LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

UCRL-TR-234578

Analysis of Images from Experiments Investigating Fragmentation of Materials

C. Kamath, O. Hurricane

September 13, 2007

This work was performed under the auspices of the U.S. Department of Energy by University of California, Lawrence Livermore National Laboratory under Contract W-7405-Eng-48.

Analysis of Images from Experiments Investigating Fragmentation of Materials

Chandrika Kamath and Omar A. Hurricane
Lawrence Livermore National Laboratory

August 30, 2007

Abstract

Image processing techniques have been used extensively to identify objects of interest in image data and extract representative characteristics for these objects. However, this can be a challenge due to the presence of noise in the images and the variation across images in a dataset. When the number of images to be analyzed is large, the algorithms used must also be relatively insensitive to the choice of parameters and lend themselves to partial or full automation. This not only avoids manual analysis which can be time consuming and error-prone, but also makes the analysis reproducible, thus enabling comparisons between images which have been processed in an identical manner. In this paper, we describe our approach to extracting features for objects of interest in experimental images. Focusing on the specific problem of fragmentation of materials, we show how we can extract statistics for the fragments and the gaps between them.

1 Introduction

The computational modeling of complex phenomena involves the validation of the simulation output by comparing it with experimental data. When the experimental output is in the form of images, a direct one-to-one comparison with the simulation "images" is often not possible for several reasons. First, the two images may be of different sizes, with information at different scales. The simulation output, which is usually at a higher resolution than the experimental data, can include both coarse- and fine-scale structures, while the lower-resolution experimental data captures only the coarse-scale structures. Second, the experimental image may contain parts of the experimental setup unrelated to the phenomena being observed such as portions of the diagnostics which are in the field of view of the object being imaged. Third, the experimental images are often noisy as a result of noise from the sensors, or other sources. This noise must be reduced prior to comparison.

In light of these issues, validation is usually done by identifying the structures of interest in the data, extracting features or characteristics which represent the structures, and then using the features to compare the simulation to experimental images. The features may need to be scaled appropriately to enable a one-to-one comparison. Depending on the problem, we can use either a direct comparison based on the size or the shape of the structures [6], or an indirect comparison using the distributions of the features. In addition to being used in validation, these features can also provide insights into the scientific phenomena being studied.

In this report, we describe the use of image processing techniques to analyze images from experiments investigating the fragmentation of materials. This report is organized as follows:

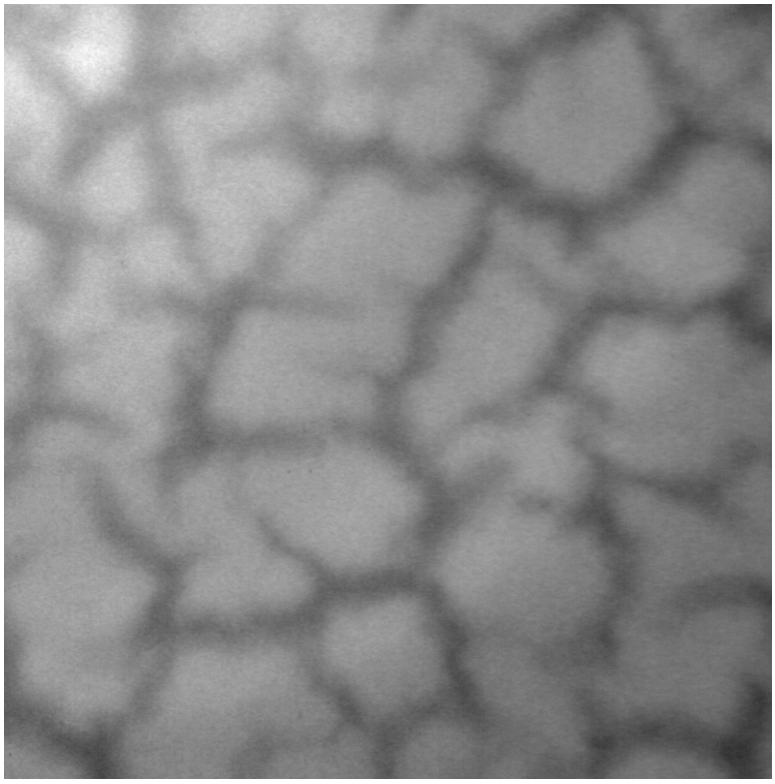
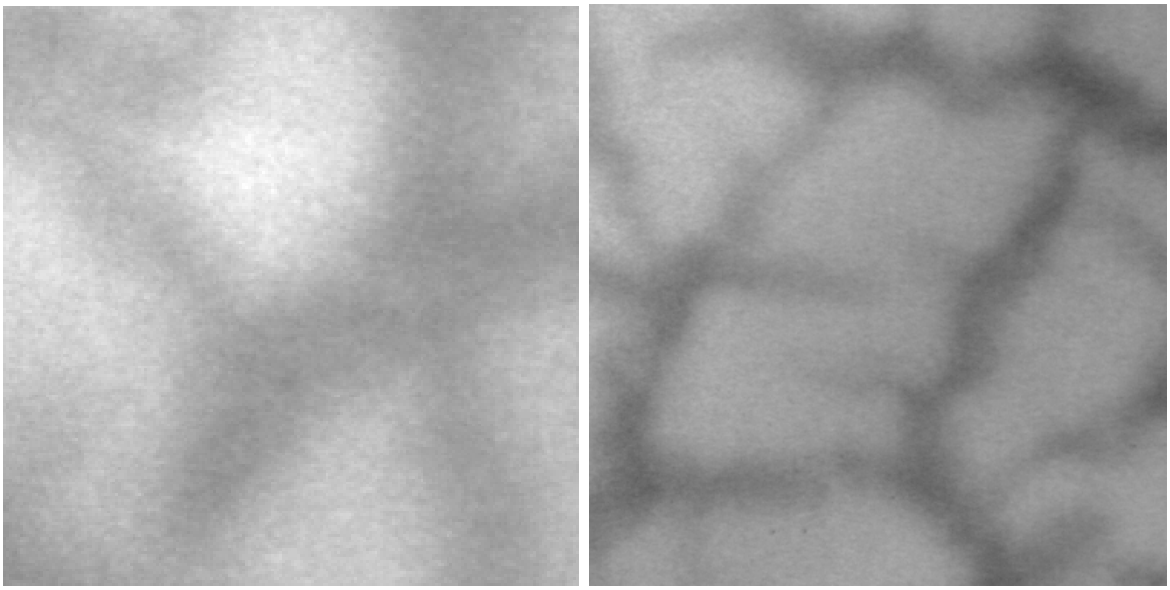


Figure 1: A sample image used in the analysis. This is a 800×800 subset of a larger image. Note the variation in intensity from the top left to the bottom right.

We first describe the problem and the data in Section 2, followed by our analysis approach in Section 3. This includes the enhancement of the image to reduce noise, the identification of the objects in the data, and the extraction of the relevant characteristics of interest for these objects. We present the results of these analyses in Section 4. In Section 5, we describe related work and discuss approaches used in problems similar to ours. We conclude with a summary in Section 6.

2 Problem Description

The images used in our study are from experiments investigating the fragmentation of materials. An image is taken of the material as it fragments, as shown in Figure 1. In this image, the lighter areas are the fragments of the material, while the darker regions represent the space between the fragments. The goal of the analysis is to obtain characteristics for both the fragments, such as their size, and the space between the fragments, such as the length and width of the gaps. The distributions of these characteristics for the whole image, in the form of histograms, can then be used to provide a concise summary of each image. As the analysis will be applied to images from several experiments conducted on different materials under different conditions, it is desirable that any techniques used for the analysis be as automated as possible, require few parameter settings, and be relatively insensitive to the settings of these parameters.



(a) (b)

Figure 2: Zoomed-in subsets of Figure 1 showing (a) the graininess of the images and (b) the variation in intensity and width along a gap.

A key challenge to the analysis is immediately apparent from Figure 1. There is a rather large variation in intensity from the upper left of the image to the lower right corner. In fact, the intensity in the darker gap regions at the upper left corner is close to the intensity of the lighter fragments at the lower right corner. Further, some fragments, such as the ones near the bottom of the image, have non-uniform intensity, with lighter pixels in one part and darker pixels in the other. This implies that segmentation of the images to separate the fragments from the background must be done with care to ensure that we extract the fragments correctly.

Another challenge is that the images are rather grainy, as shown in Figure 2(a), which is a zoomed-in region of the image in Figure 1. This sub-image also shows that there is no sharp demarcation between the darker gap regions and the fragments. Figure 2(b), which is another subset of the image in Figure 1, shows that some of the gaps extend part way into a fragment. In addition, both the intensity and the thickness along a gap can vary considerably. These issues further add to the challenges in segmenting these images.

3 Analysis Approach

To address our goal of characterizing the fragments in the image, and the gaps between them, we first need to segment the image into the foreground fragment objects and the "background" which forms the gaps. As there is no sharp gradient at the boundary of a fragment, gradient-based segmentation techniques, such as the Canny edge detector [16], are not an appropriate choice. A closer examination of the pixel intensities in the image indicated that a simple thresholding might work well, at least locally. However, given the varying illumination in the image, it was unlikely we could segment the image using a single global threshold. This indicated that we first had to enhance the image to make the illumination more uniform and reduce the noise, before we could segment the image and extract the necessary characteristics

for the foreground fragments and the background gaps.

3.1 Image enhancement

Non-uniform illumination in an image, such as the illumination in Figure 1, can be considered to be multiplicative noise. A common technique to address such problems is the Retinex algorithm, first proposed by Land [8]. This exploits the fact that the intensity at each pixel in the image can be represented as the product of the reflectance and the illuminance at that pixel. If we first take the natural logarithm of the image, thus converting the product into a sum, we can then "subtract out" the non-uniformity of the illumination.

There are several different Retinex methods which vary in how the illumination is approximated. For our work, we used the Multi-Scale Retinex (MSR) technique proposed by Jobson, Rahman, and Woodell [11, 5, 4, 12]. Let $I(x, y)$ represent the two-dimensional image. Then, the single-scale retinex output, $R(x, y)$ is defined as

$$R(x, y) = log_e(I(x, y)) - log_e(F(x, y) * I(x, y)), \tag{1}$$

where the second term represents the illumination and is the convolution of the image with a Gaussian filter F of the form

$$F(x, y) = K \exp[-(x^2 + y^2)/\sigma^2]. \tag{2}$$

The standard deviation σ of the Gaussian determines the scale and the constant K is selected such that

$$\iint F(x, y)\, dx\, dy = 1, \tag{3}$$

where the integral is taken over the filter. The multi-scale formulation of the Retinex method is essentially a weighted sum of N applications of the single-scale retinex algorithm

$$MSR(x, y) = \sum_{i=1}^{N} w_i R_i(x, y), \tag{4}$$

where each $R_i(x, y)$ is obtained using a different scale σ_i and the weights sum to 1.

For our images, we found that it was sufficient to use two equally weighted scales, one small with $\sigma = 20$, and the other larger, with $\sigma = 80$. We implemented the convolution using fast Fourier transforms to reduce the computational time for such large kernels.

Figure 3 shows the sub-image in Figure 1 after the application of the multi-scale retinex algorithm. Note that the illumination is much more uniform across the image.

This image was further smoothed to reduce the grainy noise by convolving twice with a 5×5 Gaussian filter followed by a single application of the minimum mean-squared error (MMSE) filter [18]. The latter is an adaptive filter which exhibits varying behavior based on local statistics of the image. It is defined as

$$I'(x, y) = I(x, y) - \frac{\sigma_n^2}{\sigma_l^2}[I(x, y) - m_l(x, y)] \tag{5}$$

where $I(x, y)$ is the pixel value at location (x, y) in the original image I, $I'(x, y)$ is the corresponding pixel in the denoised image, σ_n is the estimate of the noise variance for the image, σ_l is the variance in a window centered at pixel (x, y), and $m_l(x, y)$ is the mean intensity in the same window. Typically, a 5×5 window is considered.

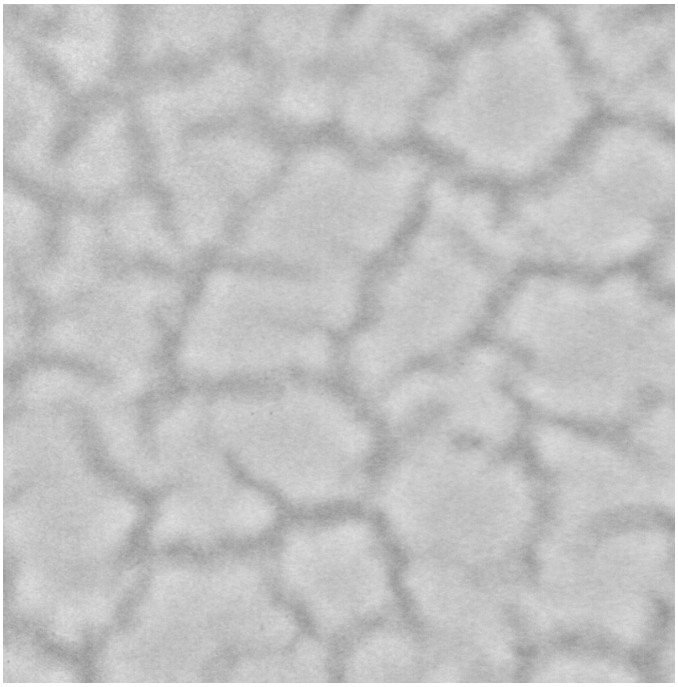


Figure 3: The sample subimage from Figure 1 after the application of the multi-scale retinex algorithm to make the illumination more uniform.

The MMSE filter leaves almost unchanged those areas of the image where the local variance is much higher than the noise variance, such as near the edges of the objects in the image. In regions which are fairly constant, that is, where the local variance σ_l is close to the noise variance σ_n, the filter returns a value close to the mean value in the window.

We estimate the image noise variance by first subtracting the mean-filtered image from the original image and then taking the variance of the resulting image after a $3\,\sigma$ clipping [17]. By subtracting the mean-filtered image from the original, we obtain a measure of the noise at each pixel in the image. By clipping values outside the $3\,\sigma$ limit, where the σ is calculated for the resulting noise image, we remove the contributions of edges or high gradient regions in the image. The variance of the noise image after clipping gives a robust estimation to the noise variance in the original, noisy image. This technique for estimating the variance of the image noise, though simple, has been found to work well for images with Gaussian additive noise [9].

The denoised image, after the application of the Gaussian and MMSE filters, is shown in Figure 4. Instead of simple techniques to reduce the grainy noise in the image, we could have used more complex techniques such as wavelet denoising or diffusion-based denoising. However, these approaches are more time consuming, require the setting of several parameters, and do not always work better than the simple techniques [19, 20, 2].

3.2 Segmentation and Cleanup

Once the image has been processed to make the illumination more uniform and reduce the graininess, we can segment the foreground fragments from the background by using a simple thresholding method. There are several ways in which we can identify an appropriate threshold. First, we can estimate a threshold by looking at the pixel intensities in the lighter and darker

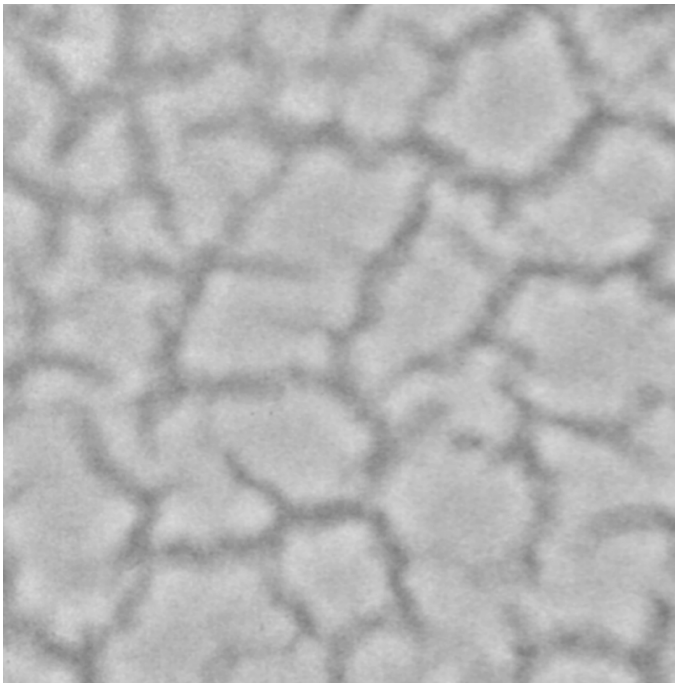
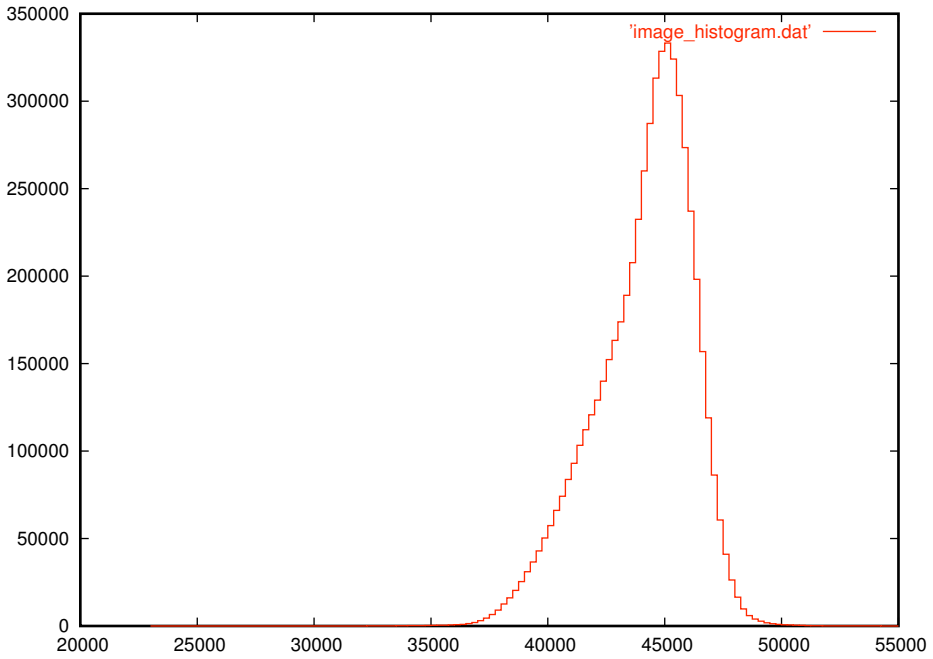


Figure 4: The sample subimage from Figure 3 after the application of denoising to remove the graininess from the image.

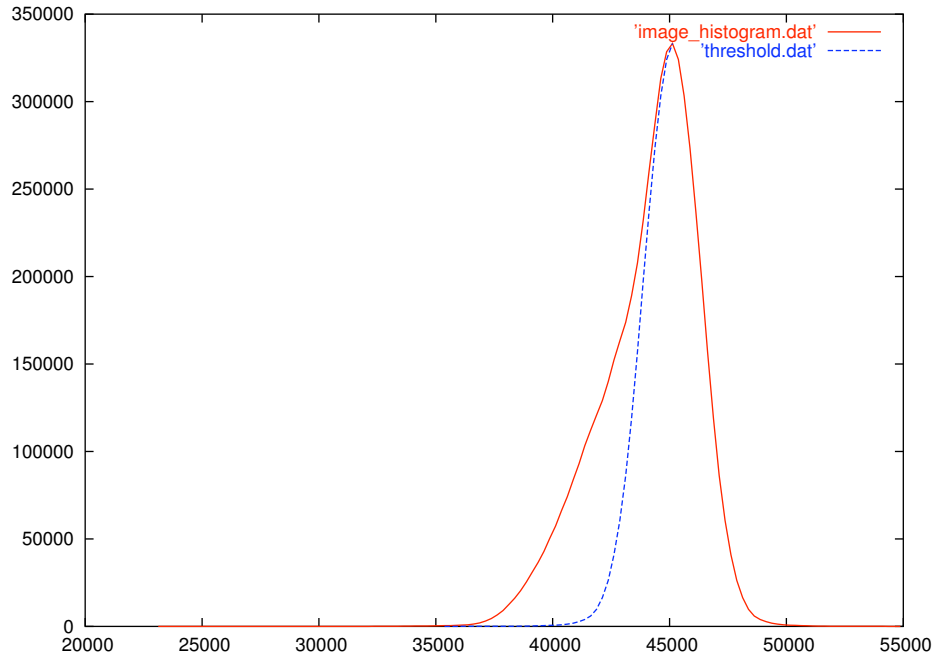
regions of the image as well as the intensity of pixels near the boundary of the fragments. This method suggested a threshold value of approximately 43500 for our sample image. An alternative approach exploits the histogram of the image. As seen in Figure 5, panel (a), the shape of the histogram is asymmetric. It has a sharp peak, with a Gaussian-like distribution to the right of the peak. These are the high intensity values from the fragments in the image. If we flip this curve over to the left about the peak (as shown in Figure 5, panel (b)), the two halves of the curve diverge around a value of 44000, with the left half of the curve having values higher than the right half. The intensity value where the two halves of the curve deviate can be chosen as the threshold.

We have observed that the statistics for the various characteristics extracted from the image are not very sensitive to the choice of threshold. Further, the threshold does not have to be very precise; we have typically chosen thresholds which are multiples of 500. Changing the threshold within a range of 2000 does not make an appreciable difference to statistics such as the distribution of the width of the gaps in the image. It may however change statistics such as the percentage of the image which is covered by the gaps, which is to be expected. Larger variations in the thresholds may change the locations of the distributions of gap widths and lengths, but does not change the shape of the distributions appreciably.

Figure 6, panel (a) shows the results of thresholding the image in Figure 4 using a threshold value of 43500. Values in black indicate pixel values below the threshold. We observe that there are several small black regions in the white area and vice-versa. This is to be expected, given the graininess of the original image. We also observe that the boundaries of the fragments are quite jagged in some places, with thin filament like structures, sometimes just a single pixel wide, as shown in the zoomed-in region in panel (b). These are again an artifact of

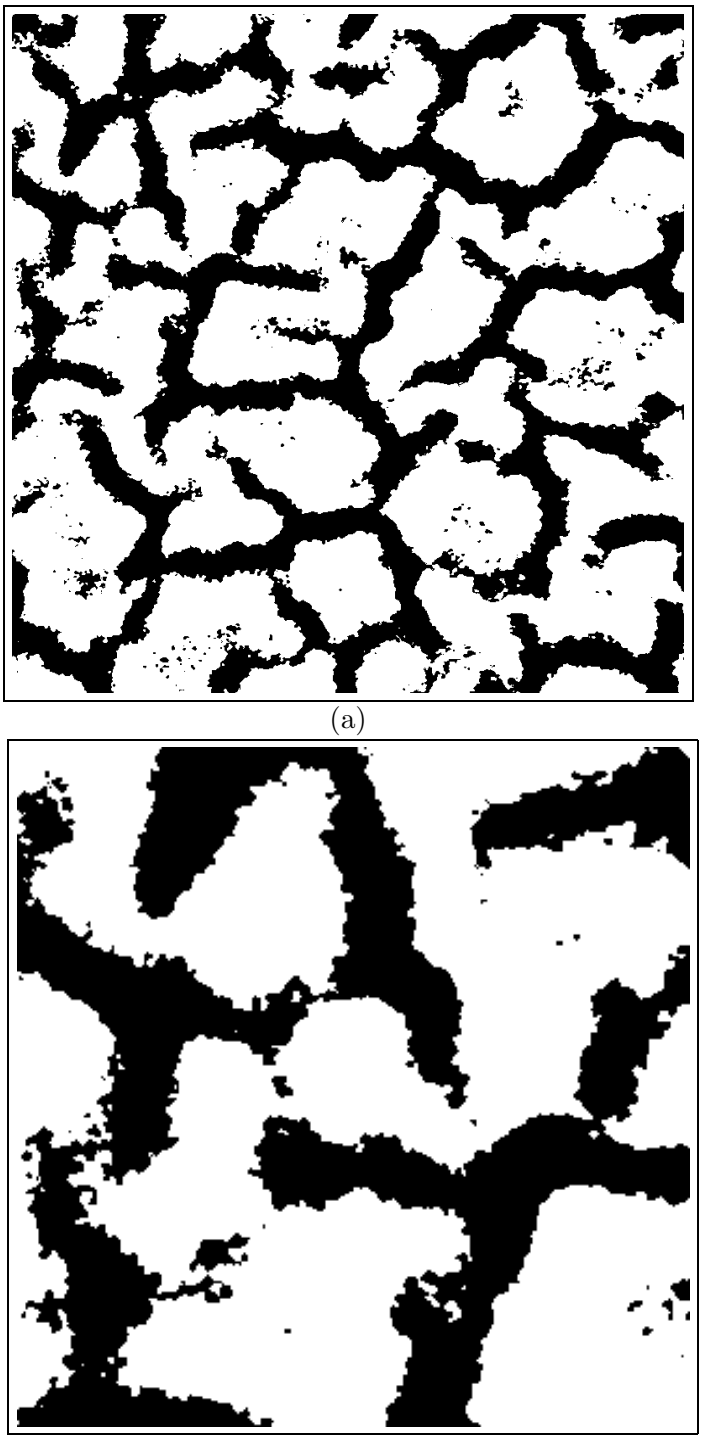


(a)



(b)

Figure 5: (a) The histogram of an image after denoising. Note the asymmetry of the intensity distribution. (b) The right half of the curve in (a), flipped over to the left is shown in blue. The value where the red and blue curves diverge is around 44000.



(a)

(b)

Figure 6: (a) The image from Figure 4 after thresholding using a threshold value of 43500. The region in black indicates pixel values below the threshold. (b) Zoomed-in region of (a) showing details of the segmented image.

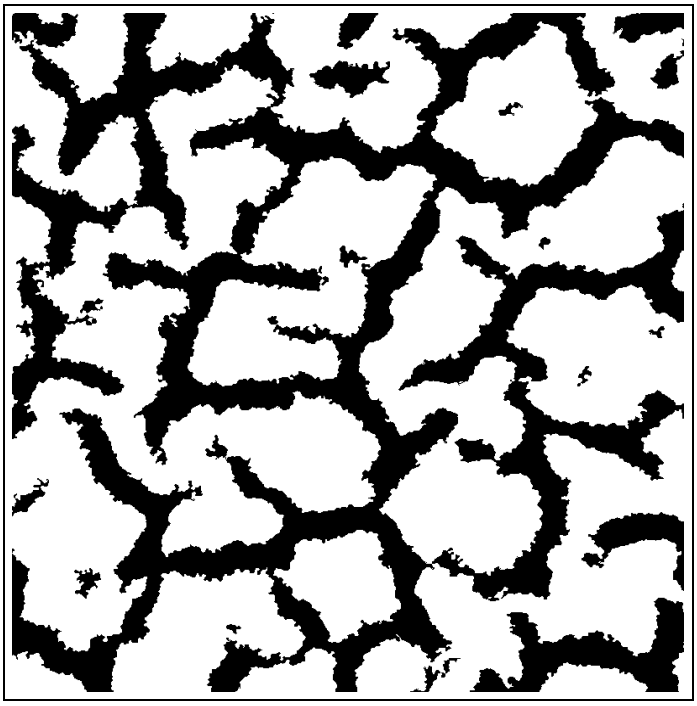


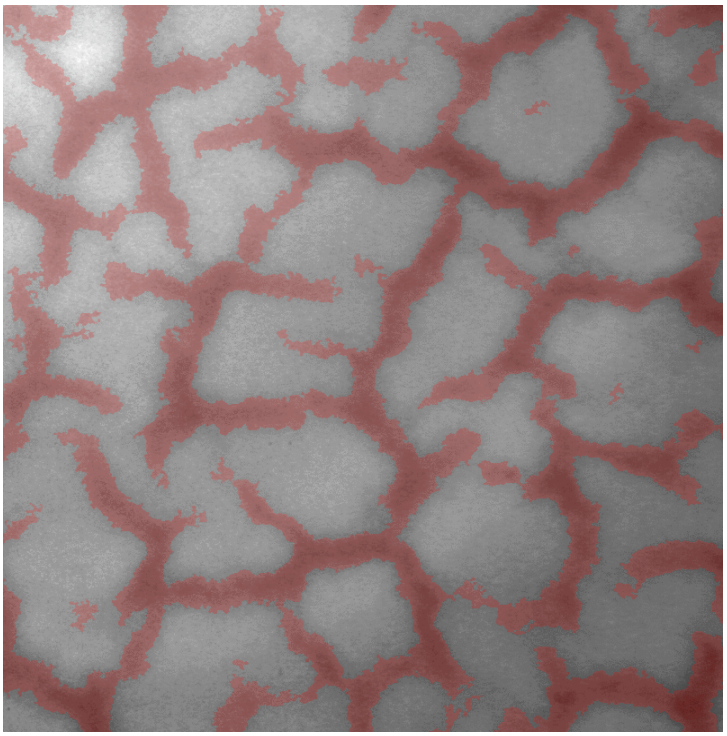
Figure 7: The image from Figure 6, panel (a) after removal of small isolated black and white regions smaller than 100 pixels in size.

the thresholding. These issues indicate that the thresholded image must be processed further before we can extract statistics on the white and black regions, corresponding to the fragments and the gaps, respectively.

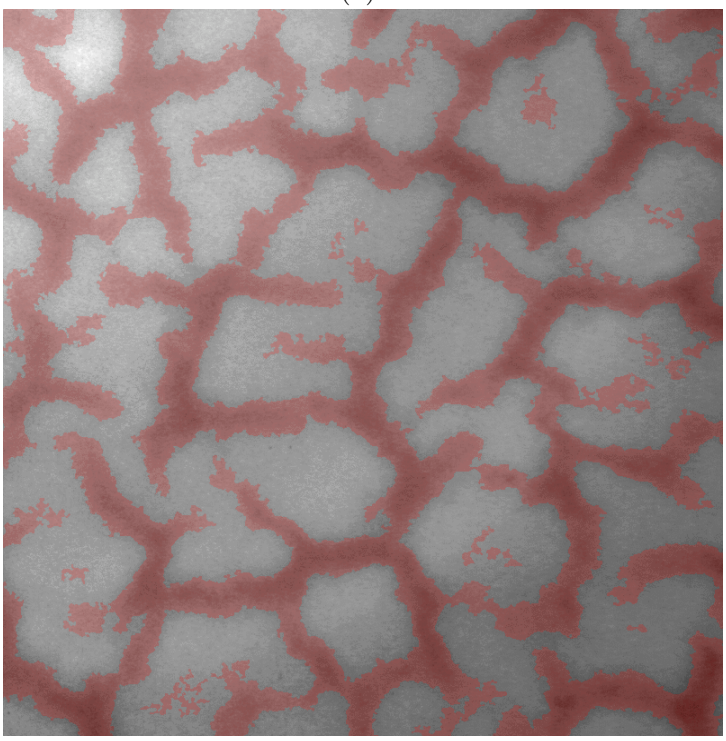
There are several ways in which the thresholded image can be cleaned. We next describe one approach in which we process the image using a sequence of cleanup operations. The order in which these operations are applied can vary, and some of them can be skipped altogether or applied more than once.

We first identify the isolated black and white regions by considering a cluster of pixels of one color which is completely surrounded by pixels of another color. Next, isolated regions less than 100 pixels in size are removed. Figure 7 shows the results after this cleanup step has been applied to the image in Figure 6, panel (a). Figure 8, panel (a) shows the image from Figure 7 overlaid on the original, with the regions in red indicating the gaps between the fragment areas of the image. Panel (b) is a similar image obtained using a threshold of 44000, and shows the effect of a different threshold value in the segmentation. As expected, with a higher value of the threshold, the gap regions are wider, and there are more small isolated regions as many of them are larger than the 100-pixel limit required for removal. In addition, some unconnected regions, which are near to each other at the lower threshold, become connected at the higher threshold.

Next, the jagged boundary of the fragments is smoothed by applying a mean filter to the binary image in Figure 7 and thresholding. We first apply a mean filter of size 11×11 three times to the image, and then set all values less than 128 to 0. This results in the image in Figure 9. Note that other ways of smoothing the fragment boundaries are also possible and smoothing may be applied more than once to obtain the desired result.



(a)



(b)

Figure 8: (a) The image from Figure 7 overlaid in red on the original image. (b) A similar image obtained after processing using a threshold of 44000 in the segmentation.

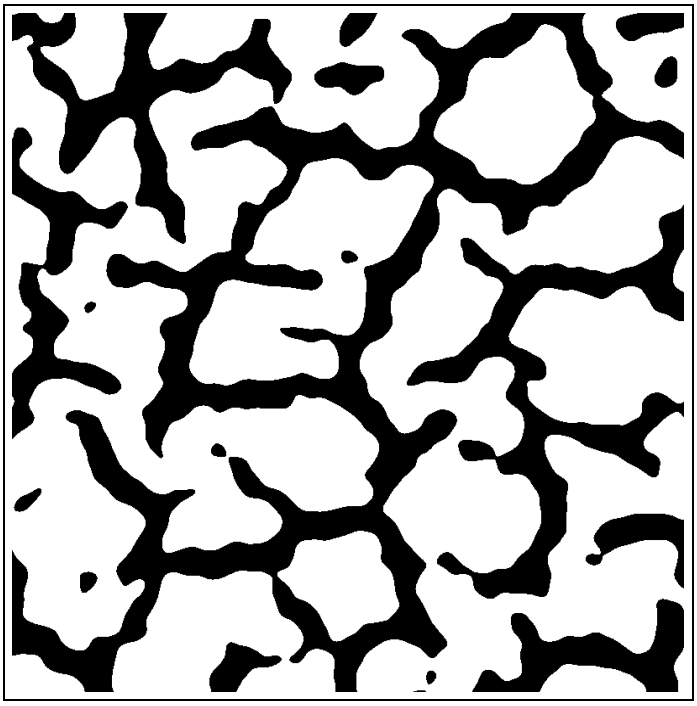


Figure 9: (a) The image from Figure 7 after smoothing to reduce the jagged nature of the fragment boundaries.

This smoothing results in the black regions of the binary image becoming wider. We next shrink these by applying the morphological operation of erosion [15] to the black regions, or alternatively, the operation of dilation to the white regions. We use a 3×3-cross structuring element of the form

$$\begin{array}{ccc} & X & \\ X & X & X \\ & X & \end{array}$$

and dilate the white regions by tagging as white any pixel which is covered by the structuring element when the center of the element is placed, in turn, on each of the white pixels. This results in black pixels adjacent to white pixels being converted to white pixels, thus shrinking the black regions. The resulting image is shown in Figure 10. Note that if we had selected a segmentation threshold which was smaller than optimal, resulting in narrower gap regions, we could have skipped the dilation step as the widening of the gap regions during the smoothing step would have compensated for the choice of threshold.

Finally, we notice that there are some small black regions which had not been removed earlier as their size prior to processing was greater than 100 pixels. These isolated black regions of size 1000, or smaller, can be removed as described earlier. The resulting image is shown in Figure 11. Figure 12, panel (a) shows the image from Figure 11 overlaid on the original, with the regions in red indicating the gaps between the fragment areas of the image. The corresponding image obtained using a segmentation threshold of 44000 is shown in panel (b).

This cleaned image can now be used for extracting the characteristics of interest in the data.

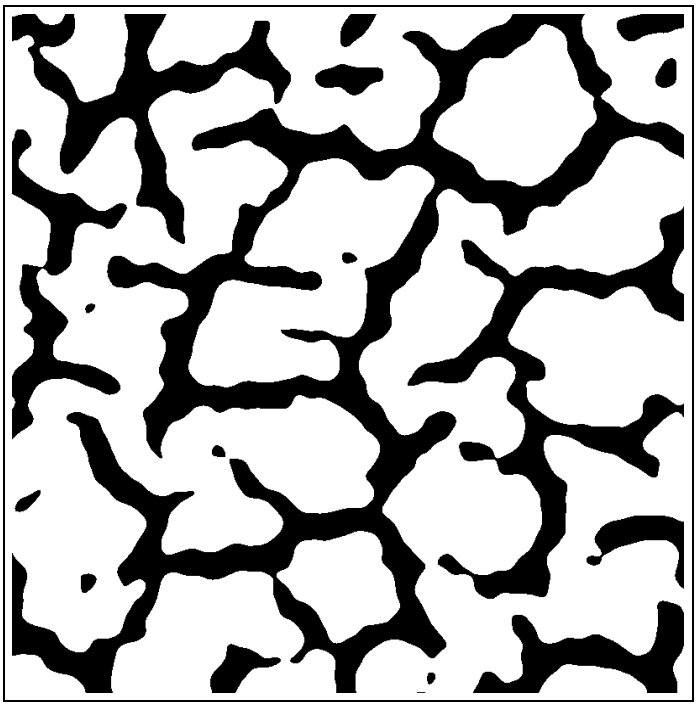


Figure 10: The image from Figure 9 after dilation of the white regions to shrink the black regions which had become wider in the smoothing step.

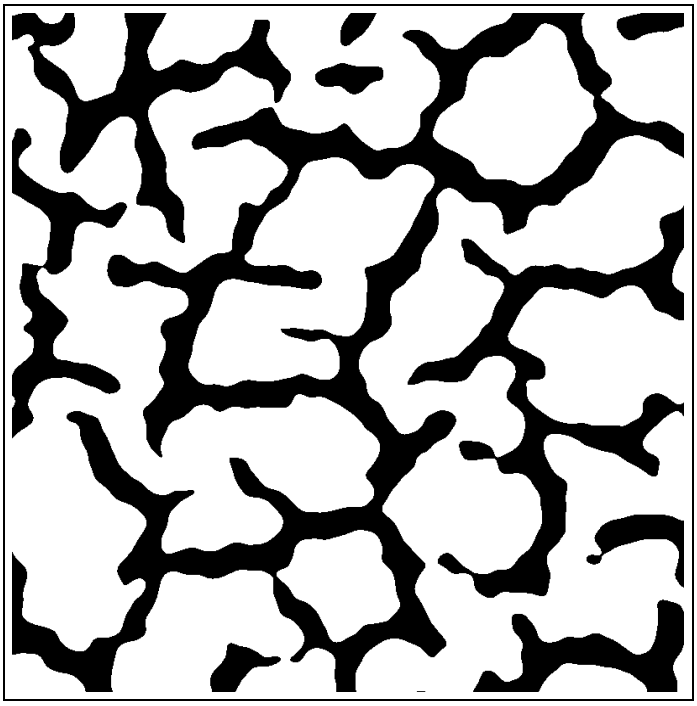
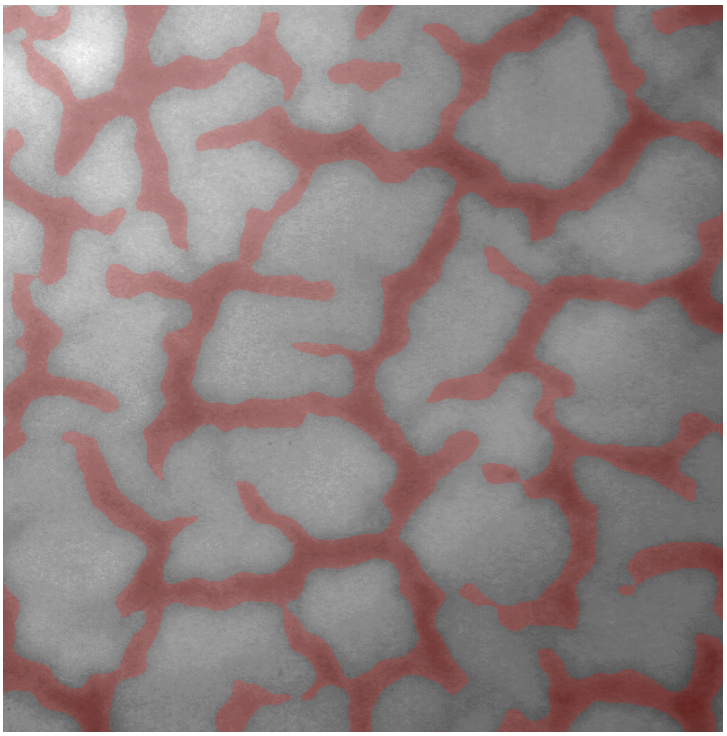
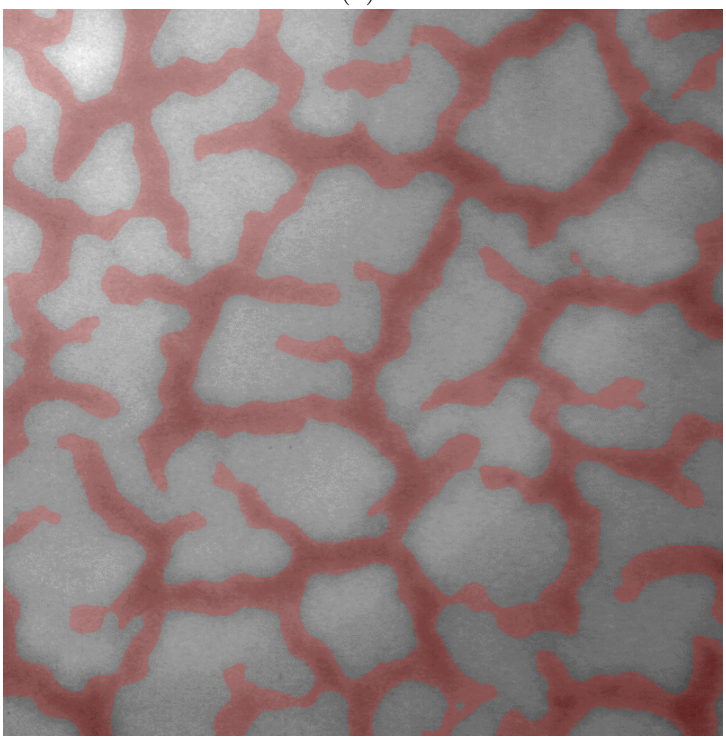


Figure 11: The image from Figure 10 after removal of isolated black regions smaller than 1000 pixels in size.



(a)



(b)

Figure 12: (a) The image from Figure 11 overlaid in red on the original image. (b) A similar image obtained after processing using a threshold of 44000 in the segmentation.

3.3 Extracting Characteristics of Interest

The image in Figure 11 obtained after de-noising, segmentation, and cleanup of the original image can now be used to extract features or characteristics of the objects in the data, namely the fragments, in white, and the gaps, in black. Certain statistics can be easily extracted from the image, such as the percentage of black pixels, the sizes of the fragments, and the distribution of the sizes of the fragments. Other characteristics, such as the length and width of the gaps, are more difficult to extract. We next describe one way in which this information can be obtained.

The gaps between the fragments can be thought of as edges of a graph, where the edge has a varying width along its length. At many of the corners of the fragments in the image, one or more edges meet in what can be considered as a node of the graph. One way of extracting the connectivity of the black regions of the image is to obtain the medial axis or the skeleton of the black regions of the image.

We obtain the skeleton using Hilditch's algorithm [3], a very simple algorithm which thins the black regions iteratively. For this purpose, we consider each pixel in the black region to have value 1 and each pixel in the white region to have value 0. Consider the eight neighborhood of a pixel P1, whose neighbors are labeled as

P9	P2	P3
P8	P1	P4
P7	P6	P5

For a pixel $P1$, we then define two functions,

$$B(P1) = \text{Number of nonzero neighbors of P1} \tag{6}$$

and

$$A(P1) = \text{Number of } 0,1 \text{ patterns in the sequence P2, P3, P4, P5, P6, P7, P8, P9, P2} \tag{7}$$

In each iteration of the Hilditch's algorithm, we make a complete pass over all black pixels in the image. On each pass, we decide whether to convert a black pixel to white, which thins the black regions. If a black pixel $P1$ satisfies the following four conditions

$$2 \leq B(P1) \leq 6 \tag{8}$$

$$A(P1) = 1 \tag{9}$$

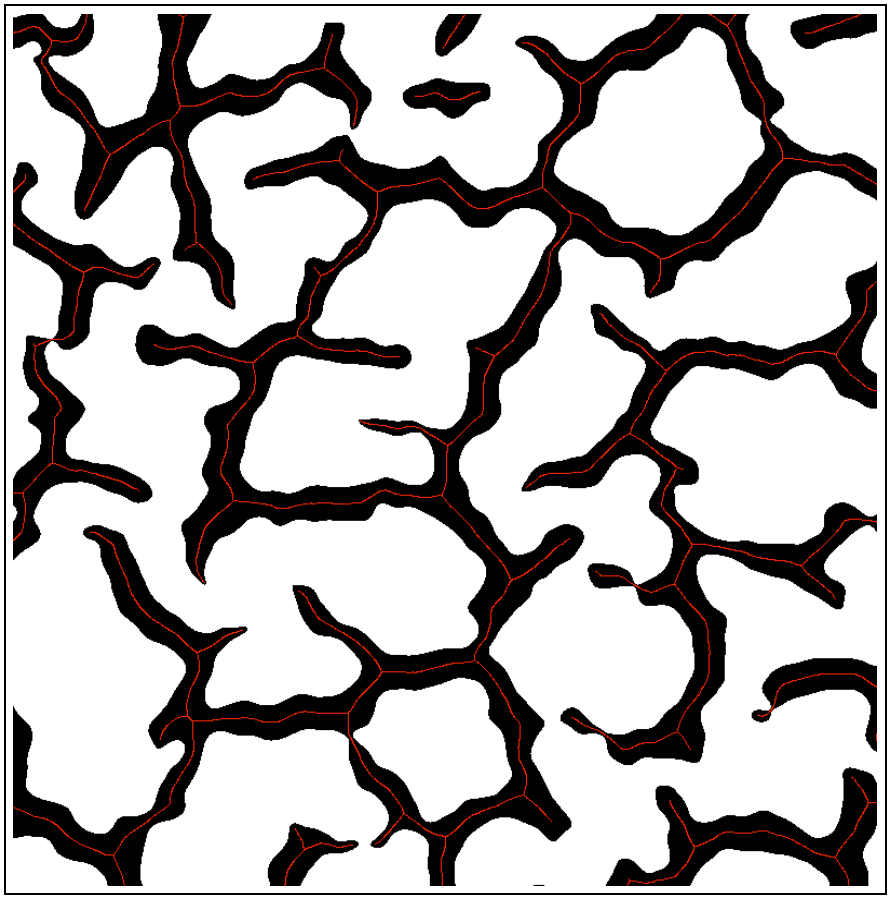
$$P2 * P4 * P8 = 0 \quad \text{or} \quad A(P2) \neq 1 \tag{10}$$

$$P2 * P4 * P6 = 0 \quad \text{or} \quad A(P4) \neq 1 \tag{11}$$

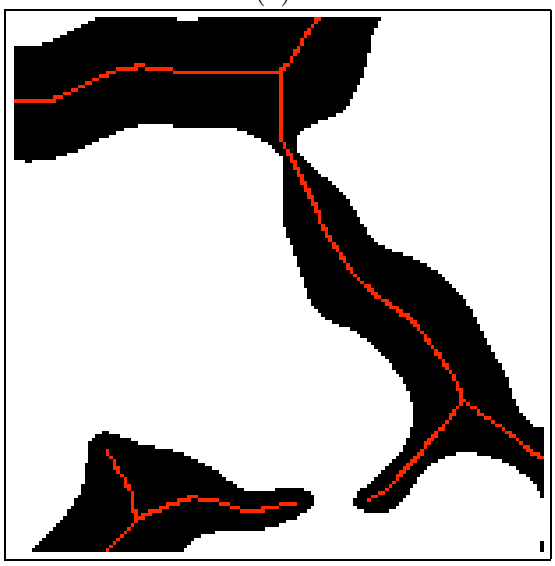
where $*$ indicates a product, then a pixel is changed from black to white. Multiple passes are made through the image, each time thinning the black regions progressively, until no pixel changes color. Note that we could have also used morphological operations or the medial axis transform to obtain the skeleton of the black regions.

Figure 13, panel (a) shows the skeleton pixels (in red) superposed on the binary image from Figure 11. This skeleton can be considered as a graph, with the nodes identified as pixels where more than two edges meet. One can then conceivably "walk" along the edges of the graph and obtain the length of each edge as the distance in pixels from one node to the next.

However, a closer look at a zoomed-in region in panel (b) indicates that some cleanup might be necessary before we extract the length and width of the gaps between the fragments



(a)



(b)

Figure 13: (a) The skeleton for Figure 11 superposed in red on the binary image. (b) A zoomed-in portion of the skeleton.

in the image using the edges of the skeleton. We observe that there are some small "dendrites" caused by little bumps in the boundary of the black regions. This is a known problem with skeletonization algorithms which tend to be rather sensitive to small changes in the boundary. The smoothing done after the segmentation helps to reduce these structures, but some still remain in the process of skeletonization and must be removed prior to the extraction of the characteristics of interest. We also notice that if we walk along the edge of a skeleton, we often encounter more than one path to the next neighboring pixel, as in the following structures

```
X          X
X          X
X             X
X  X          X  X
   X             X  X
   X                X  X
   X                   X  X  X
```

These are the result of the skeletonization algorithm and to ensure that we calculate the correct length of the edges, we should have a single path from one pixel to the next in the skeleton.

To accomplish these two steps in the cleanup of the skeleton, we first label each pixel in the skeleton as either an edge pixel, a node pixel (that is, a junction point), or a tip pixel (that is, an extremal point). This process of labeling the skeleton converts it into a graph with nodes and edges between the nodes clearly identified. The labeling is done by using simple heuristics derived from the eight-neighborhood of typical edge, node, and tip pixels. For example, a pixel with only one neighbor is a tip pixel and a pixel with only two neighbors is an edge pixel. When the number of neighboring pixels is three or more, then we also include the number of 0-1 transitions, that is, the value of $A(P1)$ and use it to differentiate between edge and node pixels. Additional heuristics are included as appropriate. An alternate approach to distinguishing node pixels from edge pixels is to consider a circle around each pixel and count the number of intersections of the circle with the skeleton. However, the results are very dependent on the radius of the circle, and we can get incorrect results in the presence of short edges.

Once the skeleton pixels have been labeled, the graph can be thinned further to ensure that there is only one path from one pixel to the next. This is done by walking along each edge of the graph starting with a node or a tip pixel, and checking the number of "unseen" pixels which are the 8-neighbors of a current pixel. If there are two such pixels, one of which is a 4-neighbor and the other is an adjacent diagonal neighbor, then the 4-neighbor can be removed from the edge. This allows us to move from the current pixel to its diagonal neighbor using a single diagonal step rather than a horizontal and a vertical step.

The labeling of the skeleton pixels, especially the differentiation between edge and node pixels, is non-trivial. A large part of the problem is the result of the skeleton being more than a single pixel wide in places, especially near the nodes and along edges at an angle. This is a known drawback of the Hilditch algorithm, which though very simple, yields less than perfect results for our purpose. An alternative is to use more complex techniques for thinning, such as the Ahmed and Ward approach [1], which uses 20 rules applied simultaneously to each pixel, along with an extension proposed by Rockett [13] which handles two-pixel wide lines. This would have resulted in a cleaner skeleton, making the labeling easier. However, the complexity of the task would have shifted to the thinning operation rather than the labeling.

Once the pixels in the skeleton have been labeled, we can remove the short edges resulting

from bumps in the boundary of the black regions. This is done by considering all edges which end in a tip pixel and removing those whose Euclidean distance from the node which connects the tip edge to the rest of the graph is less than a certain threshold. Isolated edges which begin and end in a tip pixel and are less than the threshold in length are also removed. In our work, we set the threshold to 30.

Note that the use of the Hilditch algorithm for thinning often results in several extra skeleton pixels near a node or a junction point. When a short tip edge is removed, the node pixel to which the tip was attached now becomes an ordinary edge pixel in the graph. The thinning algorithm must be applied again to remove the extraneous skeleton pixels near this node, followed by relabeling of the graph to convert the node pixel to an edge pixel.

In addition, we also found that in a gap region which is a dead end, there are often several tip edges whose length is below the threshold for short edges, as in Figure 14. Rather than remove all these short edges, we retain the longest of them.

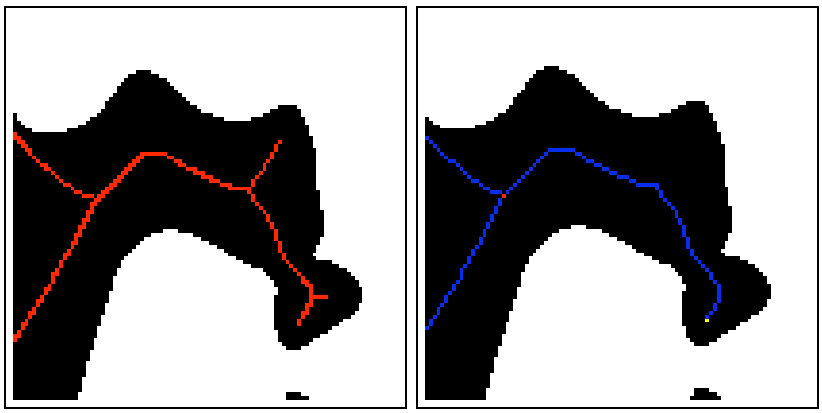


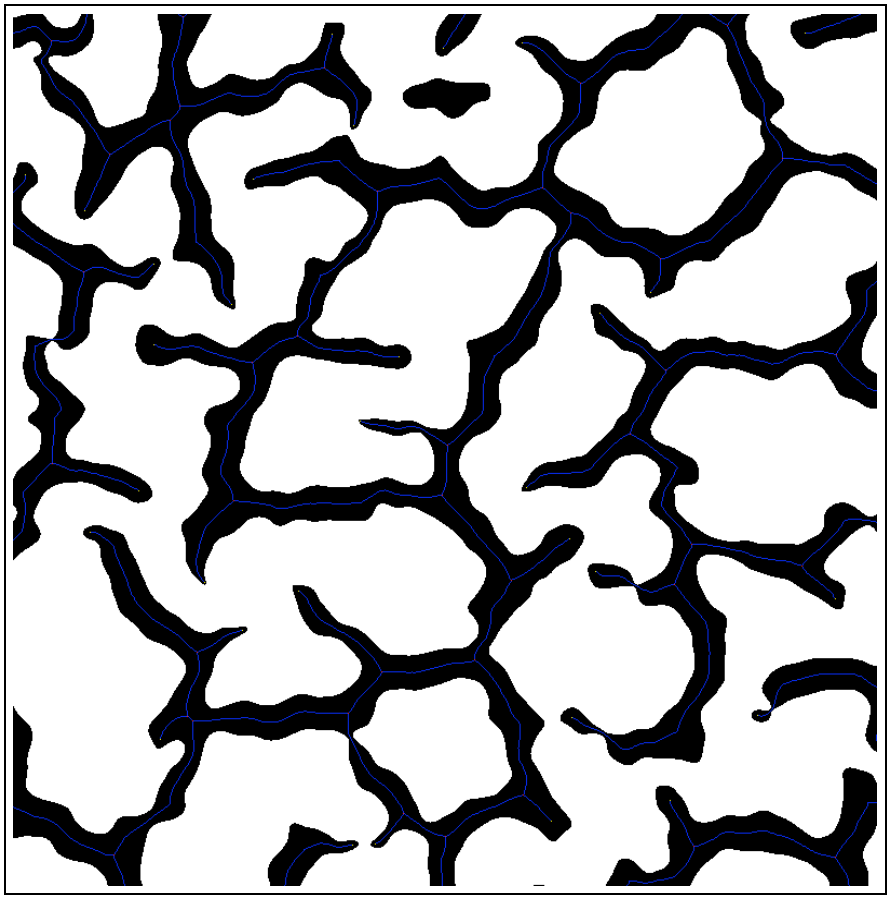
Figure 14: In the case of two tip edges which are below the short-edge threshold and lie at the end of a dead-end gap region (left panel), we retain the longest of these tip edges in the cleanup process (right panel).

Figure 15 shows the graph after cleanup to thin the edges and remove short edges. Figure 16 shows the graph superposed on the original image for the segmentation thresholds of 43500 and 44000. Note that the two images are very similar, indicating that the final graph obtained is not sensitive to the choice of the threshold in the segmentation.

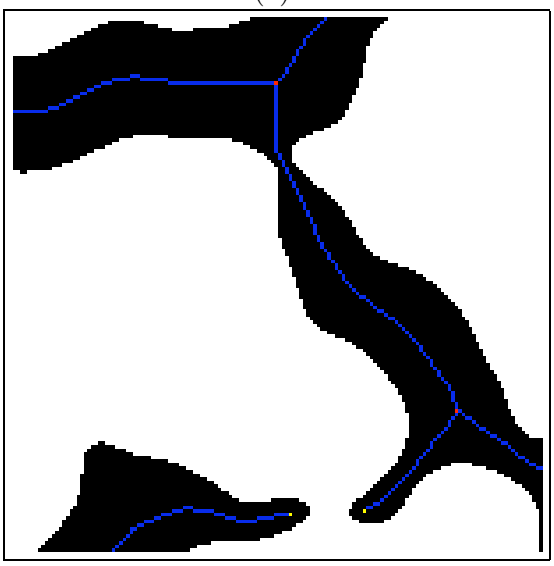
The desired characteristics of interest, namely the length and the width of the gaps between the fragments in the image can now be obtained easily using the graph and the binary image obtained after segmentation and cleanup. The length of the gaps is just the length along the edge of the graph. This can be obtained by counting the number of pixels from node to node or from node to a tip in the case of an edge which ends in a tip. For greater accuracy, diagonal pixels can be multiplied by $\sqrt{2}$ to account for the greater distance along the diagonal.

The width of the gap at any pixel on the graph is calculated by considering the diameter of the largest circle which is centered at the given pixel and just fits inside the black region. A circle which is a pixel larger in radius would not be completely contained in the black region. Alternately, if the skeleton had been obtained using a medial axis transform, we could have extracted the distance of each graph pixel to the boundary of the black region by using the radius function.

Once we have obtained the width of the gap at any pixel of the graph, we can obtain the

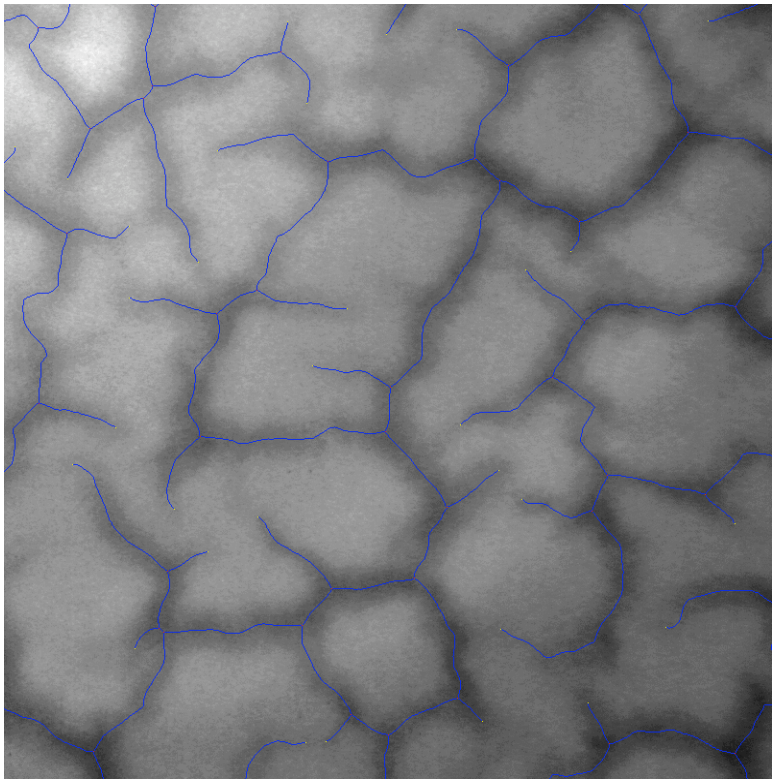


(a)

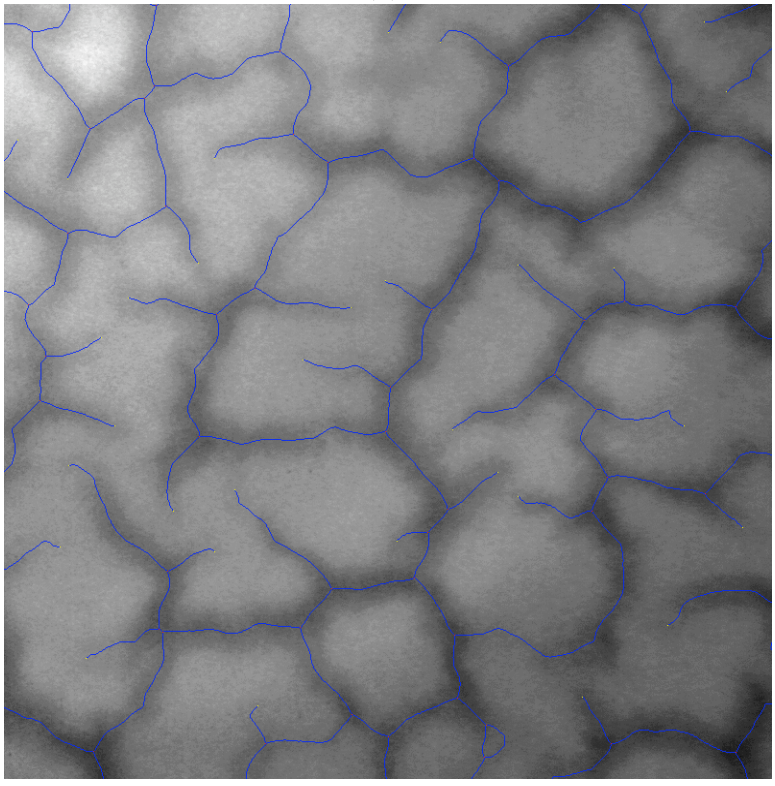


(b)

Figure 15: (a) The graph corresponding to the skeleton in Figure 13, after cleanup, superposed on the binary image. Blue pixels are edge pixels, red are node pixels, and yellow are tip pixels. (b) A zoomed-in portion of the graph.



(a)



(b)

Figure 16: (a) The graph from Figure 15 overlaid on the original image. (b) A similar image obtained after processing using a threshold of 44000 in the segmentation.

average width of the gap by taking the average of the widths for all pixels along the edge from node to node or node to tip.

4 Analysis Results

We next present the results of our analysis for the image considered in this study. These results are for the full image, not the subset that was used in the previous sections to illustrate the steps in the analysis.

To understand the effect of the threshold selection in the segmentation, we include results for the two threshold values of 43500 and 44000. The percentage of the image area covered by the fragments is 70.0% and 63.1%, respectively, for the two thresholds. This is to be expected as the higher threshold selects more pixels as belonging to the gap regions.

Figure 17 shows the distributions of the size of the fragments in number of pixels using histograms of 100 bins. At the higher threshold, some of the fragments split into smaller pieces when a gap, which extended part way through a fragment at the lower threshold, gets completed, thus splitting the fragment into two smaller pieces. To focus on the distribution of the smaller fragments, Figure 18 shows the same histogram with all fragments larger than 20000 pixels in size counted in the last bin and all fragments smaller than 200 pixels counted in the first bin.

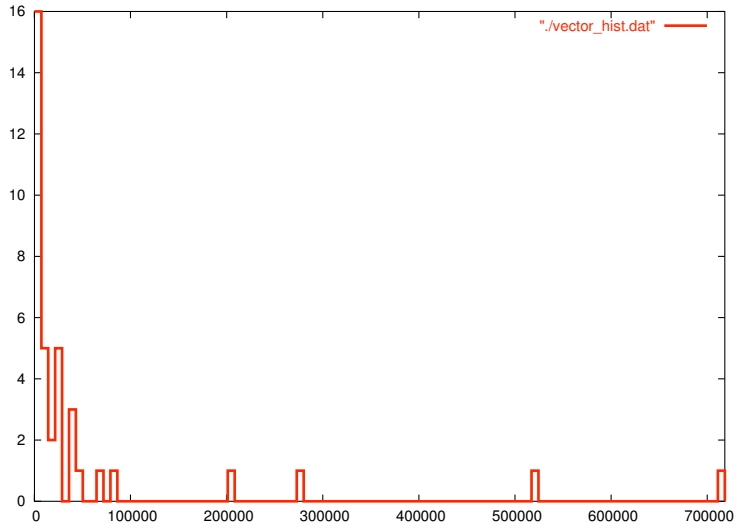
Figure 19 shows the distribution of the length of the edges of the graphs for the two threshold values. The edge length is obtained by considering the distance along the edge pixels from one node to another node or to a tip pixel of the graph. As mentioned earlier, a 4-neighbor pixel contributes 1 to the edge length, while a diagonal pixel contributes $\sqrt{2}$. Due to the cleanup step in the analysis, which ensures that there is only one path from one pixel in the edge to the next, no extraneous pixels are counted in the length.

Figure 20 shows the distribution of the average width of the edges of the graphs for the two threshold values. For each edge, the width is calculated at each pixel along the edge and the average width for that edge is obtained. To avoid unduly biasing the width by including wider gap regions around the node pixels, we do not consider the width at an edge pixel if the maximal circle at the pixel includes a node pixel. This effectively removes edge pixels very close to a node from being considered in the average. Figure 21 shows the width distribution for all edge pixels considered above, ignoring the averaging along the edge.

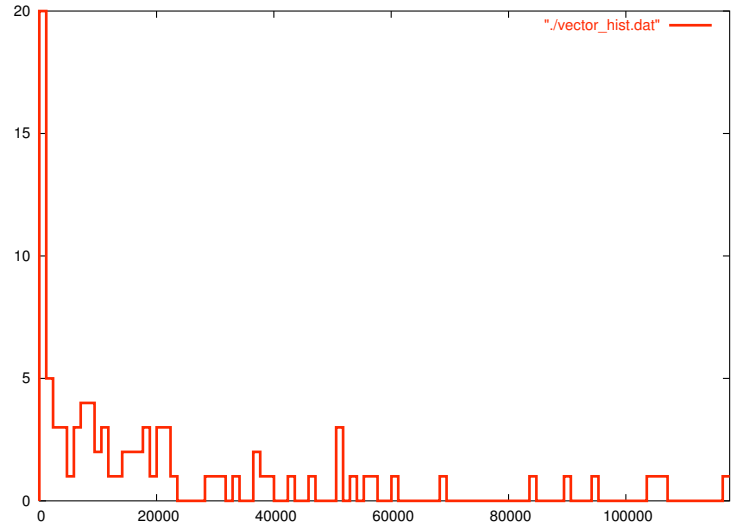
5 Related Work

The techniques used to analyze the fragments and the gaps between them have been used in other problem, albeit in different application areas. For example, Koh at al. [7] have used similar techniques to analyze high-resolution, optical micrograph images of neuron structure. In these images, the neuron appears as a long thin structure (referred to as the spine or the backbone), with smaller structures (the dendrites) branching from it. The images are first enhanced using deconvolution and median filtering to reduce the noise, followed by threshold-based segmentation to separate the neuron from the background. The spine and the dendrites are then obtained using a medial-axis algorithm. The cleanup step to reduce spurious cell debris, or handle disconnected features, is done after the extraction of the medial axis. In our work, we perform this cleanup prior to the extraction of the medial axis.

Similar techniques have been used to study void space geometry in rocks [14, 10] from X-ray

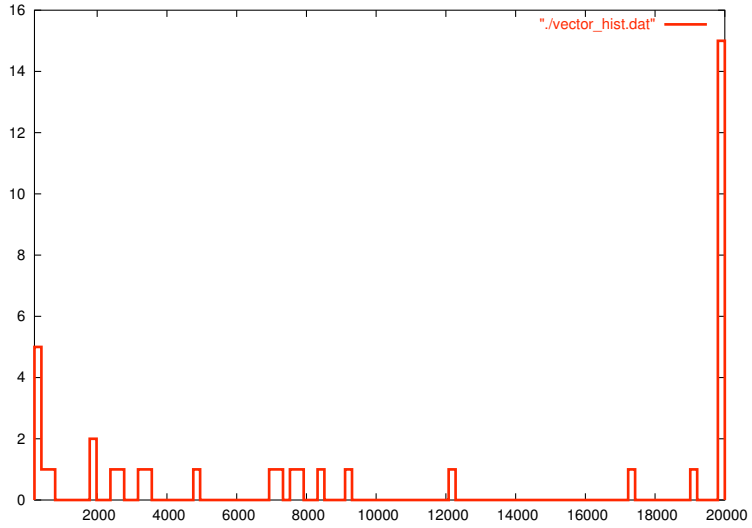


(a)

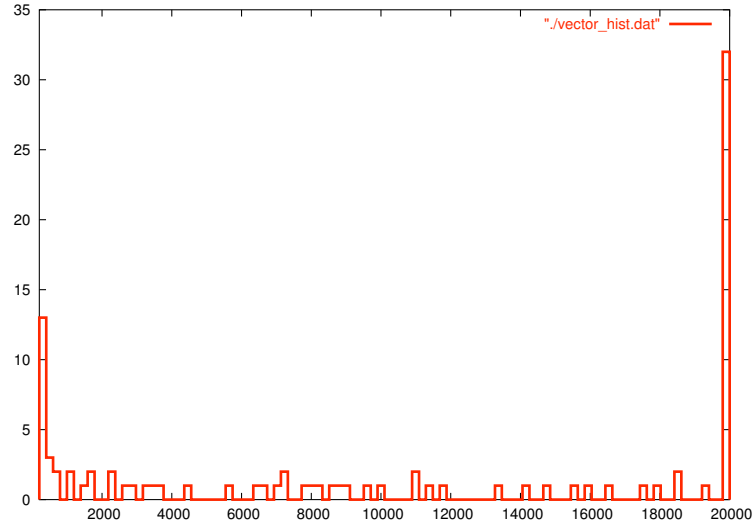


(b)

Figure 17: The distributions of the fragment sizes, measured in number of pixels, for the segmentation thresholds of (a) 43500 and (b) 44000, using 100 bins.

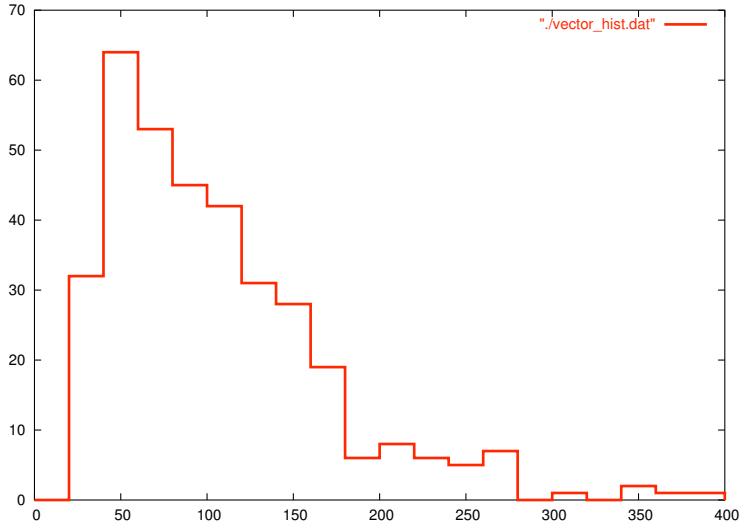


(a)

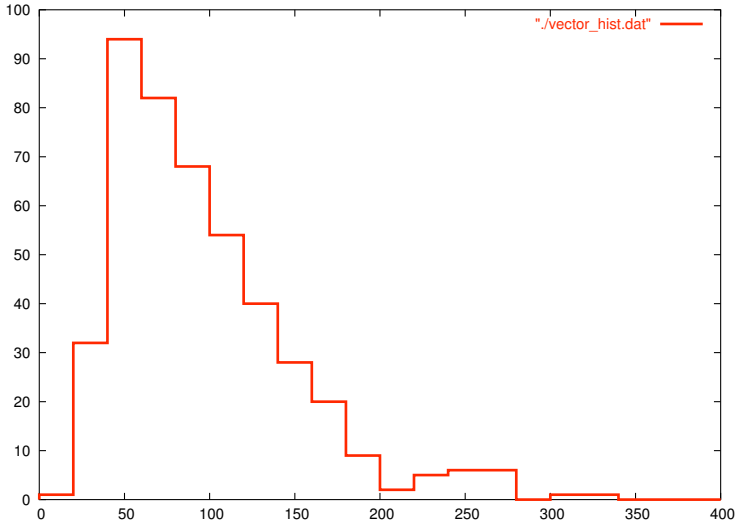


(b)

Figure 18: The distributions of the fragment sizes, measured in number of pixels, for the segmentation thresholds of (a) 43500 and (b) 44000, using 100 bins. All fragments smaller than 200 are counted in the first bin and those larger than 20000 are counted in the last bin.

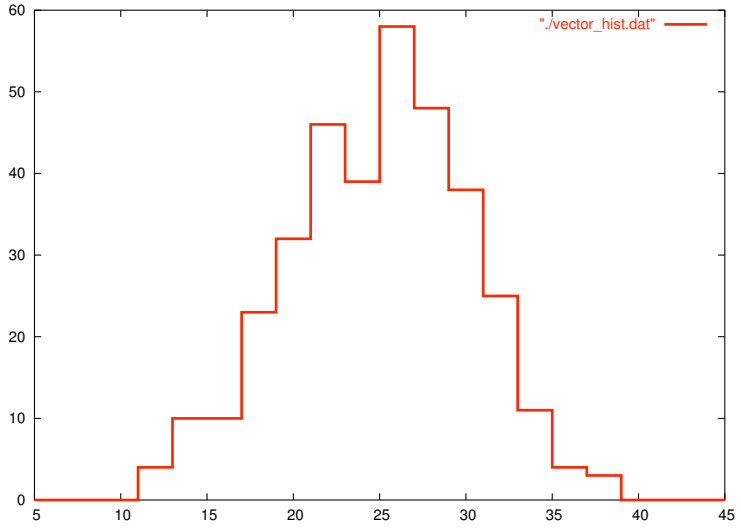


(a)

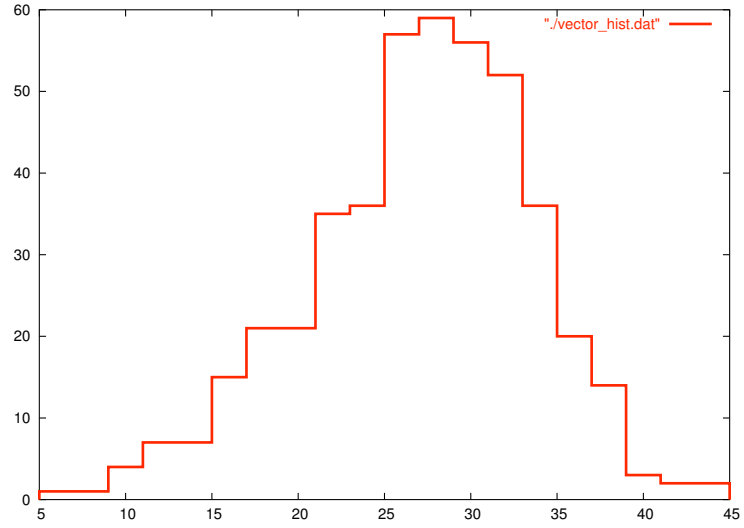


(b)

Figure 19: The distributions of the edge lengths, measured in number of pixels, for the segmentation thresholds of (a) 43500 and (b) 44000, using 20 bins, with the smallest bin at 0 and the largest at 400.

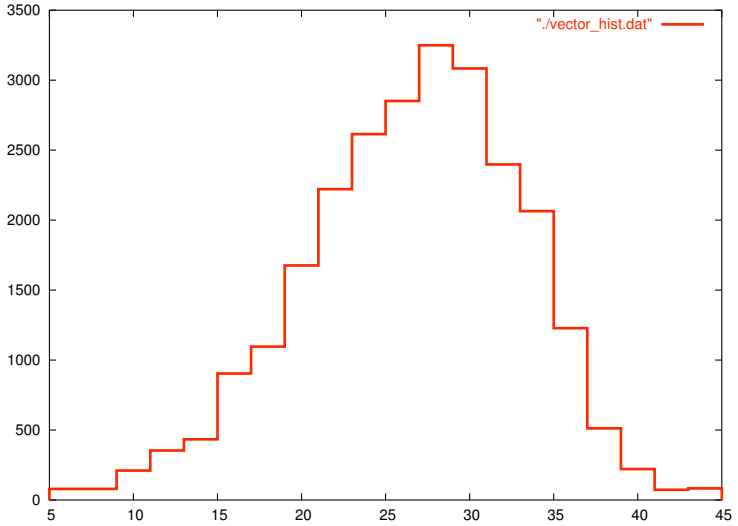


(a)

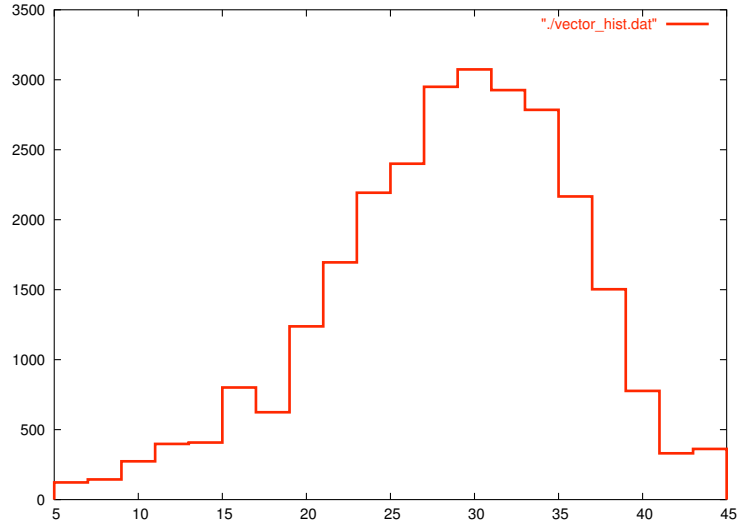


(b)

Figure 20: The distributions of the average edge width, measured in number of pixels, for the segmentation thresholds of (a) 43500 and (b) 44000, using 20 bins, with the smallest bin at 5 and the largest at 45.



(a)



(b)

Figure 21: The distributions of the actual edge width, measured in number of pixels, for the segmentation thresholds of (a) 43500 and (b) 44000, using 20 bins, with the smallest bin at 5 and the largest at 45.

tomography images. The space between the solid particles in a rock, referred to as the pores, can affect macroscopic rock properties. Geometrical properties such as the size, the shape, and the connectivity of the pores, as well as their spatial distributions, are key to determining the transport of fluids through the rock. By first segmenting the image to identify the pores and then obtaining the medial axis of the pore regions, followed by cleanup to handle problems caused by digitization of the images, the authors show how they can extract the characteristics of interest. This problem is more challenging as the images are three-dimensional and the pore cross-section varies along its length.

6 Summary

In this report, we have described the use of image analysis techniques to extract objects of interest, and their characteristics, in images from experiments studying the fragmentation of materials. In comparison with manual approaches, these semi-automated techniques ensure reproducible results. The algorithms used are computationally inexpensive and have few parameters so that they can be easily tuned to the specific problem. We have also found that the results are relatively insensitive to the choice of the parameters, making these techniques a viable alternative for the analysis of such images.

7 Acknowledgment

This analysis was done using the Sapphire scientific data mining software developed at LLNL. We would like to acknowledge the Sapphire team members who contributed to the software and Rebecca Springmeyer and Jeffery Long for their encouragement of this work.

This work was performed under the auspices of the U.S. Department of Energy by University of California Lawrence Livermore National Laboratory under contract No. W-7405-Eng-48.

References

[1] Ahmed, M., and Ward, R. A rotation-invariant, rule-based thinning algorithm for character recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence 24*, 12 (2002), 1672–1678.

[2] Fodor, I. K., and Kamath, C. Denoising through wavelet shrinkage: An empirical study. *SPIE Journal of Electronic Imaging 12*, 1 (2003), 151–160.

[3] Hilditch, C. J. Linear skeletons from square cupboards. In *Machine Intelligence, Volume 4*, B. Meltzer and D. Michie, Eds. Edinburgh University Press, 1969, pp. 404–420.

[4] Jobson, D. J., Rahman, Z., and Woodell, G. A. A multiscale retinex for bridging the gap between color images and the human observation of scenes. *IEEE Transactions on Image Processing 6*, 7 (1997), 965–976.

[5] Jobson, D. J., Rahman, Z., and Woodell, G. A. Properties and performance of a center/surround retinex. *IEEE Transactions on Image Processing 6*, 3 (1997), 451–462.

[6] Kamath, C., and Miller, P. L. Image analysis for validation of simulations of a fluid mix problem. In *Proceedings of the IEEE International Conference on Image Processing* (2007). To appear.

[7] Koh, I., Lindquist, W., Zito, K., Nimchinsky, E., and Svoboda, K. An image analysis algorithm for the fine structure of neuronal dendrites. *Neural Computation 14* (2002), 1283–1310.

[8] Land, E. H. Recent advances in the Retinex theory and some implications for cortical computations: Color vision and the natural image. *Proceedings of the National Academy of Sciences, USA 80* (1983), 5163–5169.

[9] Olsen, S. I. Estimation of noise in images: An evaluation. *CVGIP: Graphical Models and Image Processing 55*, 4 (1993), 319–323.

[10] Prodanović, M., Lindquist, W., and Seright, R. 3D image-based characterization of fluid displacement in a berea core. *Advances in Water Resources 30*, 2 (2007), 214–226.

[11] Rahman, Z., Jobson, D. J., and Woodell, G. A. Multi-scale retinex for color image enhancement. In *Proceedings of the IEEE International Conference on Image Processing* (1996), vol. 3, pp. 1003–1006.

[12] Retinex image processing web page. http://dragon.larc.nasa.gov/retinex/background/background.html, June 2007.

[13] Rockett, P. I. An improved rotation-invariant thinning algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence 27*, 10 (2005), 1671–1674.

[14] Shin, H., Lindquist, W., Sahagian, D., and Song, S.-R. Analysis of the vesicular structure of basalts. *Computers and Geosciences 31*, 4 (2005), 473–487.

[15] Soille, P. *Morphological Image Analysis: Principles and Applications*. Springer, 1999.

[16] Sonka, M., Hlavac, V., and Boyle, R. *Image Processing, Analysis, and Machine Vision*. PWS Publishing, 1999.

[17] Starck, J.-L., Murtagh, F., and Bijaoui, A. *Image Processing and Data Analysis: The multiscale approach*. Cambridge University Press, 1998.

[18] Umbaugh, S. E. *Computer Imaging: Digital Image Analysis and Processing*. The CRC Press, 2005.

[19] Weeratunga, S., and Kamath, C. PDE-based non-linear diffusion techniques for denoising scientific and industrial images: an empirical study. In *Proceedings, Image Processing:Algorithms and Systems, SPIE Electronic Imaging* (2002), vol. 4667, pp. 279–290.

[20] Weeratunga, S., and Kamath, C. A comparison of PDE-based non-linear, anisotropic, diffusion techniques for image denoising. In *Proceedings, Image Processing:Algorithms and Systems II, SPIE Electronic Imaging* (2003), vol. 5014, pp. 201–212.