol. 16, No 5., 507-513.
- [17] Press W et al, "Numerical Recipes in C", second edition, Cambridge University Press, 1992.
- [18] J. Russ, "Fractal Surfaces", Plenum Press, New York, 1994.

MONTANA STATE UNIVERSITY LIBRARIES



3 1762 10230965 3