The Thesis Committee for Jordan Hildebrandt
Certifies that this is the approved version of the following thesis:

3-D Fracture Network Tracing for X-Ray Computed Tomography Data

APPROVED BY
SUPERVISING COMMITTEE:

Richard A. Ketcham, Supervisor

M. Bayani Cardenas

John M. Sharp

3-D Fracture Tracing for X-Ray Computed Tomography Data

by

Jordan Hildebrandt, B.A.; B.S.

Thesis

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Master of Science in Geological Sciences

The University of Texas at Austin

December, 2016

Acknowledgements

I would like to first thank my family for their continual support throughout my graduate school journey – from Walker Brother's apple pancake courier to phone calls. Thank you, all. I love you very much. A special thank you to my Grandpa Bill, who missed the last year of cheering. We all miss you tremendously.

I must thank my advisor, Dr. Rich Ketcham, for just about everything else. He plucked me out from the obscurity of the tiny, liberal arts Wittenberg University and brought me to the forefront of geological research at UT-Austin, specifically to that of X-ray computed tomography imaging. As all my previous advisors are aware, I'm a bit quirky and require no small amount of patience and sense of humor to keep up with, and Rich has been extremely kind and understanding. I have learned so much from him, and I really appreciate his time, effort, and understanding, which was at an extremely high level I felt completely undeserving of in the last few semesters. Thank you also to Dr. Sharp and Dr. Cardenas for agreeing to be part of my committee.

Speaking of advisors, I would like to thank Dr. John Ritter, Dr. Mike Zaleha, Dr. Adam Parker, Dr. John Holbrook, Dr. Alberto Malinverno, and J. Bryan Blair for their role in getting me to grad school in the first place. Also, Philip Guerrero for his help and softball skills once I got there.

Finally, I would like to thank the fine folks of the UT-CT lab for their support and company over the past couple years. Working with such great people has made my experience at UT so positive. Thank you Jessie Maisano, Matthew Colbert, Gary Zuker,

Romy Hanna, David Edey, and Travis Clow. And Julia Holland, who became a close friend without me paying her to.

Abstract

3-D Fracture Tracing for X-ray Computed Tomography Data

Jordan Hildebrandt, MSGeoSci

The University of Texas at Austin, 2016

Supervisor: Rich Ketcham

X-ray computed tomography (CT) is a nondestructive imaging method that shows differences in X-ray attenuation, which is a proxy for density. Sensitivity to density differences makes CT a good choice for imaging open natural fractures because of the significant density contrast between air and rock. The image data, however, are prone to artifacts and blurring, which makes taking accurate measurements challenging. As a first step to addressing this problem, a tool was created to quantify the spatial resolution of CT data with a point-spread function (PSF). The PSF tool permits accurate measurement of fine-scale features in CT data – critical for measuring fractures, which are often thinner in one dimension than the PSF size, in turn influencing measurement of aperture. The difference between the measurements given by the PSF method and a simple threshold value pick is shown to demonstrate a non-trivial improvement in accuracy.

In addition, a 3-D fracture network tracing algorithm was developed, for which the PSF is a necessary input, to characterize accurately the network's attributes, such as

fracture orientations, apertures, and roughnesses. To date, only single, isolated, relatively flat fractures have been characterized thoroughly in 3-D for CT data, and most numerical modeling has been conducted on 2D subsets. Additionally, research involving fracture networks is currently limited to simplified numerical models and simulations. This new work extends CT fracture characterization to the majority of fractured materials with 3-D networks, thus providing a source of real data for studying the difference between fracture networks and single fractures. A sample of Packsaddle schist, with a large number of thin fractures and complex, anastomosing bifurcations, was selected as a testing ground for the network tracing algorithms being developed. Preliminary analysis verifies the method's efficacy.

Table of Contents

List of Tables x

List of Figures xi

Chapter 1: Introduction 1

Chapter 2: Background 4

2.1: Fundamentals of CT Data 4

2.2: CT Scan Artifacts 7

2.3: Obtaining Fracture Network Information 9

2.3.1: Computational thresholding 11

2.3.2: Accurately measuring fracture apertures 13

Chapter 3: Preliminary Investigation 16

3.1: The PSF Tool 16

3.2: How the PSF Varies 17

3.3: The PSF Tool Functionality 19

3.4: Measuring Fracture Apertures: IPSF vs. FWHM 23

Chapter 4: 3-D Fracture Network Tracing in Blob3D 31

4.1: Blob3D and Object-Oriented Programming 31

4.2: The Object-Oriented Paradigm 33

4.3: An Object Model for Fracture Tracing 34

4.4: Fracture Tracing Method 37

4.4.1: User procedure 38

4.4.2: Tracing algorithm walkthrough 40

4.5: Packsaddle Schist Test Data 45

4.6: Discussion 49

4.6.1: Comparing failure conditions 52

4.6.2: Fracture length vs maximum aperture 56

4.6.3: Roughness in 3-D 57

4.7: Future Work 57

4.7.1: Applications 59

Chapter 5: Conclusions 62

Appendix A: Packsaddle Test Images 63

Appendix B: Pseudocode 65

References 69

List of Tables

Table 1. Ten PSF samples taken on the Packsaddle schist sample image.....46

List of Figures

Figure 1: Measuring fractures digitally with CT is advantageous over physical measurements in that it can nondestructively image a 3-D fracture and collect large volumes of fine-scale information. While thresholding is a common, simple way to find fractures in CT, the IPSF method is much more accurate. A PSF tool was created to facilitate determining material boundaries in CT data. The PSF tool was implemented in this thesis to iteratively trace fractures in a slice, given a user input traverse across a fracture, with the intent of being fully automated for 3-D networks. A simplified flow chart of the fracture tracing algorithm described in Chapter 4 is shown on the bottom right. 3

Figure 2: From Ketcham et al. (2010), the plot depicts different aperture measurement techniques. The full-width-half-max (FWHM) works well on large fractures, but on small fractures in which the lowest CT value in the aperture is not reached, it is an underestimation. The peak height (PH) method uses the anomaly height (bolded vertical arrows) and works decently on smaller apertures. The missing attenuation method (MA) is signified by the green shading on the plot, as it uses the entire area of the anomaly and converts that area to a non-blurred equivalent; the MA method performs better on large apertures and is weaker on smaller apertures. 13

Figure 3: Relationship of $r_{PSF,v}$ vs. radial distance for two scans with flat-lying grid phantoms. The upper scan was performed with 800 views, the lower with 2000. Error bars show two standard errors based on nine parallel traverses. 18

Figure 4: The screenshot shows the Inspect Viewer and Line Viewer of Blob3D, with the addition of a PSF measuring tool described in Ketcham and Hildebrandt (2014). The Line ROI tool has been selected and used to draw a traverse across a contact (the green line). The bottom left annotation marks that the line is 1.796mm long on the sample. In the Line Viewer, the far left side shows the PSF value that fits the traverse; first, as a single traverse, then below it as the mean of nine parallel traverses, and below that with the necessary geometric corrections.20

Figure 5: From Ketcham and Hildebrandt, (2014), a schematic of a central traverse surrounded by eight parallel traverses. The set of nine traverses collectively is called a "tile" in this paper. The orange dots represent the midpoints of the fracture aperture that the traverses are intersecting. The gray square represents a best-fit plane calculated from those nine fracture aperture midpoints. The normal vector that describes the best-fit plane is used for calculating orientation of the fracture and in determining the location of the next step to which the tracing algorithm jumps.....21

Figure 6: Relationship of $r_{PSF,v}$ to orientation of traverse to interface, for data with two different phantoms scanned at a tilt. Error bars show two standard errors based on nine parallel traverses. 23

Figure 7: From Ketcham et al. (2010): The top surface of the fracture in the welded tuff sample that was analyzed. 25

Figure 8: CT images of a slice from the welded tuff sample analyzed in Ketcham et al. (2010) displayed in the software package Avizo. The user-selected threshold was gradually increased in an attempt to identify the entire fracture. The FWHM criterion is 20,000, but clearly this underestimates the extent of the fracture. 26

Figure 9: Topological maps of the fracture surface in the welded tuff was generated in Avizo for threshold values of both 20,000, the FWHM criterion (bottom, purple), and 27,000, the point at which the fracture appeared to be represented to the eye (top, blue). 27

Figure 10: Grow and shrink operations were attempted to bridge the jump of a real fracture feature in the center left (having cropped out the parts of the clast that were recognized as part of the threshold, in comparison to the uncropped Figure 5). These operations only served to further distort the measurement accuracy. 28

Figure 11: A two-dimensional view of the difference of the Avizo aperture estimate (for a CT number threshold of 27,000) minus the IPSF measurement. Hotter areas, which cover the majority of the map, represent overestimation by the thresholding. 29

Figure 12: The object model for fracture tracing describes a fracture network as being comprised of "sheets" of fractures. A sheet is built of "tiles," which are sets of nine traverses, or "tracers." Pictured is a schematic for a small fracture network consisting of two intersecting B3D_Sheets, colored red and blue. The sheets each are composed of the square B3D_Tiles, which are defined by a set of nine B3D_Tracers, each represented by an arrow. 36

Figure 13: Example of a fracture bifurcation. The traverse (top) from A to B contains a finger of the lighter matrix material, reflected in the CT values (middle). The plot of the traverse's second derivative (bottom) shows a major swing within the walls of the fracture, from a concave up feature at voxel 10 to a concave down feature at voxel 15, then back to a concave up feature at voxel 20.....42

Figure 14: Two trials of tracing the long, lower fracture feature. The tracing began at the blue line. The traverses are marked in green where the tracing algorithm encountered a failure condition. The top image was created with a reference departure of 3 as the failure condition. The bottom image used a ratio of 6 instead. Point A signifies a failure due to a 3-D pore very large relative to the fracture aperture, Point B signifies a failure due to a bifurcation, and Point C signifies a failure due to termination. 48

Figure 15: Tile data for the aperture relative deviation versus mean aperture. 49

Figure 16: Tile data for the reference departure versus mean aperture. 50

Figure 17: Tile data for the reference departure versus mean aperture for the experiment in changing the ratio restrictions. The blue points are for the run at a ratio of 3, the red for a ratio of 6. 51

Chapter 1: Introduction

The recent boom of hydraulic fracturing in the oil industry has lifted the study of fractures to a new prominence, though the importance of studying fractures is not unique to the energy industry. Other areas of fluid flow research, such as in many aspects of hydrology, study the transport of fluids through fractures because when fractures are present, they are generally the predominant conduits for transporting fluids, solutes, and contaminants. For any discipline, quantifying natural fracture networks *in situ*, however, has remained a thorny challenge. It is difficult to take measurements on a natural fracture network without disturbing or destroying the sample in the process.

Park (1997) defines a fracture as "a crack across which the cohesion of the material is lost, and may be regarded as planes or surfaces of discontinuity;" this generally manifests itself as a planar pore structure. A fracture network is a set of connected fractures.

Fractures can be described in terms of their:

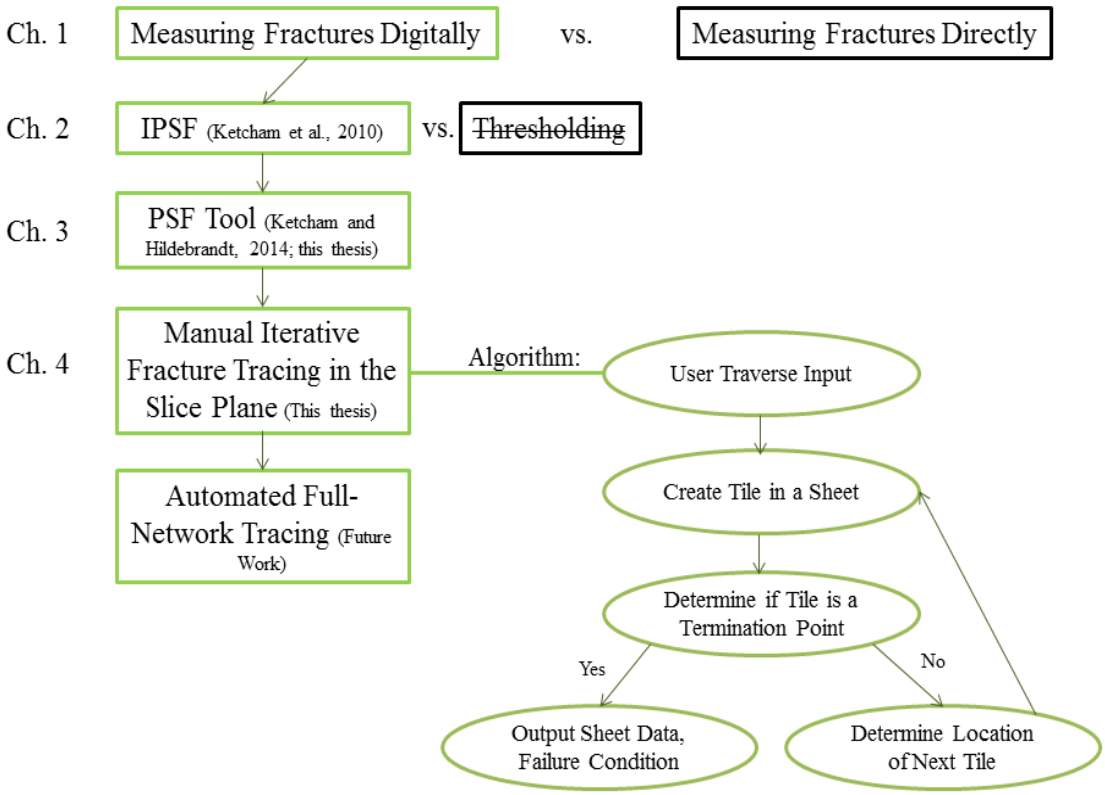
1) Orientation – In simple cases, this might be described with terms like strike and dip, though curviplanar geometries complicate this,
2) Aperture – The distance between the two walls of the fracture,
3) Roughness – The degree to which fracture walls depart from being planar,
4) Density – The number of fractures present per volume of rock, and
5) Connectivity – How many fracture branches are connected to other branches within the network,

among other attributes. The study of fractures has been performed with the use of many

different models, including the parallel plate model, equivalent porous media, discrete fracture, theoretical, double-porosity, and equivalent parallel plate (Sharp, 1993). With advances in X-ray computed tomography (CT) imaging technology, however, simulations can be performed and hypotheses can be tested on precise data taken from real natural fractures and fracture networks.

Imaging natural fracture networks in the field in detail is currently unfeasible, but samples brought back to a CT lab can be scanned nondestructively, producing a high resolution image volume. The images acquired from the scanning can be mined for data to identify and quantify the fractures digitally (e.g., Vandersteen et al., 2003, Slottke, 2010, Ketcham and Hildebrandt, 2014). The digital representation of the fracture network can then be analyzed for geometric properties and used as input for simulations (e.g., Cardenas et al., 2007, Watanabe et al., 2011, Wang and Cardenas, 2014). However, the current methods for measuring the attributes of entire fracture networks are somewhat limited in scope and accuracy. Figure 1 provides a general schematic for the work of this thesis.

Figure 1: Measuring fractures digitally with CT is advantageous over physical measurements in that it can nondestructively image a 3-D fracture and collect large volumes of fine-scale information. While thresholding is a common, simple way to find fractures in CT, the IPSF method is much more accurate. A PSF tool was created to facilitate determining material boundaries in CT data. The PSF tool was implemented in this thesis to iteratively trace fractures in a slice, given a user input traverse across a fracture, with the intent of being fully automated for 3-D networks. A simplified flow chart of the fracture tracing algorithm described in Chapter 4 is shown on the bottom right.



Chapter 2: Background

2.1: Fundamentals of CT Data

A nondestructive technique, X-ray CT provides 3-D image data of the interiors of solid objects based on X-ray attenuation, which is a proxy for density. The higher a material's density, the more X-ray radiation the material will attenuate. Higher attenuation values (higher CT numbers) are represented in a CT image by brighter signals. Less dense, less attenuating materials will appear darker. The air within a fracture attenuates very little compared to the surrounding rock matrix, so the contrast is usually very high. A high contrast between the two materials means that the fractures in CT images will be readily visible to the naked eye even if the aperture is a small fraction of the width of a pixel (Ketcham and Carlson, 2001).

CT data is generated by projecting X-rays into an object from a range of different directions. The X-ray beam is attenuated by the object, and the signal that passes through the object is registered on a detector. Assuming the X-ray source is monochromatic and well-collimated, a homogeneous material attenuates the X-ray beam from the source in a manner according to Beer's Law, (Wellington and Vinegar, 1987):

(1) $I/I_0 = e^{-\mu h}$, where:

I = the intensity of the X-ray entering the material,

I_0 = the intensity of the X-ray leaving the material,

μ = the material's linear X-ray attenuation coefficient, and

h = the thickness of the sample.

For a heterogeneous material, the attenuation can be determined by replacing the

exponent in Beer's Law, $^{-\mu h}$, with the sum of this product of attenuation times thickness for each material (Johns et al., 1993).

Many different types of CT scanners exist. Some scanners utilize a rotary turntable for the sample, and the source and detectors remain stationary; others rotate the source and detectors around the object being scanned. There are also different beam types for the source. Some scanners have a fan beam that projects X-rays out in a wide fan-shape pattern to image a single slice at a time, whereas others use a cone beam to image multiple slices simultaneously. The advantage of the volume CT cone beam is the ability to scan multiple slices at a time, yielding shorter scan times. However, the reconstruction for a cone beam is more prone to distortion and scattering artifacts, and is more computationally complex. Whatever the type of scanner, the main similarity is that a beam is projected thousands of times through the object, generating many different "views" (Ketcham and Carlson, 2001).

A view is displayed in an image called a sinogram, so called because any point within the scanned object corresponds to a sinusoidal curve. The raw data in a sinogram is constituted by each line representing a single set of detector readings in a particular view. To see the sample, the sinograms must be converted into slice images through a reconstruction process. Filtered backprojection is a common reconstruction technique. In this technique, the sinogram data are convolved with a filter, for example the Laks filter (Ramachandran and Lakshminarayanan 1970). Then, each view is placed on a square grid at the angle that corresponds with the angle at which it was acquired. The attenuation information in the sinogram is converted through this process to a 16-bit

range of values, giving a range of CT number values between 0 and 65535. To convert to 8-bit data, this range is scaled from 0 to 255. The variability inherent in the scanning process – including the scanners, operators, reconstruction techniques, etc. – means that the CT value for a material in the same exact sample scanned on two different occasions may vary from day to day or from lab to lab. In many cases, the density of a material does not correspond to a standard scale value of CT numbers. Therefore, only looking at a particular CT number is not a reliable way to determine what a certain material is, though the relative values will maintain a similar relationship (Ketcham and Carlson, 2001).

The noise and blurring induced by aspects of the scanning process is a major influence on the CT number. Blurring simply means that the CT number of any given voxel is a function of the material in that particular voxel and surrounding voxels as well. Blurring has many different sources, including large X-ray focal spots, geometric imprecision, anomalies in the reconstruction algorithm, detector cross-talk, system noise, and the partial-volume effect (PVE). The PVE describes the phenomenon where a single voxel has more than one material represented within it.

Fortunately, the blurring from all these sources can be well-approximated as a Gaussian point-spread function (PSF) (ASTM, 1992). The PSF is a 2-D function for CT acquired by the slice, but for volume CT data it is 3-D and often anisotropic. Ketcham et al. (2010) describe the conceptual model as the convolution operation:

(2) $F = G * PSF + \varepsilon$, where:

F = the data obtained by the scanner

G = the reality of the data (what F "should be," if it was not blurred and noisy)

PSF = the point-spread function associated with the scan, and

ε = an overprint of random noise.

In other words, reality smeared by the blurring inherent in the scanning process, plus some extra noise, equals the data the scanner provides as an image. This model works for most cases; the PSF and its utility are described in more detail in Section 3.1 of this paper.

2.2: CT SCAN ARTIFACTS

Prevalence of scan artifacts is one problem that interferes with the underlying PSF model. Beam hardening is the most common artifact. The effects of beam hardening are recognized by examining if the edges of an object are brighter than the center, and not due to any change in material. As a polychromatic X-ray beam passes through the object, the beam gets attenuated more and more, with the lower energy part of the X-ray spectrum being more readily attenuated first. The remaining beam is "harder" because the beam contains a greater percentage of the higher energy range of the X-ray spectrum. The upshot of beam hardening is the same material will have different CT values throughout the image, another complication in trying to use a standard CT value reference scale (Ketcham and Carlson, 2001).

No perfect solutions for beam hardening exist, though there are several methods to help reduce its effects. The simplest method to lessen the impact of beam hardening is to use a higher energy X-ray source so less of the lower-energy X-rays in the beam will be attenuated. Alternatively, the beam can pre-filtered by passing them through an

attenuator that eliminates low-energy X-rays, though this might not eliminate the artifact and just lessen it (Ketcham and Hanna, 2014). Another way is to use a "wedge calibration;" a wedge is a cylinder-shaped sample of a material with a similar attenuation coefficient to the scan object (Ketcham and Carlson, 2001). Scanning the wedge will provide a basis of comparison to calibrate the original target object. Bertels et al. (2001) encountered beam hardening artifacts in their measurement of fracture aperture distribution in CT data. To combat the artifact problem, they placed the sample core within an aluminum sleeve to displace the hardening effects away from the core itself. A new method for beam hardening correction was developed that iteratively finds an artifact-minimizing function; this computational, iterative solution to beam-hardening has the benefit of adapting to the varying influences of all the materials in a heterogenous sample (Ketcham and Hanna, 2014). Thematically, the beam hardening correction proposed by Ketcham and Hanna (2014) is similar to the PSF tool of Ketcham and Hildebrandt (2014) detailed in Section 3.1: the aim of both is to limit the *ad hoc* influence of a user on selection criteria and obtain rigorous, quantifiable results.

The other common type of artifact is ring artifacts, which manifest as concentric circles in the middle of the image where the rotation axis was centered. Shifted outputs from the X-ray detectors cause these artifacts. Diverging detectors is caused by ambient factors like temperature as well as varied sensitivity to changing hardness of the X-ray beams (Ketcham and Carlson, 2001). Many of the same methods that apply to beam hardening corrections also apply to fixing or lessening the impact of ring artifacts: using higher energy X-ray sources, pre-filtering the X-rays through an attenuator, and scanning

a wedge for calibration. Both mild beam hardening and ring artifacts are present in the Packsaddle schist sample scanned for this paper. The method described in this paper to identify fractures and to obtain information about fracture characteristics still works reasonably well, even in the presence of artifacts.

2.3: Obtaining Fracture Network Information

The first step in obtaining information about fracture network characteristics is to identify the region corresponding to the fracture. The most basic identification technique is some variety of thresholding, which assigns each individual voxel to be either part of the fracture or not using a demarcation boundary CT value. Sezgin and Sankur (2004) identified six different types of thresholding algorithms; they grouped them into the following categories: histogram, clustering, entropy, object attribute, spatial, and local. Otsu (1979) is a clustering-based algorithm. The full-width-half-maximum (FWHM) is a histogram-based approach to determining a threshold value; the FWHM threshold is defined as the median value between the grayscale of the matrix and the air (Toda et al., 2011).

The problem with most of these thresholding algorithms is they do not function very well on heterogeneous samples and instead operate under the assumption that the materials on both sides of the fracture are the same. While thresholding algorithms are typically quick, easy, and effective on clean, homogeneous datasets without prevalent artifacts and blurring, in the case of imaging natural fractures, the desire to maximize the spatial extent imaged to get more data on area, roughness, connectivity, and overall representativeness often in practice leads to the apertures being near the edge of

resolution. Although the contrast is good between the fracture and the surrounding matrix, fractures are thin in one dimension, so they are highly influenced by blurring inherent in the scan. Thin features in CT data are especially prone to blur because the feature is smaller than the PSF, and in more extreme cases, the voxel resolution, so the smearing induced by blurring is more pronounced. Multiple materials then are represented in the same voxel (air and rock); this occurrence is known as the partial-volume effect. The attenuation for a sample experiencing the PVE is calculated by the volume fraction of the components multiplied by the attenuation response of their corresponding pure components (Vinegar and Wellington, 1987).

There is no correct number by which to segment thin features in CT data. The FWHM method fails when employed on apertures that are thin compared to the resolution of the data. As a result, the exercise of picking a threshold is left up to the individual performing the analysis, and can thus lead to widely varying results. In fact, one study tested the user bias thresholding problem with thirteen different experts, mostly specializing in soil science (Baveye et al., 2010). In this experiment, each expert was provided images of soil and asked to segment them using any method of their choosing and at the end provide a porosity measurement for the image. For one image, porosity measurements ranged from 24% to 75.5%; the mean porosity was 53.2% with a standard deviation of 20.47. In another, porosity estimates ranged from 12.9% to 72.7%; the mean porosity was 53.5% with a standard deviation of 14.36%. Only about half of the estimations were within 14% of the physically-determined porosity. Such large uncertainty clearly needs improvement. Section 3.4 in this paper provides a fracture

aperture example for how user input for the thresholding step drastically influences the end calculation.

2.3.1: Computational thresholding

For obtaining greater precision, an alternative to identifying the region in the data by thresholding is using some form of interpolation. One promising segmentation of thin features was performed with a Chan-Vese minimal variance method (Holtzman-Gazit et al., 2006). This approach treats the segmentation of adjacent objects with similar grayscale values as an energy minimization problem. The authors have developed and made freely available codes to perform both two-medium and multiple-medium extractions, i.e. segmenting more than just two different features in a dataset. They performed their analysis on medical images, separating blood vessels from the surrounding tissue using variational calculus. Seeds are placed on the image in the feature of interest; then, a surface corresponding to the feature grows from the seeds, creating a surface. This surface, however, is prone to splitting into discontinuous segments when the features are small compared to the sampling rate. Thus, a single blood vessel was incorrectly represented as multiple surfaces, which does not provide a description of the feature consistent with reality. Additionally, solving these partial differential equations on larger datasets, typical for geomaterials scans, would be extremely computationally expensive.

A less computationally expensive alternative to analyzing volumetric CT data to find fractures is the use of Hessian matrices (Voorn et al., 2013). Developed with the explicit concern of being able to use an ordinary desktop computer, their method involves

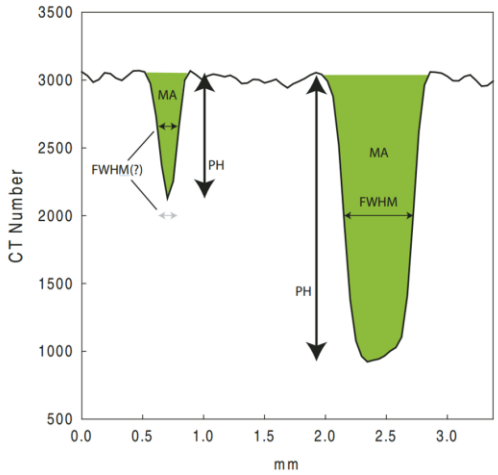
describing the local curvatures of data at each point in an image with the Hessian matrices. For 3-D image data, a Hessian matrix is a 3x3 symmetric matrix with the second order partial derivatives of the data, meaning that it "describes the second order structure of intensity variation around each point of a 3-D image" (Voorn et al., 2013). The eigenvalues of the Hessian matrix represent the magnitude of the largest local variation in grayscale contrast in each direction. Because the eigenvalue information is three-dimensional, the decomposition of the eigenvalues can be used to determine if a feature in the data is shaped like a blob, tube, or plane. Eigenvalues with high and positive magnitude represent thin fracture feature, and this relationship exhibited by Hessian matrix eigenvalues were applied to filter thin fractures from CT image data with good results for planar fracture features. However, the matrices are not very effective at intersections and termination points since those types of locations are not planar. The worst-case scenario is a 90-degree intersection between abutting fracture planes, at which the segmentation is distorted.

A precursor quantitative calibration to measuring fracture apertures on CT data was carried out by Johns et al. (1993). This study used two naturally fractured granite cores to study this problem. The method implemented is based on missing attenuation (MA). According to the partial volume effect, the voxels in the fracture gap correlate linearly with the material's attenuations. Integrating across the fracture valley sums up the missing attenuation that would equal the matrix CT values if the fracture was not present. A linear least squares equation was fitted to the integrated values, and the fracture apertures were calculated from this equation.

2.3.2: Accurately measuring fracture apertures

Another approach to measuring aperture width is the peak height (PH) method. PH uses the difference between the CT values of the matrix on the edges of the fracture and the minimum CT value reached by the fracture. Vandersteen et al. (2003) showed that PH results are less noisy for smaller apertures, but MA works more effectively on larger ones. The weakness of both the MA and the PH method is encountering heterogeneous materials, namely with different materials on either side of the fracture, as an accurate MA or PH calculation assumes they are the same in order to integrate or fit a Gaussian point spread function (PSF) respectively. Both of these techniques are illustrated in Figure 2.

Figure 2: From Ketcham et al. (2010), the plot depicts different aperture measurement techniques. The full-width-half-max (FWHM) works well on large fractures, but on small fractures in which the lowest CT value in the aperture is not reached, it is an underestimation. The peak height (PH) method uses the anomaly height (bolded vertical arrows) and works decently on smaller apertures. The missing attenuation method (MA) is signified by the green shading on the plot, as it uses the entire area of the anomaly and converts that area to a non-blurred equivalent; the MA method performs better on large apertures and is weaker on smaller apertures.



A combination of the MA and PH methods called the "Inverse PSF" method has the strengths of both and minimizes their weakness to heterogeneous materials (Ketcham et al., 2010). The basic concept is a deconvolution operation that takes advantage of the previously discussed relationship between the blur inherent in the scan being represented by the PSF. Assuming the PSF is uniform and isotropic throughout the data, (this assumption is not always the case – see: Ketcham and Hildebrandt (2014)), deconvolving the CT image with the PSF accounts for the inherent blurring and yields the "real" boundaries of the sample materials (Equation 2). Performing this operation helps counteract the PVE and account for heterogeneities. The deconvolution used by the IPSF technique is not based on the Fourier transform as the typical numeric Richardson-Lucy deconvolution (Lucy, 1974). The formula is based on an underlying assumed model of the PSF and the physical reality of an aperture valley between two materials that are not necessarily the same; the procedure outputs each boundary of the fracture explicitly, a benefit that counteracts the increased computational cost (Ketcham et al., 2010).

This was successfully implemented for a roughly planar fracture (Ketcham et al., 2010), but a fracture network is much more complex than a single plane, and requires a way to trace out the entire system of connected fracture planes that comprise a larger network. The goal of this paper is to describe a methodology to ultimately trace entire fracture networks without limiting the scope to linear features and 2-D subsets. Once the fracture network has been traced, the series of constituent points are available for further analysis. However, the complications of natural fractures render this work an incremental step rather than a full solution. There is still ongoing work that will improve

the tracing algorithm for the large range of complications natural fracture networks will present, such as more complex 3-D bifurcations and large pores connected to fractures. Once these challenges are resolved, the software will function as general tool – providing any branch of research studying fractures the capability to extract information related to the 3-D fracture network.

Models of fracture networks will only ever be as good as the data upon which they are based. While CT instrumentation has greatly improved over the last decade, there are still limitations to data resolution, namely with features that are thin compared to the PSF. Traditional segmentation algorithms do not properly counteract the blurring of the PSF, causing widely inaccurate fracture aperture measurements. This problem has been addressed (Ketcham and Hildebrandt, 2014; Ketcham et al., 2010) in the context of standardizing calibration procedures and measuring simple planar fracture apertures. This paper extends the idea of determining fracture aperture on the basis of the PSF into 3-D fracture networks, thus providing accurate, real fracture network inputs for further topology, modeling, and simulation studies.

Chapter 3: Preliminary Investigation

3.1: The PSF Tool

The PSF can be thought of as the number of voxels it takes to transition from one material to another in a CT dataset. In reality, the boundary between two materials is very distinct. In a digital scan, the boundaries are blurred. Lower scan quality produces larger PSF values because the boundary between two materials is blurred over a larger number of voxels. Higher quality scans have less blurring and sharper distinctions between materials, thus a lower value PSF.

A tool to measure the PSF provides a simple way to ascertain the resolution quality of CT data; the work described in this chapter for creating the tool was published as Ketcham and Hildebrandt (2014). The tool facilitates quantification of the spatial resolution of CT data by its PSF, in which the user draws a traverse across a sharp interface between two materials and a Gaussian PSF is fitted to the blurring across that interface. Geometric corrections account for voxel shape and the angle of the traverse to the interface, which does not need to be orthogonal. The PSF is a necessary input into the 3-D fracture tracing algorithm, so an easy measuring tool for this parameter was a prerequisite to accurately measuring 3-D fractures. Additionally, this development is important because despite the proliferation of scanners and data processing methods and tools, and scientific studies utilizing them, relatively little emphasis has been given to ensuring that results are comparable or reproducible. The tool described can be used to both calibrate processing algorithms that use deconvolution operations and help evaluate

scans on a routine basis within and between CT research groups – all with respect to the features within the imagery that are being measured.

The one-dimensional Gaussian form for the PSF in a discrete voxel domain is defined as:

(3) $g_i = \exp(-8(x_i - r_c)^2 / r_{PSF,v}^2)$, where:

g_i = the PSF value at voxel coordinate i,

$r_{PSF,v}$ = the PSF parameter, in voxel edge lengths, in the x-y plane,

r_c = the smallest integer greater than or equal to r, and

x_i = integer voxel coordinates from 0 to $2r_c$, which is the convolution kernel size. The factor of 8 creates a Gaussian function out to approximately four standard deviations, which ensures smoothness. The variable $r_{PSF,v}$ is interpreted as 96% of the signal from a point will be within a region with that voxel diameter. In other words, most of the transition between materials occurs over an $r_{PSF,v}$ number of voxels. Dividing by the voxel dimension will yield the geometric r_{PSF}.

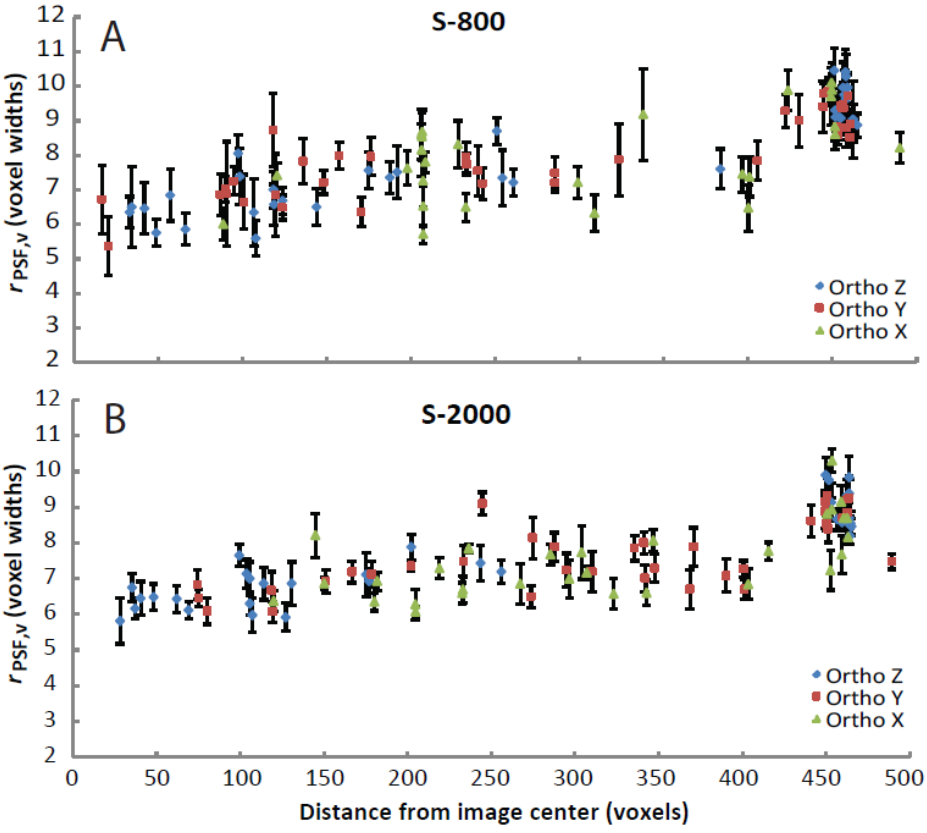
3.2: How the PSF Varies

The PSF measurement does not work in all cases, namely when the volume contains features, such as scan artifacts, that are not well-accounted for in the underlying blurring model. In general, whenever possible the PSF should not be measured across scan artifacts, such as beam hardening and ring artifacts. The effects of artifacts are not as severe on the IPSF method as they are on the MA method, but still noticeable (Ketcham et al., 2010). It is uncertain how to best quantify the PSF in phase-contrast data, a type of CT image data produced by synchrotron particle accelerators, as in

Cloetens et al. (1996), so not all CT imaging techniques are suitable.

The PSF is not constant; it (i.e., the $r_{PSF,v}$) will be larger on less resolved images. The PSF can vary with distance from the center of the image, with a gradual increase towards the edges of the image, as shown in Figure 3. The increase in PSF toward the image edge occurs because the volume is more heavily sampled in the center during the scanning process.

Figure 3: Relationship of $r_{PSF,v}$ vs. radial distance for two scans with flat-lying grid phantoms. The upper scan was performed with 800 views, the lower with 2000. Error bars show two standard errors based on nine parallel traverses.



The characteristics of the detector used will influence the PSF. For area detectors, the

blurring caused by focal spot size and detector crosstalk is similar in the x, y, and z directions relative to the rotation axis. A collimated source that scans slice-by-slice will cause very low inter-slice blurring, so the PSF will be small in the z-direction compared to the x-y directions within the slice plane. The PSF changes with scanner hardware, e.g. spot size, and software calibration corrections, thus necessitating being able to measure the PSF on any dataset instead of only phantoms. The PSF also varies from scanner to scanner, and from scan to scan based on the individual scanning conditions. Standardizing a measurement for a scan quality was the main objective behind the PSF tool addition to the Blob3D software (Ketcham, 2005).

3.3: THE PSF TOOL FUNCTIONALITY

The PSF tool introduced in Ketcham and Hildebrandt (2014) allows a user of Blob3D to draw a traverse across an interface between two materials directly on an image slice. A screenshot example of its usage is highlighted in Figure 4.

Figure 4: The screenshot shows the Inspect Viewer and Line Viewer of Blob3D, with the addition of a PSF measuring tool described in Ketcham and Hildebrandt (2014). The Line ROI tool has been selected and used to draw a traverse across a contact (the green line). The bottom left annotation marks that the line is 1.796mm long on the sample. In the Line Viewer, the far left side shows the PSF value that fits the traverse; first, as a single traverse, then below it as the mean of nine parallel traverses, and below that with the necessary geometric corrections.

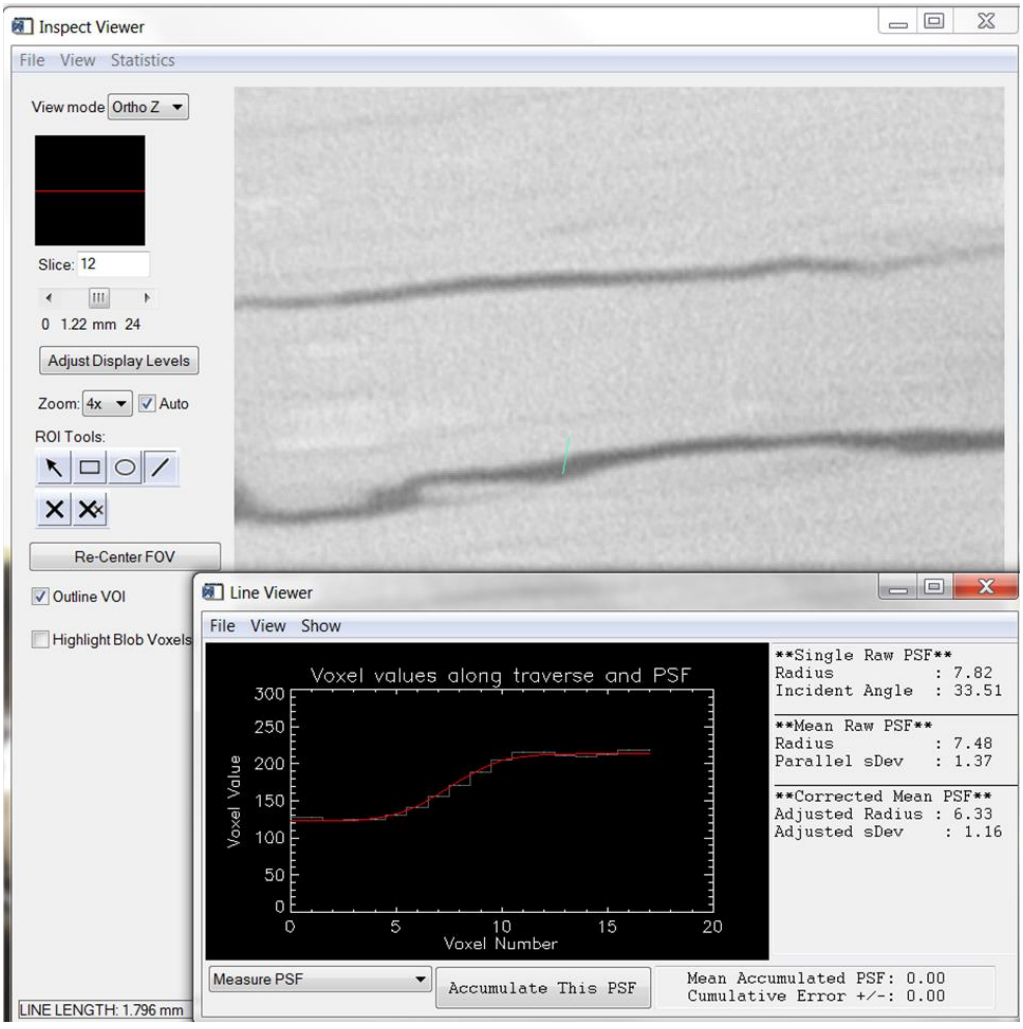
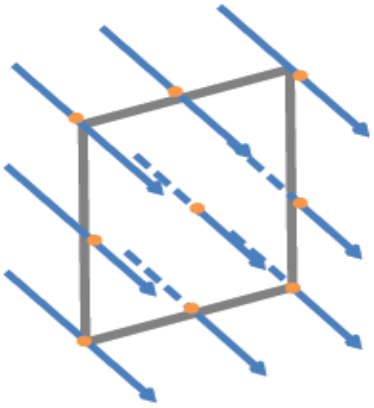


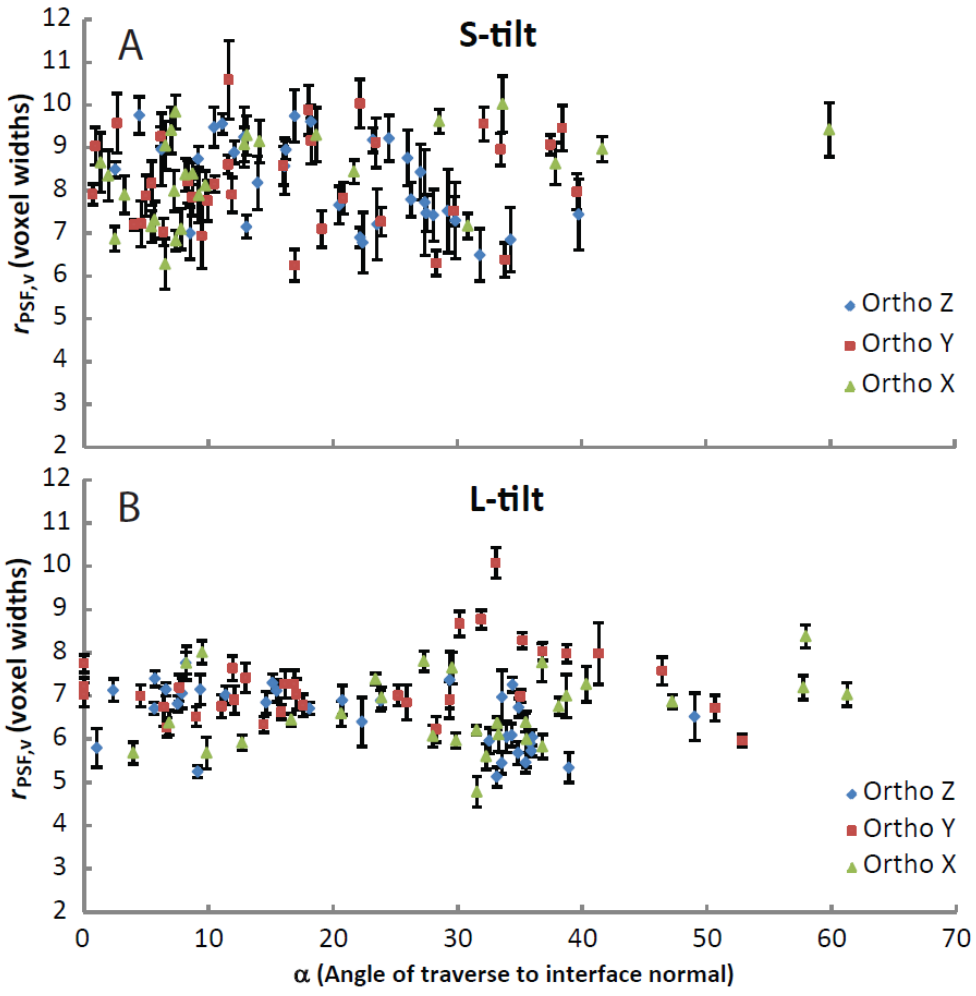
Figure 5: From Ketcham and Hildebrandt, (2014), a schematic of a central traverse surrounded by eight parallel traverses. The set of nine traverses collectively is called a "tile" in this paper. The orange dots represent the midpoints of the fracture aperture that the traverses are intersecting. The gray square represents a best-fit plane calculated from those nine fracture aperture midpoints. The normal vector that describes the best-fit plane is used for calculating orientation of the fracture and in determining the location of the next step to which the tracing algorithm jumps.



The program calculates the PSF for the user-drawn traverse by first determining which voxels the user selected using a modified version of the Bresenham (1965) algorithm that will sample non-isotropic voxel sizes. An iterative function fits the traverse array of CT values by comparing it to a series of model traverses convolved with a certain radius PSF until it arrives at a good fit. As a measure of uncertainty, the PSFs are calculated for eight parallel traverses to the original user-specified traverse (Figure 5). The mean of these nine PSF results are reported as the average PSF for that traverse, along with the associated standard deviation. After the average raw PSF is calculated, the program implements a geometric correction to account for a traverse not drawn orthogonal to the interface. This was demonstrated to yield the same PSF regardless of the angle of

incidence: Figure 6 shows $r_{PSF,v}$ as a function of angle between traverse and interface normal. Although the data show scatter in excess of that predicted by the error bars, there is no systematic change with interface angle, or with respect to which orthogonal plane the measurement is taken from. It can thus be inferred that the angular corrections for non-normal traverses and anisotropic voxels are generally accurate and do not impart bias. Verification was performed by taking repeated measurements at various angles through the same interface location. The user can "accumulate" multiple traverses to calculate an average PSF for the dataset.

Figure 6: Relationship of $r_{PSF,v}$ to orientation of traverse to interface, for data with two different phantoms scanned at a tilt. Error bars show two standard errors based on nine parallel traverses.



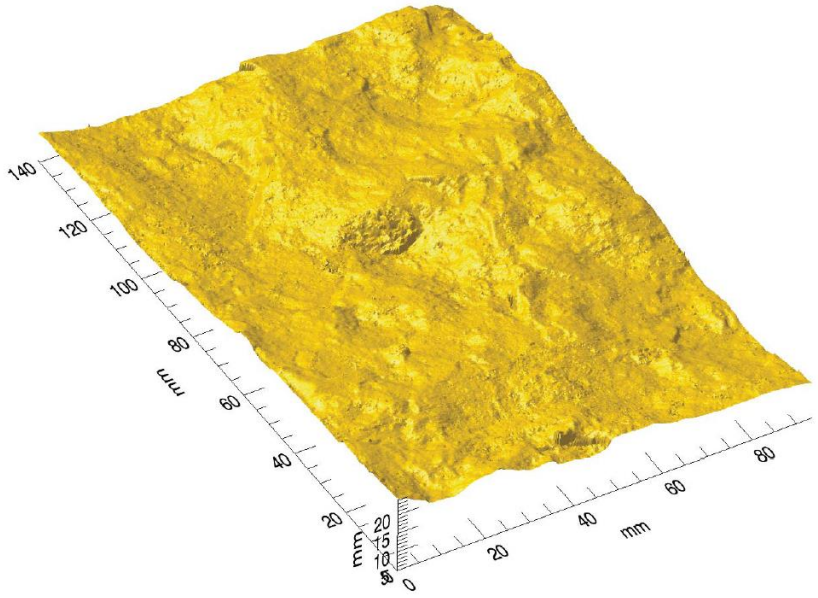
3.4: MEASURING FRACTURE APERTURES: IPSF VS. FWHM

In this section, the aperture of a fractured rock is measured using two different procedures – one based on the principles of the PSF and the other using a constant user-chosen threshold value. The work described in this section was presented by Ketcham et al. (2013). This experiment is demonstrative of the significant inaccuracies faced when trying to get a quick and easy answer for a fracture's apertures and roughnesses. Because

most ways to obtain this information are susceptible to user input bias, the improvement is to generate reproducible, unbiased measurements that reflect the reality of the physical system being measured. The previously mentioned soil porosity study (Baveye et al., 2010), demonstrated the need for a standardization, with over a dozen experts each finding widely different soil porosity with varying software tools. As those experts found it difficult to estimate porosity, it is shown here that it is also difficult to measure fracture apertures due to similar problems with observer variability.

For this test, results from a scanned fractured welded tuff were used, in which the IPSF method was used to generate a 2-D surface map of the apertures (Ketcham et al., 2010). The surface map showing the aperture thickness, in mm, is shown in Figure 7. The same CT data was imported into Avizo, a software program for 3-D data visualization. Figure 8 shows an attempt to pick a threshold value that would represent the fracture apertures. The "correct" threshold value should theoretically be the midway point between the mean value of the surrounding rock and the mean value of the air. In this case, the matrix CT value was around 32,000, and the air value was about 8,000, so the proper threshold value should be about 20,000.

Figure 7: From Ketcham et al. (2010): The top surface of the fracture in the welded tuff sample that was analyzed.



Unfortunately, determining fracture roughness and aperture is not as simple as picking a voxel value. As shown in Figure 8, the theoretically "correct" segmentation value from the FWHM method does not incorporate the entire set of voxels that define the fracture. In 3-D, the inaccuracy of this value is even more obvious. Figure 9 depicts the aperture map generated from a 20,000 threshold value. A large portion of the fracture sheet is absent. The next logical step for someone trying to perform a quick segmentation would be to increase the threshold value. Unlike using the value halfway between air and rock, arbitrarily increasing the value used is non-reproducible. The new threshold will recognize more of the fracture, at the expense of overestimating the thickness in places. Adjusting the threshold value in Avizo to a value that covers most of the visible fracture features occurs at a value of about 27,000 (Figure 9). The non-fracture feature was

cropped out of the fracture system. Growing and shrinking operations (Figure 10) were not particularly effective, as they contracted and swelled the feature unrealistically while still not accounting for the entirety of the fracture. The 27,000 value resulted in a more complete aperture map, but in the attempt, some portions of the fracture still registered an aperture of 0, and much of the apertures were an overestimation of the values measured by the more accurate IPSF method.

Figure 8: CT images of a slice from the welded tuff sample analyzed in Ketcham et al. (2010) displayed in the software package Avizo. The user-selected threshold was gradually increased in an attempt to identify the entire fracture. The FWHM criterion is 20,000, but clearly this underestimates the extent of the fracture.

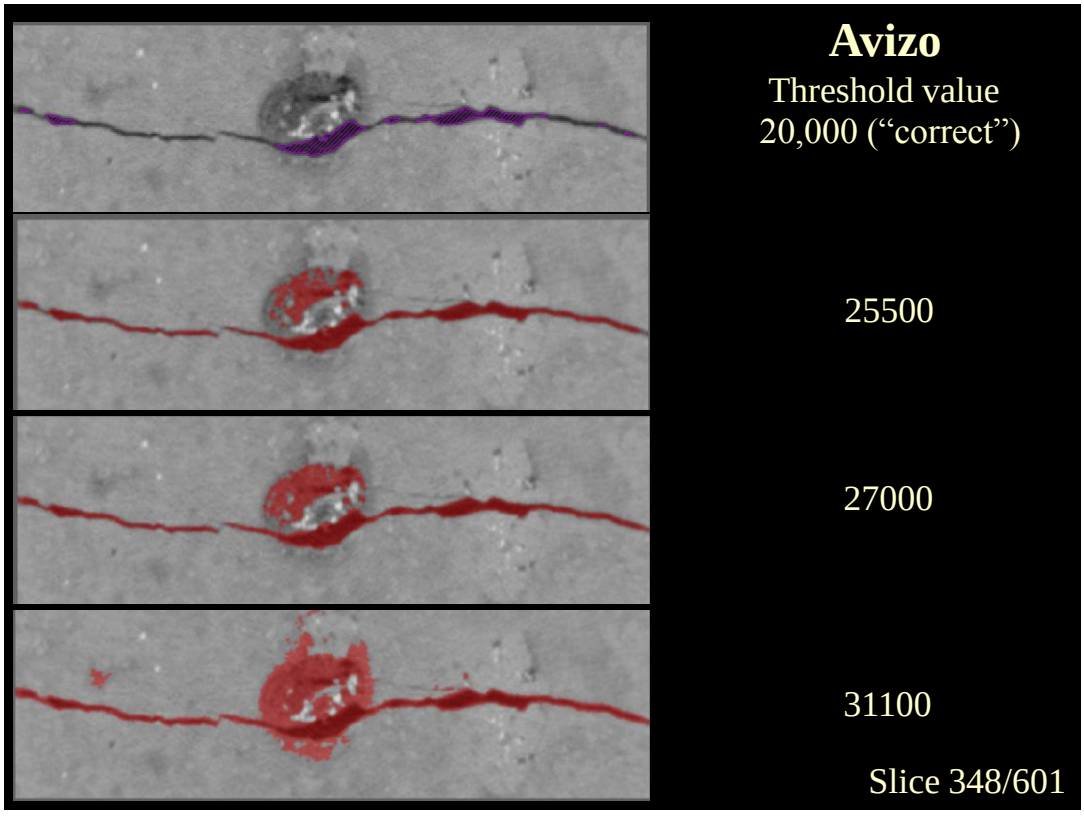


Figure 9: Topological maps of the fracture surface in the welded tuff was generated in Avizo for threshold values of both 20,000, the FWHM criterion (bottom, purple), and 27,000, the point at which the fracture appeared to be represented to the eye (top, blue).

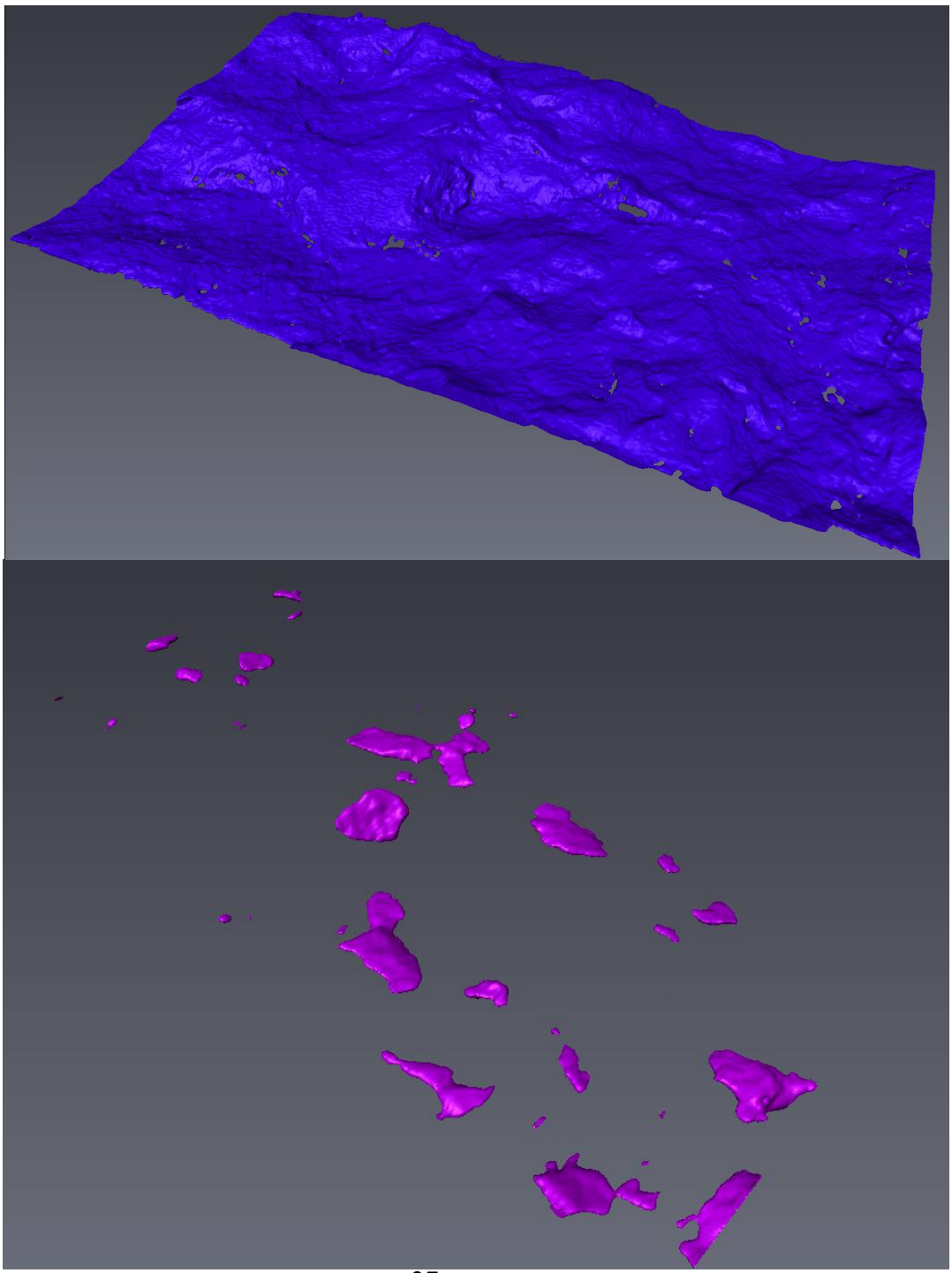


Figure 10: Grow and shrink operations were attempted to bridge the jump of a real fracture feature in the center left (having cropped out the parts of the clast that were recognized as part of the threshold, in comparison to the uncropped Figure 5). These operations only served to further distort the measurement accuracy.

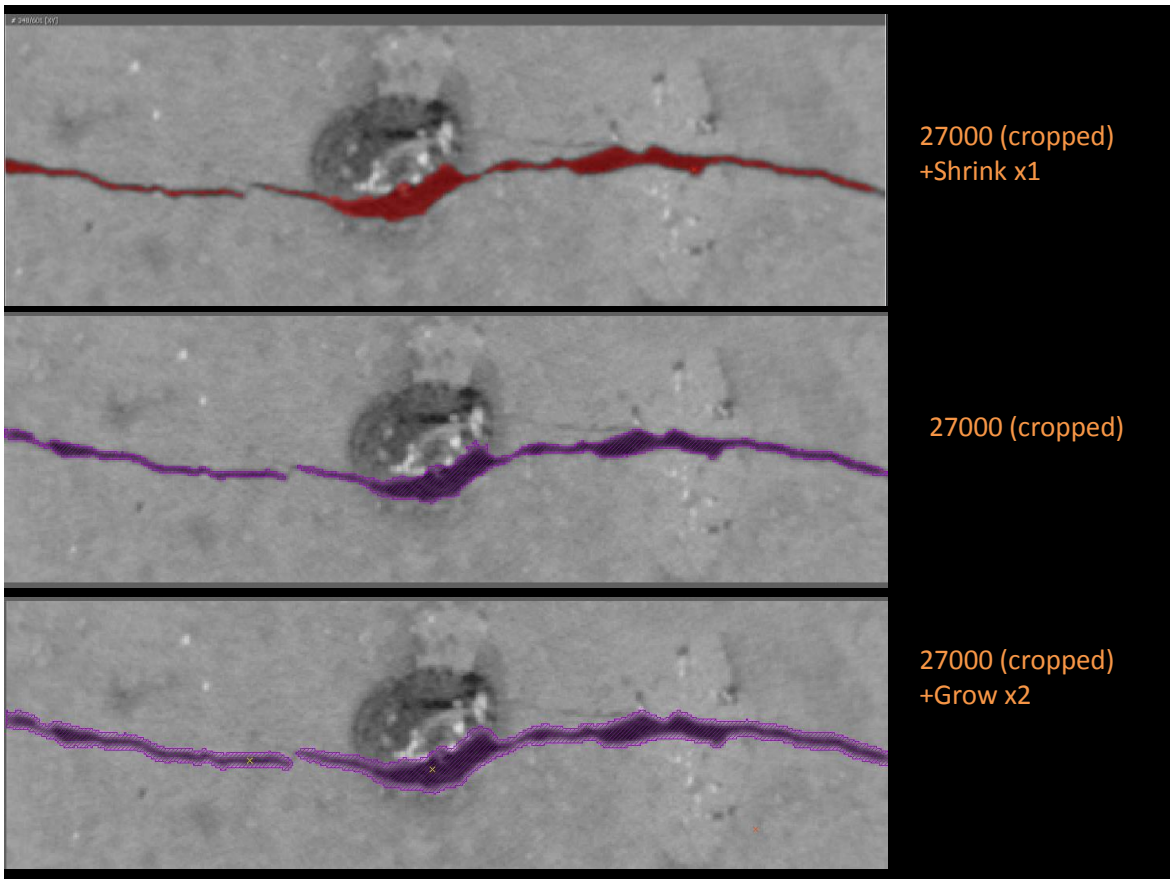
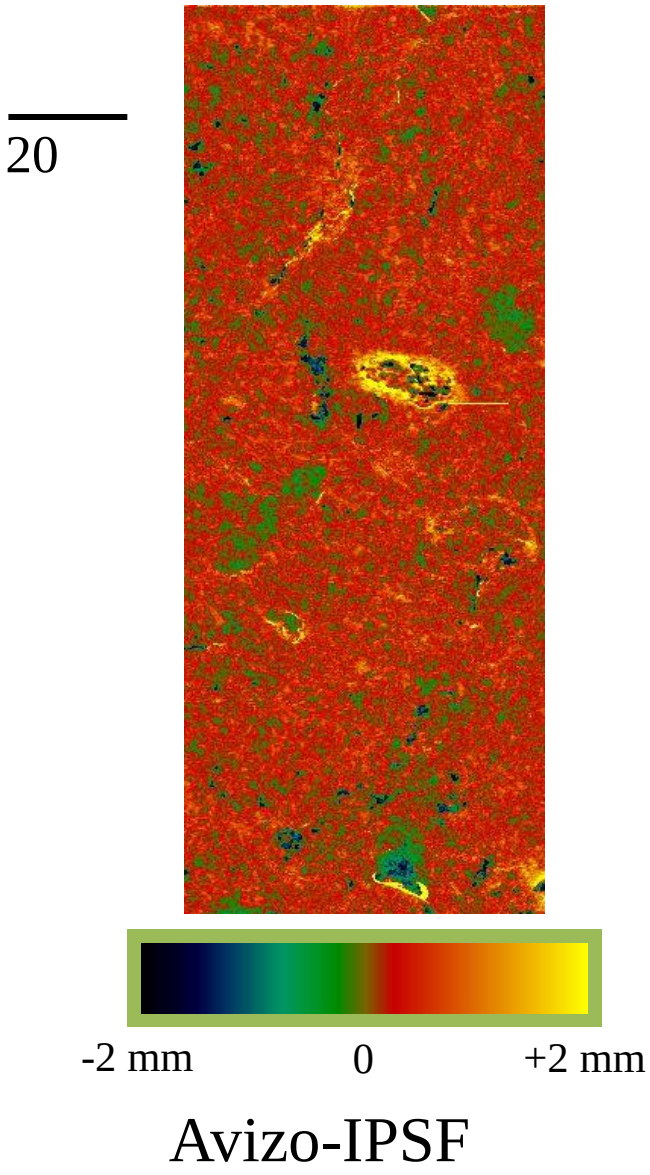


Figure 11 depicts a strip down the center of the fracture sheet, and the values represent a simple subtraction of the apertures measured at each point by the FWHM minus the same points measured by the IPSF method. Hotter colors represent larger measured apertures from Avizo, and cooler colors signify areas where the Avizo estimate is lower than the IPSF apertures. The median aperture given by the IPSF method was 0.689 mm, whereas the Avizo method gave a median aperture of 0.757 mm. The median

difference between the two was the Avizo overestimating by 0.301 mm. This is a difference of 43.7% - certainly nontrivial.

Figure 11: A two-dimensional view of the difference of the Avizo aperture estimate (for a CT number threshold of 27,000) minus the IPSF measurement. Hotter areas, which cover the majority of the map, represent overestimation by the thresholding.



It has been shown that the PSF itself is a more realistic way to think about the underlying physical reality of CT data, and the IPSF method generates smaller errors when it encounters artifacts and heterogeneous materials. From this foundation, confident that this iterative method is a sound approach that will account for local variabilities, the 3-D network tracing application implemented in Blob3D will now be described.

Chapter 4: 3-D Fracture Network Tracing in Blob3D

4.1: BLOB3D AND OBJECT-ORIENTED PROGRAMMING

Blob3D was developed to process large 3-D data sets generated by X-ray CT scans (Ketcham, 2005a). The main purpose of the program has three steps: to identify, separate, and measure any number of individual objects within a data volume (known as "components"). The components are collections of voxels that represent a material within the scan volume, such as void spaces or particular minerals. The objective is to determine which voxels in the entire volume belong to each material (component). Then, each component is separated into individual sets of contiguous voxels, or "blobs." After every component's blobs have been identified, they are analyzed for information. The three following processes: Segment into components, Separate components into blobs, and Extract the data for each blob, are the three modules within the Blob3D program. For more information about these capabilities, and Blob3D in general, the reader is referred to Ketcham (2005a).

These three modules are not used, though, in the fracture tracing. Instead, the fracture tracing is operated through the Inspect Viewer interface of Blob3D. The first time the data, typically in the form of TIFF files in a single directory, are loaded into Blob3D, the user is prompted for the inter-slice spacing and inter-pixel spacing (in millimeters), which are the height of the voxels in the z-direction and the length/width of the voxels in the x-y direction, respectively. The documentation provided by the scanning service should contain the voxel dimension information. After entering the unit voxel size, the (Inspect) button is enabled. Clicking the Inspect button launches the

Inspect Viewer (shown in Figure 4).

There are many different operations the user can perform from this window to simply check out the data, without using any of the three Blob3D modules. The File menu has an export image option; the View menu permits the user to reset scale bars and to specify a custom coordinate system. The Statistics menu reports the accumulation statistics from the PSF tool, as previously described. On the left side of the interface are several tools for looking at the volume data. The data can be viewed from the ortho- x, y, or z plane. A slider bar is provided to toggle through every TIFF in the dataset. The display levels and zoom can be adjusted. Some ROI tools are provided to get specific information about a particular point (coordinates, location, and value), line (length in millimeters), or area (number of constituent voxels, minimum CT number, maximum CT number, mean CT number, and standard deviation). The bottom section of the interface contains some features for blobs that will not be required for the fracture tracing.

The Line ROI tool doubles as the PSF-measuring device described in Ketcham and Hildebrandt (2014). Because the PSF is a key parameter in the fracture tracing algorithm, this is a very important tool. The mechanics of its use are described in the section The PSF Tool. The mean CT values given by the area ROI tools (either rectangle or oval) are also key inputs for the tracing algorithm. It is important to obtain a mean value for the air, which is easily doable on the outside of a sample if the interior fractures are too thin and contaminated by blurring. Currently, the fracture tracing is also built into this Line ROI interface. Ideally, the fracture tracing would be able to perform separate from Blob3D, to reduce storage and user interface overhead from the other Blob3D

program modules, though it could still keep the same Inspect Viewer interface, as there is a stand-alone version of the Inspect Viewer into which the procedures could be inserted.

4.2: THE OBJECT-ORIENTED PARADIGM

Blob3D was compiled from code written in Interactive Data Language (IDL), proprietary software owned now by Exelis Visual Information Solutions. It is a vectorized language with an emphasis on processing large amounts of data. IDL supports object-oriented programming (OOP), and this paradigm is the organizational scheme for Blob3D. OOP is a method of writing code in which data structures called "objects" interact with each other to accomplish a task; there are six main principles that characterize OOP (Clark, 2013). These concepts are worth explaining now as they will assist with contextualizing the programming decisions made in the production of the tracing algorithm.

First, the OOP paradigm maintains that the world is made of objects, all of which have both fields of data and operated procedures, also called methods, for working with that data. For example, a mineral object would have certain inherent attributes such as hardness, crystalline habit, color, and so on. Outside influences might stress the material to cause the mineral to fracture. The procedure of fracturing would be akin to a method for the mineral object. The fracture tracing algorithm is comprised of many objects. A single voxel traverse across a fracture plane, and all the information associated with that one traverse, is represented by an object named B3D_Tracer.

Abstraction and encapsulation are another two characteristics of OOP. Abstraction means that every possible attribute is not represented by the object, thus

helping make necessary simplifications. Encapsulation means that the attributes that are present are hidden, and the user or programmer has to interact with or ask the object for information. The fourth characteristic of OOP is polymorphism. Polymorphism refers to the fact that different objects can respond to the same function call, but each object will implement its own version of that procedure; a print call to a B3D_Tracer will print out its length and midpoint, whereas a print call to a B3D_Tile will print out its location. A B3D_Tile is a collection of nine B3D_Tracers, which is an example of the characteristic that object also can be aggregated – meaning one object can consist of a multitude of other objects. A fracture network, for another example, is comprised of many smaller fracture sheets.

Finally, objects are classified by hierarchies, a trait which is called object inheritance. Commonly referred to as the "is-a" relationship, inheritance means that some classes inherit the properties of overarching classes much like a sandstone "is-a" sedimentary rock, and thus has the properties that all sedimentary rocks share, but with unique encapsulated properties. In summary, the philosophy of the object-oriented programmer is that the world is made of hierarchical objects that can be abstracted, encapsulated, aggregated, and queried in similar ways (Clark 2013). The OOP philosophy guided the original construction of Blob3D as well as how the tracing algorithm described by this paper operates with the B3D_Tracer and B3D_Tile object classes.

4.3: AN OBJECT MODEL FOR FRACTURE TRACING

A fracture network, as outlined in Chapter 1, is simply a set of connected

fractures. Each individual fracture that is part of the network has its own set of attributes. These two observations mesh with the preceding discussion of object-oriented programming. Fractures can be abstracted in the context of the object-oriented hierarchy to aggregate and characterize the overall topology in a meaningful way. Connectivity information helps define the network's graph, and the other attributes of each portion of the network are neatly encapsulated.

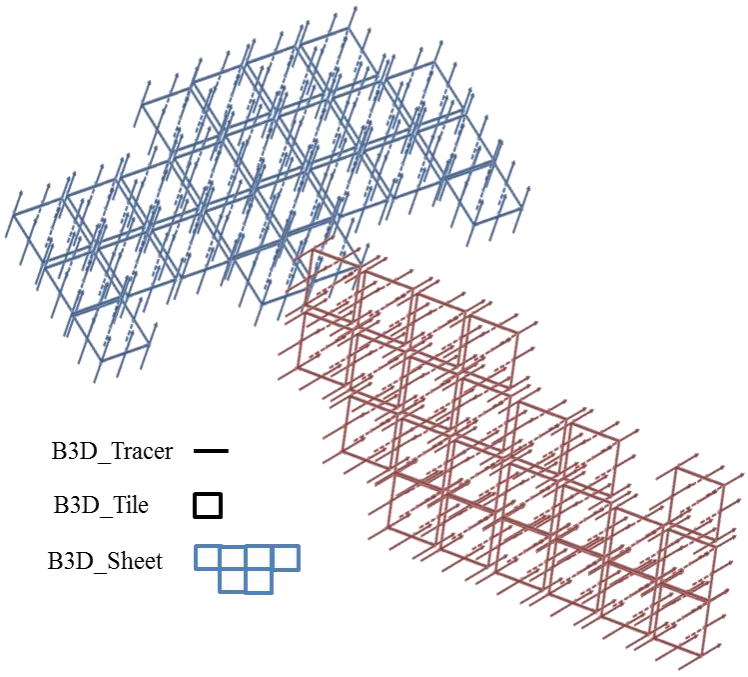
The most basic unit of this fracture tracing algorithm, that is, the smallest object, is a single traverse across the fracture plane. The object class B3D_Tracer contains all the information associated with one traverse. (The class B3D_Traverse was already a reserved class name from the PSF tool developed in Ketcham and Hildebrandt (2014). The B3D_Traverse class describes a traverse of voxels across a single boundary, whereas the B3D_Tracer is slightly more complicated in that it defines a voxel traverse across the width of an aperture – two boundaries. The information in a B3D_Tracer includes the main array of voxels that comprise the traverse as well as calculated information such as the traverse's length and aperture midpoint. Additionally, the B3D_Tracer class contains subroutines to help figure out where to go to trace the fracture.

The next largest unit defined is a "tile" of traverses. A tile is a data structure with information pointing to a single traverse in addition to eight surrounding parallel traverses (as described for the PSF tool). This tile of traverses, a B3D_Tile object, has information pertaining to the best-fit aperture midpoint plane, defined by a normal vector calculated amongst the calculated fracture midpoints of all the B3D_Tracers that comprise the tile. Additionally, a tile contains calculated metrics, further described in

Section 4.4.2, to determine whether it is at a termination point or not.

Further, a collection of tiles constitutes a fracture "sheet," for which the object class B3D_Sheet is designated. A sheet is a searchable list of tiles designated to be connected. If the sheet encounters a bifurcation in the network, the sheet is terminated at that point and two new sheets are generated at the point of bifurcation. Collectively, the set of fracture sheets define the entire fracture network. A simple schematic using the object framework of a fracture network containing a single bifurcation is illustrated in Figure 12.

Figure 12: The object model for fracture tracing describes a fracture network as being comprised of "sheets" of fractures. A sheet is built of "tiles," which are sets of nine traverses, or "tracers." Pictured is a schematic for a small fracture network consisting of two intersecting B3D_Sheets, colored red and blue. The sheets each are composed of the square B3D_Tiles, which are defined by a set of nine B3D_Tracers, each represented by an arrow.



This hierarchical system of tracers, tiles, and sheets is a new way of thinking about fractures at the quantitative level. The fracture tracing algorithm currently features the tracers and tiles and a partial implementation of sheets. Currently, sheet connectivity information is confined to a single fracture plane, but future implementations will fully extend sheets into the entire 3-D network.

4.4: FRACTURE TRACING METHOD

This section provides a detailed elaboration on the steps taken by the program to characterize the fracture network, beginning with the first user inputs and ending with a product of fracture data to export. A pseudocode outline version is available in the Appendix and the full program project folder is available on request. The user inputs are minimal, consisting only of four elements, apart from the volume data itself:

1) The CT number for the feature being traced – usually air, in the case of a fracture
2) The highest threshold value for the feature being traced – usually this will be the halfway point between air and the surrounding matrix (the FWHM criterion)
3) The PSF for the dataset
4) A line drawn across the fracture plane

From these four inputs, the program traces a fracture to its endpoint, defined as a bifurcation or termination. Future implementations will follow bifurcations to create sheets that will define the entire fracture network connected to the input feature the user's

line was drawn across. The database of points describing the top and bottom fracture surfaces, the fracture midpoints and apertures, and metrics describing the quality of the data are all outputted to .csv files. A MATLAB code was written to analyze and cross-plot these outputs against each other. Example results are shown in Section 4.6.

4.4.1: User procedure

As with any image processing project, the first step is to load the images. The preferred image type is 8-bit TIFF files. TIFF files are preferred to JPEG files because JPEGs use a lossy compression format that alters the data in an irrecoverable way that may impact accurate fracture measurements. Smaller 8-bit files are preferred because the datasets are already large enough without them being composed of 16-bit data. While the functionality is the same, the processing takes much longer. Most CT scanners produce 16-bit images, and the bits lost when converting to 8-bit images typically are not significant. On opening Blob3D, the user is prompted to Load TIFFs from the relevant directory. The user is prompted for the voxel dimensions, which are the inter-slice and inter-pixel spacing parameters from the scan. Should the two dimensions be different, the PSF tool corrects for the anisotropy (Ketcham and Hildebrandt, 2014).

After loading the data, the Inspect button becomes active. Clicking this opens the Inspect Viewer. It is imperative to scan through the slices and check for large or prevalent artifacts. While the IPSF method is forgiving of some artifacts, there are limitations when it comes to very poor data. After examining the data, the user determines the user inputs for the tracing algorithm. First, the ROI area tool is used to select relatively homogeneous regions of air on the image and determine a mean CT

value for air. It is advisable to sample several areas and come up with an average value to not introduce bias from what might be a patchy spot in the data. The IPSF method for calculating aperture has flexibility in its iterations, so a small difference in the CT number used for air will not matter so much, but it is still important to be as accurate as possible when determining the inputs. Similarly, the user performs the same ROI selection operation on a matrix material with the closest CT value to air (the darker material) to find a mean value. The purpose of deciding a mean CT value for air and a mean CT value for the matrix is to determine a threshold value that will demarcate air from everything else in the volume: that is, values that indicate that there is no component of air in a given matrix voxel, and vice versa. When the algorithm encounters a voxel at the CT value specified to be completely within the matrix, it is confident that the voxel is outside the region of the data influenced by the fracture because the fraction of blurred air is zero in the voxel, meaning no partial-volume effects.

Next, the user must determine a PSF value for the volume. An accurate PSF reading is very important for generating accurate information about the fracture network. The PSF can vary throughout the volume, as previously explained, but taking several reasonable measurements will suffice. The PSF can be determined to within one voxel from only six measurements (Ketcham and Hildebrandt, 2014). For determining the sample PSF, the line should be drawn across a flat boundary between two materials, orthogonal to the contact. When a line is drawn with the ROI tool, the Line Viewer window will pop up (Figure 4). The graph display plots the CT value of each voxel in the traverse over the voxel number. In the drop-down menu at the lower right, selecting

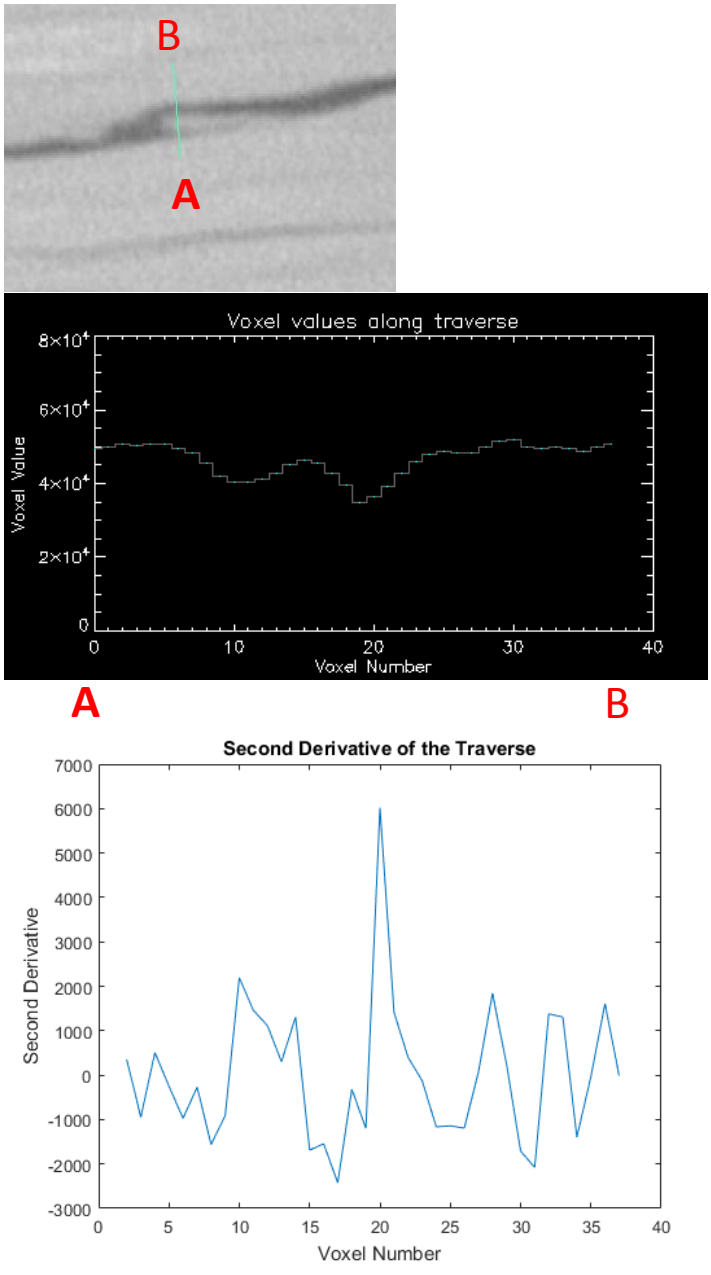
the “Measure PSF” option will calculate the PSF for that single line, called the “Single Raw PSF.” Then, the program calculates the “Mean Raw PSF” for that line and eight parallel traverses of equal length (same as a B3D_Tile). The geometric corrections are applied and an adjusted mean PSF radius is reported. If the PSF model (the red line) fits the real data (the white stair steps) reasonably well, with no anomalies from other materials or artifacts, the user should accumulate the PSF. The “Mean Accumulated PSF” and “Cumulative Error” is continuously updated as the user accumulates more PSF measurements. After about six traverses are accumulated, the user can note this mean accumulated PSF as the third input value. With the required three numerical inputs in hand the process is ready to begin. The “Trace Fracture Network” option is selected from the dialogue box, and the CT air value, air threshold value, and PSF should be entered. The program then enters the tracing algorithm subroutine.

4.4.2: Tracing algorithm walkthrough

First, initiating a new fracture sheet, a tile of nine traverses is created, centered on the user-drawn line. Each traverse initialization calculates the boundaries and midpoint of the fracture aperture along that particular traverse. A flag, denoting which traverses were successful or not, is registered. At this stage, a traverse will typically fail if the IPSF method does not converge to a solution, meaning a fracture is not present or too small for the detection limit. The best-fit plane of the midpoints is calculated from the successful traverses; points outside a 95% confidence interval, defined as 1.96*standard deviation +/- the mean, are ignored. The normal vector representing that plane is stored in the tile object. The traverse is then checked for bifurcations, using the second

derivative of the voxel traverse data. If a substantial change in the second derivative, as depicted in Figure 13, is found between the walls of the aperture, a bifurcation is likely present because the traverse was likely crossing from matrix to air, then air to matrix, then matrix back to air, then finally air back to matrix. The inflection points in these material switches are reflected in a dynamic second derivative. In the event of a bifurcation, a new sheet is initiated for each branch; if not, the next tile is constructed.

Figure 13: Example of a fracture bifurcation. The traverse (top) from A to B contains a finger of the lighter matrix material, reflected in the CT values (middle). The plot of the traverse's second derivative (bottom) shows a major swing within the walls of the fracture, from a concave up feature at voxel 10 to a concave down feature at voxel 15, then back to a concave up feature at voxel 20.



The location of the next constructed tile is based on the previous tile's midpoint and best-fit plane orientation. For a given step size, the next midpoint is projected to be

(4) $(\text{x-coordinate}_{previous} + \text{StepSize}, \text{y-coordinate}_{previous} + \text{StepSize} * \text{y-coordinate}_{xp})$,

where y-coordinate_{xp} is the y value in the cross-product of the best-fit plane and the vector [0,0,1]. This cross product produces a line parallel to the best-fit plane of the tile, thus projecting the fracture's orientation forward along the fracture. The new start and end points are projected in a similar manner. After all of the tiles have been created on the slice in the positive x-direction from the original user-drawn traverse, the process is repeated from in the negative x-direction, thus traversing the entire fracture on the slice. After the first slice, any tile with a valid traverse leading to the next slice is a candidate for extending the sheet. New tiles are created following the curvature of the fracture in 3-D. The entire database of sheets, with their composite tiles, is then available for export. The database includes information such as the set of points defining the tops and bottoms of the apertures, as well as the midpoints. The orientation vectors for each tile and the aperture width are also available.

Three metrics to help demonstrate that a tile contains a termination point for the fracture being traced are also associated with the sheet. The first is a sum of squared distances calculation of the tile midpoints to the best-fit plane of the midpoints of that tile. The sum of squared midpoint distances of the tile containing the original user-drawn traverse is compared to that of the current tile. The first traverse is used as an estimate of the amount of variation that should be encountered as one traces the fracture, and a sharp rise in variation signals an anomaly, whether it is a termination, a bifurcation, or an

algorithm failure. If this ratio, termed the "reference departure," is greater than 3, the tile is determined to be a termination point because the midpoints are expected to remain similarly close to the best-fit-plane. If multiple traverses have anomalously distant midpoints, it is likely that the fracture is no longer present. The second metric is the relative deviation of a tile's apertures among the nine component traverses: the standard deviation of the apertures divided by the mean of the apertures. If this number is greater than 0.3, meaning that the apertures within a single tile vary by more than 30%, the fracture is considered to be terminated at this point. The third metric compares the angle between the calculated fracture plane, meaning the best-fit plane of the aperture midpoints, and the viewing plane. Generally, the fracture will be roughly perpendicular to the viewing plane. If the angle between these two planes registers more than 45 degrees in either direction of perpendicular, the fracture is considered to be terminated. It is unlikely that the fracture is going to helix so abruptly, an additional complication slated to be addressed in future iterations of the program, and in the CT data this behavior is indicative of a voxel traverse that does not contain a fracture – a traverse simply within the matrix.

The expert reader will observe that a fracture network could actually spider out, then stretch back to the same slice, either above or below the area that was just mapped. Fractures can have many complex variations and paths, though time constraints have limited progress for this very difficult challenge. Currently, the program is adept for tracing more planar than chaotic behavior. An improved structure for keeping track of continuity would help address this issue.

4.5: PACKSADDLE SCHIST TEST DATA

A sample of Packsaddle Schist was selected for testing the algorithm because of its large number of thin fractures and complex, anastomosing bifurcations. Reese et al. (2000) describe the Packsaddle schist unit in the field as being comprised of folded and deformed tonalites, gabbros, and granodiorites that formed in different tectonic environments, as it is a laterally extensive unit. Age dates were acquired for some samples in the Llano uplift to constrain the regional geology. The Packsaddle metavolcanic unit was dated at about 1.2 billion years old. The authors determine that the Packsaddle is younger than the Valley Spring Gneiss, which contradicted previous findings. The contact between the two units is a ductile thrust zone.

The Packsaddle schist sample, collected for a study on fracture surface roughness (Slottke, 2010), was scanned at the University of Texas High-Resolution CT Facility (UTCT), the scanners of which are described in detail in Ketcham and Carlson (2001). During this project, though, the scanners were updated. Instead of a collimated single-slice acquisition used previously by Slottke (2010), the new Packsaddle scan was performed using a cone-beam acquisition with a Perkin Elmer [2048x2048] XRD 1620/1621 AM/AN detector. Four brass filters were used in the scan, performed at 450 kV. The distance from the source to the object was 526 mm. No ring correction was applied, but data from some detectors were removed prior to reconstruction because they registered defective pixels. Reconstructed voxel size was 0.0974 mm. Voxels are always cubic in cone-beam scans.

Following the input procedure described in Section 4.4.1, the TIFF files of a

subset of the rock were imported into Blob3D, after converting them to an 8-bit data format in ImageJ. The rectangular ROI tool was used to sample regions on the image that were air to arrive at a mean CT value approximation of 88 for air on this particular set of images. The same process was performed to arrive at a CT number of about 210 for the matrix, meaning that the FWHM criterion input was 149. The (geometrically corrected) PSF was measured and accumulated for ten different locations, listed in Table 1, yielding a mean accumulated value of 7.35, with a standard deviation of 0.48. A PSF of 7.35 means that any fracture thinner than 7.35 voxels (equivalent to 0.715 mm for this scan) will be affected by blurring and thus requires a grayscale-aware measurement method like the IPSF to measure it accurately.

Table 1. Ten PSF samples taken on the Packsaddle schist sample image

Traverse	1	2	3	4	5	6	7	8	9	10
PSF	7.94	7.38	7.74	7.00	7.33	6.33	7.60	6.96	7.79	7.40
Std. Dev.	1.09	0.90	0.98	0.74	0.87	0.64	0.23	0.62	0.71	0.81

With these requisite input values available, the fracture network is ready to be traversed. The entirety of the automated fracture network tracing algorithm is not completed. However, it is still possible to collect 3-D data on fractures in a sample with some extra manual effort.

The Packsaddle Schist volume was cropped to 25 interior slices. Two adjacent slices from the middle of the cropped dataset, numbered 12 and 13 – seen in the

Appendix, were selected, and traverses were drawn across the possible fractures in the images. As the tracing algorithm creates tiles from the two adjacent slices in addition to the current slice, this means that a total of four slices were analyzed. With a voxel width of 0.0974 mm, the total thickness covered in this simple test was 0.3896 mm.

Six different tile measurements were exported into .csv files:

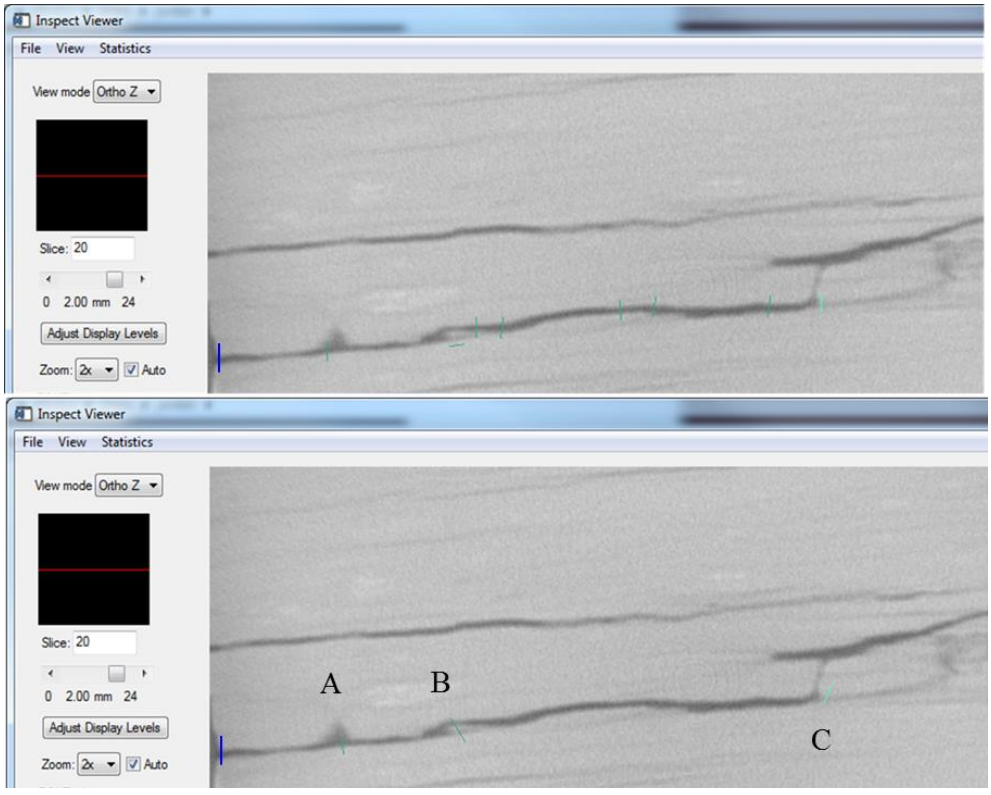
1. Mean aperture of the nine tile traverses
2. The fracture midpoint of the central traverse within the tile
3. The normal vector describing the orientation of the best-fit plane to all the tile's fracture midpoints
4. The angle between the fracture's orientation and the viewing plane
5. The relative deviation of the tile's apertures
6. The sum of squared distances ratio between the current traverse and the initial traverse

For slice 12, a total of 572 tiles were created. The mean of the apertures for all tiles is 5.9999 voxels, the mean relative deviation is 0.0955, and the reference departure is 1.1288. For slice 13, a total of 507 tiles were created. The mean of the apertures for all tiles is 5.7988 voxels, the mean relative deviation is 0.0903, and the reference departure is 1.4553.

Additionally, the same tile information was gathered on a single segment of a fracture, further described in Section 4.6, under two different reference departure failure conditions: one for a ratio of greater than 3 and one for a ratio greater than 6 (Figure 14). The purpose of this test is to see the effect of the failure conditions on the resulting tile

data. When the failure condition was at a ratio of 3, the mean of the apertures for all traverses is 5.7447 voxels, the mean relative deviation is 0.0699, and the mean reference departure is 1.5998. These data are aggregate from a total of 97 tiles. When the failure condition was at a ratio of 6, the mean of the apertures for all traverses is 7.3464 voxels, the mean relative deviation is 0.0547, and the mean reference departure is 1.3838. These data are aggregate from a total of 103 tiles.

Figure 14: Two trials of tracing the long, lower fracture feature. The tracing began at the blue line. The traverses are marked in green where the tracing algorithm encountered a failure condition. The top image was created with a reference departure of 3 as the failure condition. The bottom image used a ratio of 6 instead. Point A signifies a failure due to a 3-D pore very large relative to the fracture aperture, Point B signifies a failure due to a bifurcation, and Point C signifies a failure due to termination.



4.6: DISCUSSION

Overall, the objective of this project was to create a tool to generate 3-D data for real fractures. Figures 15-19 demonstrate a small sample of the possible relationships that can be observed with this tool. A primary focus is how the fracture aperture relates to other parameters of the fracture. Figure 15 shows the relative deviation of a tile, calculated by the standard deviation of the apertures divided by the mean of the apertures, plotted against the mean aperture. The chart shows that generally, the smaller the aperture in voxels, the greater the aperture relative deviation. The results make sense because a higher relative deviation is expected for smaller apertures: thinner fractures are more difficult to measure accurately, as they are smaller than the PSF and more prone to blurring. It is worth noting the mean relative deviation is less than 10%, which is excellent for fractures that on average have widths less than the PSF, further confirming the reliability of the IPSF model to calculate fracture apertures.

Figure 15: Tile data for the aperture relative deviation versus mean aperture.

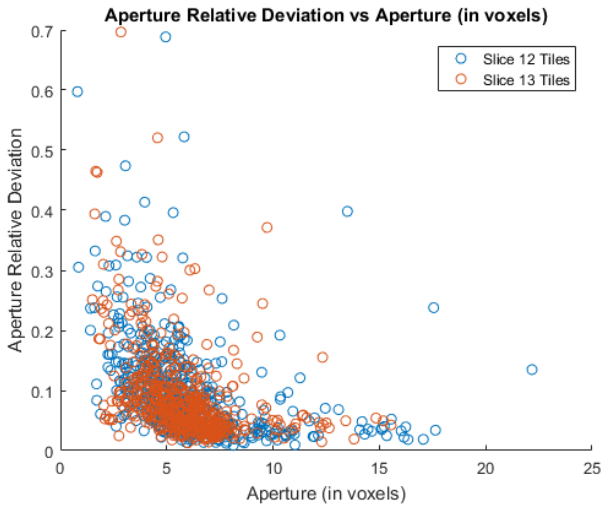


Figure 16: Tile data for the reference departure versus mean aperture.

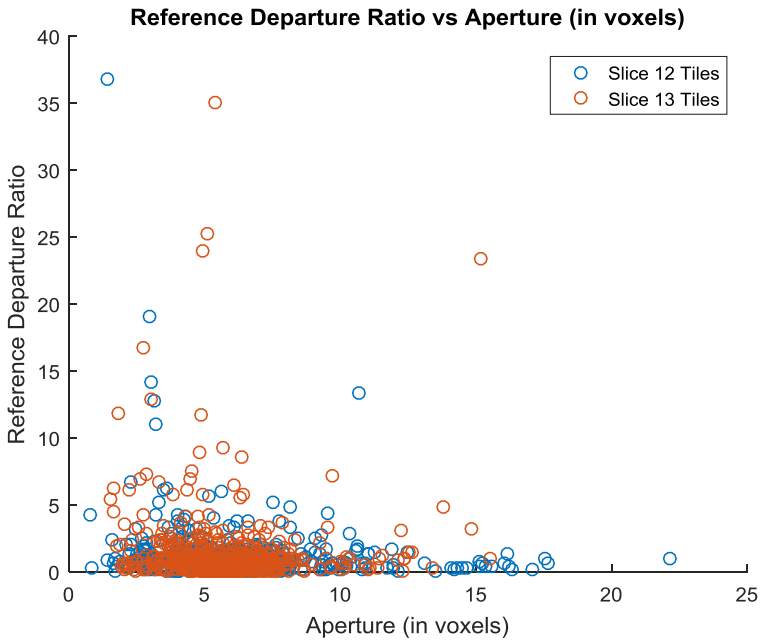
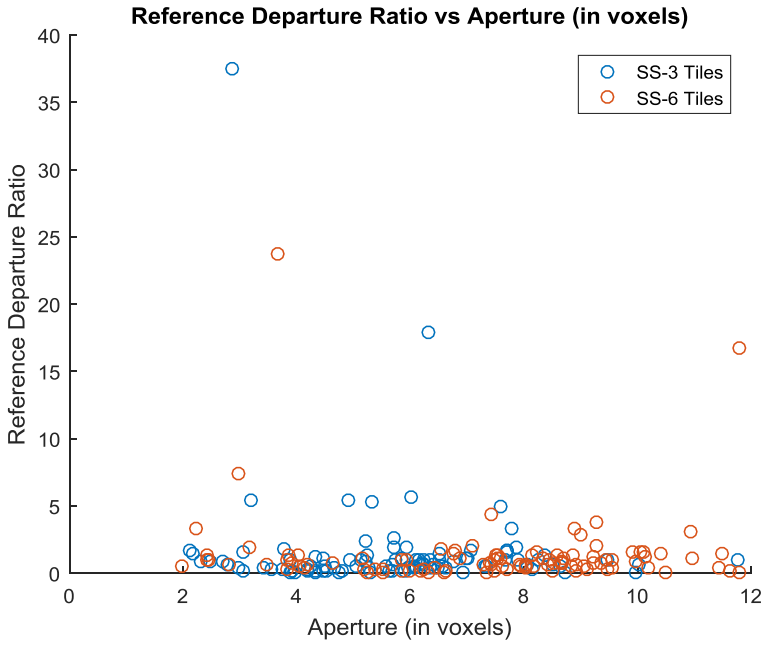


Figure 16 shows how the reference departure is not particularly affected by aperture, which is good because this is a metric used to determine if a fracture is diverging or terminating. If there was a relationship between aperture and the reference departure, a fracture traverse could be cancelled solely based on aperture size, not based on the absence of an actual fracture aperture. Overall, there is a considerable amount of variation, and because not all valid tiles will have a ratio smaller than three, some fracture tracings were cancelled prematurely. Using a higher value is likely warranted to encompass more valid tiles.

To begin to examine some of the effects of changing the reference departure, a single fracture, starting at the same location, was traced under two different reference departure failure conditions: first, the typical value of three used for the rest of the data,

and second, a value doubled to six. The fracture halted eight times on the first condition and three times on the second. When the tracing halted, the traversing was re-initialized on a line drawn adjacent to the point of failure. Only one halting, on the first run, was not due to the reference departure metric, but rather the relative deviation instead. The locations of these stops are shown in Figure 14. The greater failure condition yielded a higher aperture estimate (Figure 17). Intuitively this makes sense because higher variance permitted with the midpoints will result in greater variance in the mean aperture of all nine traverses, so it is likely the aperture estimate will increase, especially when the fracture width is less than the PSF. However, the value of 6 used is likely an overcompensation.

Figure 17: Tile data for the reference departure versus mean aperture for the experiment in changing the ratio restrictions. The blue points are for the run at a ratio of 3, the red for a ratio of 6.



4.6.1: Comparing failure conditions

Figures 18-20 plot the failure metrics against each other to see if there are any trends between them. If a clear relationship existed, it would be possible to eliminate one metric as a failure condition to streamline the program's computational operations. The colored lines across the plots mark the boundaries of failure conditions for each metric. In Figure 18, eight cases failed the angle to viewing plane check. Of those eight, only three were also above the relative deviation case. Similarly, in Figure 19, only two were above the reference departure check, so it is fair to say that skewed orientations are still worth looking for. Tiles with unusually high angles relative to the viewing plane are not always getting recognized by the other two failure cases. Further work will be necessary to optimize the tile's failure conditions.

Figure 18: Tile data for the aperture relative deviation versus the angle of the best-fit midpoint plane to the viewing plane. The lines across the plot indicate the failure boundary conditions (greater than 0.3 relative deviation, angles of less than 45 degrees, and angles greater than 135 degrees).

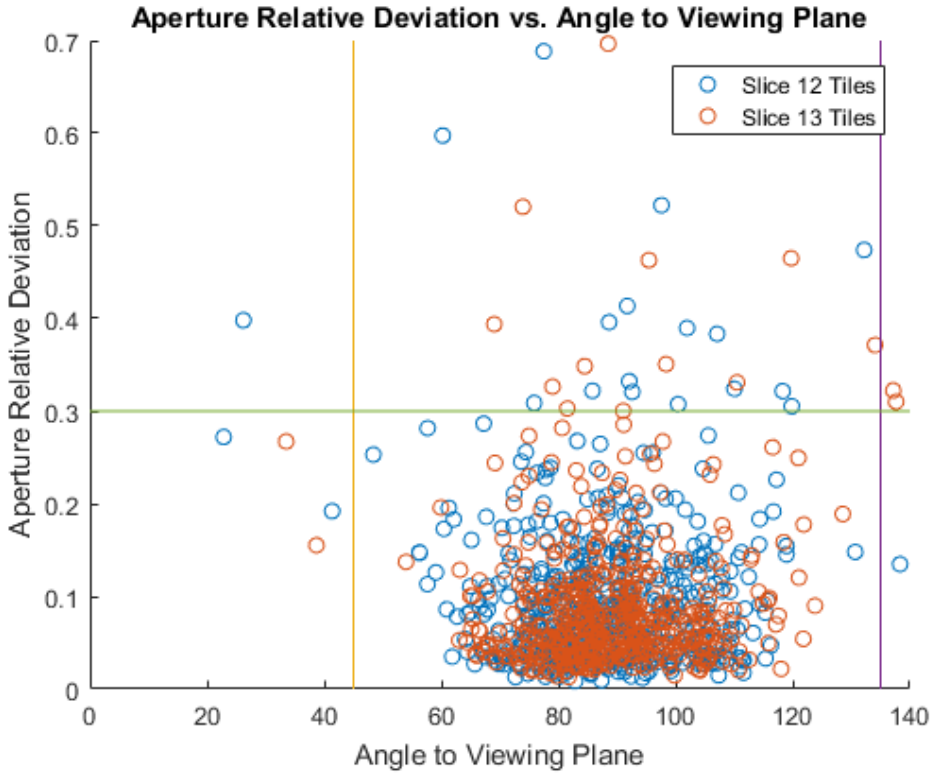


Figure 19: Tile data for the reference departure versus the angle of the best-fit midpoint plane to the viewing plane. The lines across the plot indicate the failure boundary conditions (departure ratios greater than 3, angles of less than 45 degrees, and angles greater than 135 degrees).

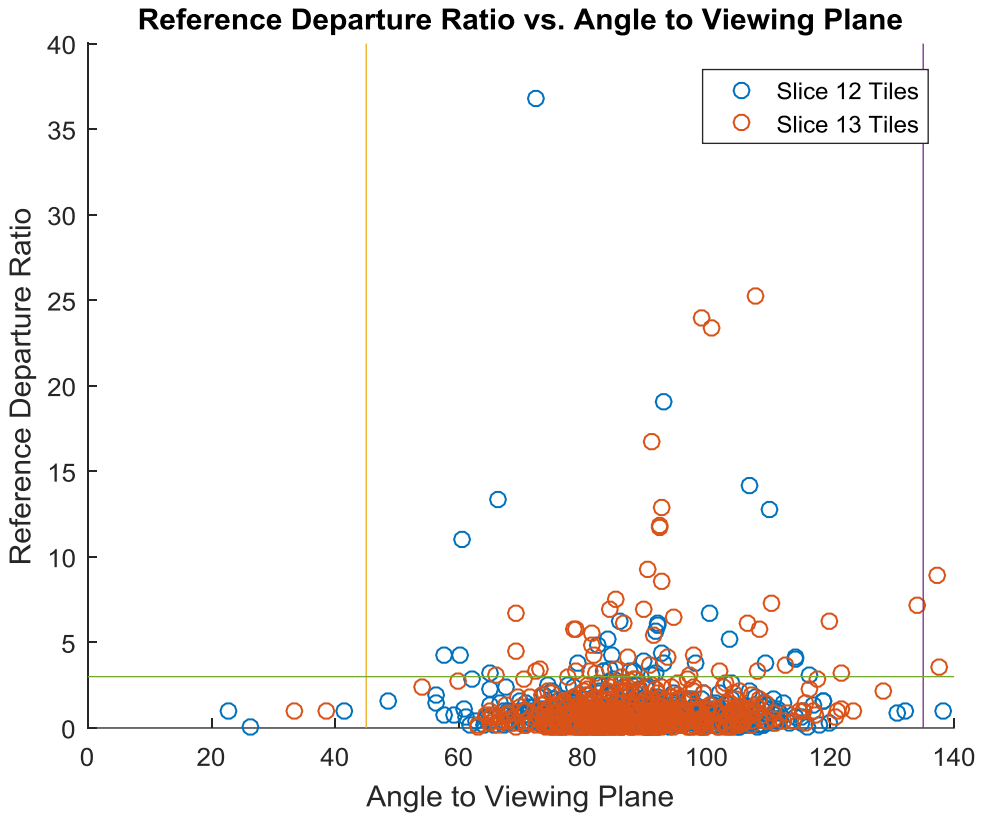


Figure 20: Tile data for the reference departure versus mean aperture. The lines across the plot indicate the failure boundary conditions (departure ratios greater than 3 and relative deviation greater than 0.3.

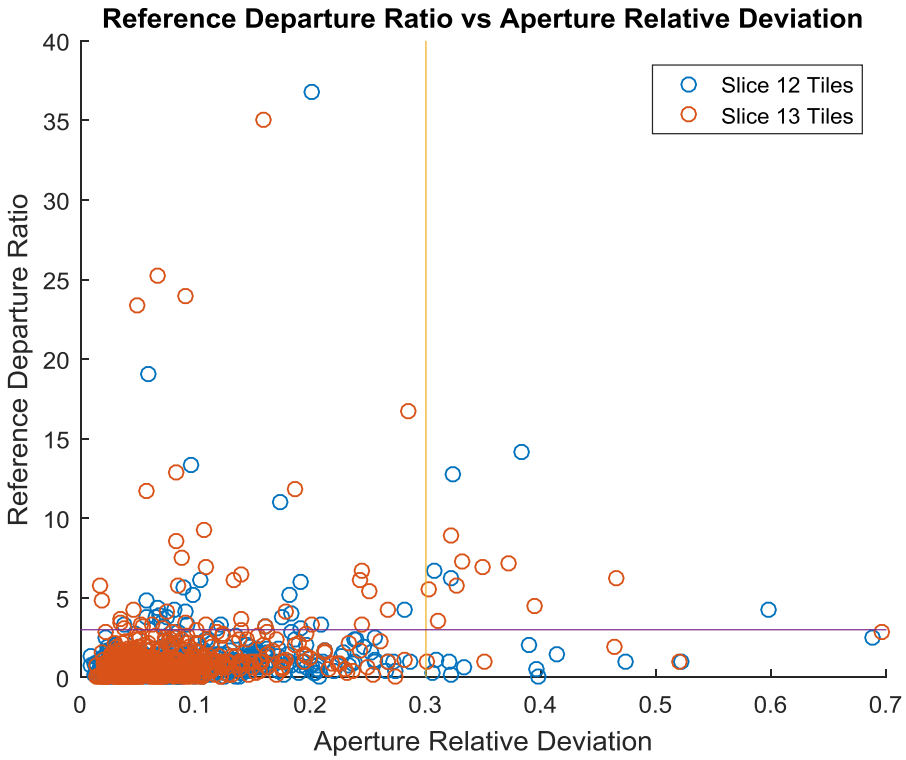


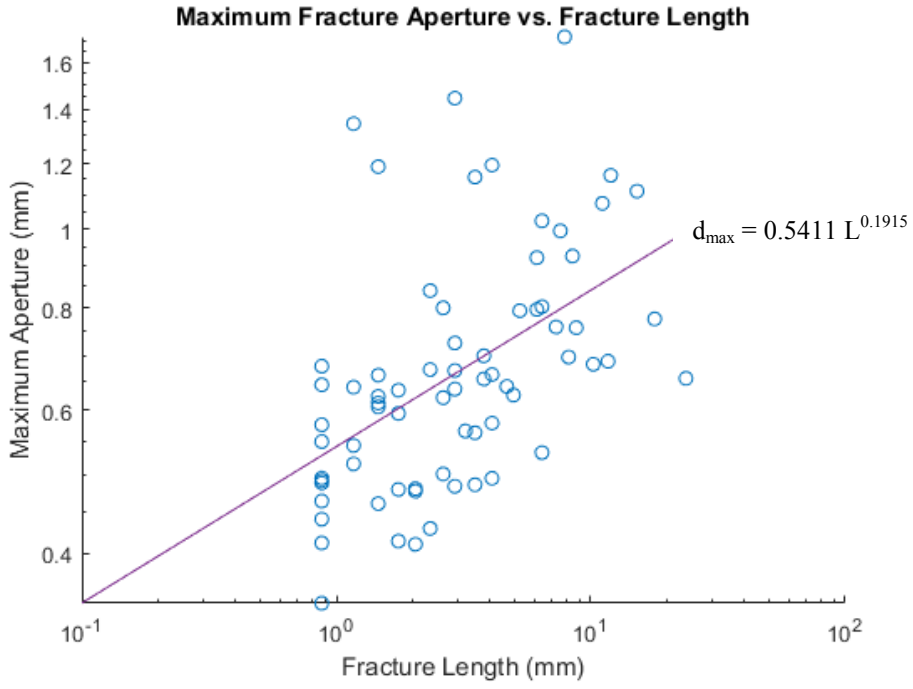
Figure 20 depicts the reference departure against the aperture relative deviation. As only eight tiles were failed due to angular issues, it is clear that among the three, most problems with traversing occurred due to the reference departure – and with no clear correlation to aperture relative deviation. Overall, this figure points to examining the reference departure as a source for some of the problems the algorithm has in traversing an entire network. The Packsaddle schist itself has very thin fractures relative to the data resolution of the scan, which makes it even more challenging. Much of the hairline fractures are simply too small relative to the PSF to trace them out. The outliers in the data are typically from attempting to start a trace on one of these hairline features. The

function used to find the aperture is unable to converge on several of the traverses within a tile across one of these features, and the function outputs highly variant midpoint locations, even if the aperture guess might be similar – likely caused by similar measurements on the average variance of the matrix itself.

4.6.2: Fracture length vs maximum aperture

Another new calculation possibility for 3-D fracture network analysis is comparing mean fracture aperture and the fracture length. The trend shown with these measurements from the CT scan can then be compared to data of fracture aperture measurements taken from the field. Figure 21 shows a roughly linear relationship between fracture aperture and length, corroborating with findings from Olson (2003).

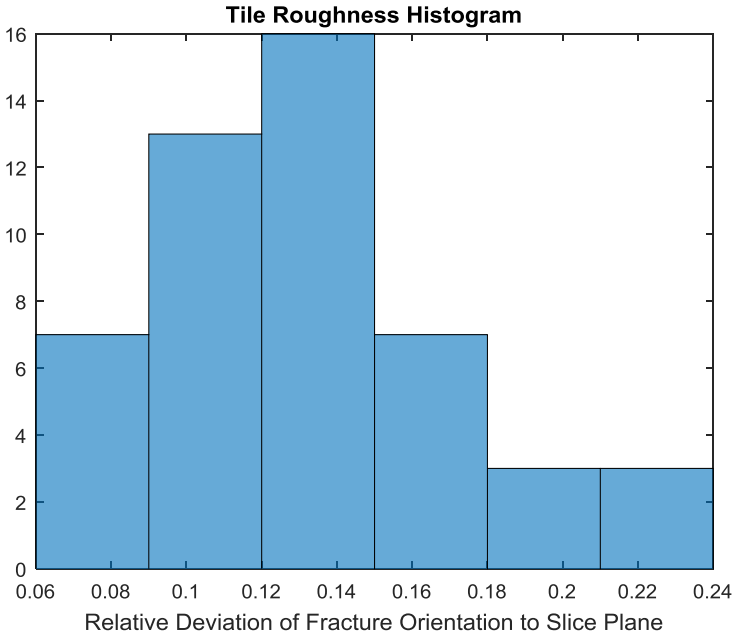
Figure 21: For sheets with at least three tiles, the maximum fracture aperture is plotted, on a log-log plot, against the total fracture length, calculated as the number of tiles multiplied by the distance between tiles.



4.6.3: Roughness in 3-D

A potential new way the data from this method can be used to quantify roughness in 3D is to take the relative deviation of tile angles to the slice plane: the standard deviation of the fracture orientation divided by the mean angle of orientation. For all sheets in the dataset with six or more tiles, this quantity was calculated and plotted in Figure 22. This method, with its high degree of accuracy in determining fracture orientations and apertures, enables new possibilities for modeling influenced by roughness with underlying real fracture data on the small-scale level of tiles.

Figure 22: For the 49 sheets sampled of more than six tiles, the standard deviation of the fracture orientation was divided by the mean angle of orientation.



4.7: FUTURE WORK

The first step in future work would be fully perfecting the tracing algorithm. There are still several cases where it could use improvement. For instance, the code

encounters difficulties when a fracture abruptly changes from very thin to very thick, as in the presence of a large void space, especially if the original user-drawn traverse was short compared to the size of the void. The handling of bifurcations, namely in 3-D, is also an area that needs improvement; in such cases, it is not entirely clear how to allocate fracture sheets and continue tracing. Additionally, the connectivity information between tiles and sheets could be improved. The data structures are in place to implement this to some degree, but the current version does not perform this task very well. Using spatial tessellations or meshes could be a solution to this problem of adjacency.

Many new quantitative relationships can be measured with the data this method provides, leading to new insights for the nature and behavior of fracture networks. For example, the standard deviation of the best-fit plane of the midpoints could be compared to the standard deviation of the best-fit planes of both the top and bottom edges of the fracture, reflecting the fracture roughness. For example, if the top and midpoint planes are more highly variant while the bottom plane has an extremely low variance, this could indicate a corroded or weathered upper surface of the fracture and perhaps a flat contact at the bottom. Another way to quantify fracture roughness would be to take the angle between the best-fit planes of the top and bottom edges of the fracture within each tile. The angle "divergence" between the top and bottom planes is another way to quantify the 3-D orientation of an individual tile; this is a measurement then that can be tracked throughout an entire sheet or full network. Given a sample with many disparate fractures, the aperture, length, and orientation information can be compared between each of the separate units.

4.7.1: Applications

After the network tracing algorithm is perfected, there are many different applications for its use. One area of study that would benefit from accurate 3-D fracture network information concerns the geometry and structure of the network itself. Fracture topology can influence various processes, such as pollution dispersal. Random network generation is a common tool used to study different properties theoretically expected (as in Huseby et al., 1996 and Andresen et al., 2012). Some 2-D studies have looked at non-simulation fractures. Natural fracture networks have been demonstrated to exhibit the same topological properties as "small-world" networks (Valentini et al., 2006). A small world network is defined as the distance L between any two nodes of the graph is proportional to the log of the number of nodes, N. The study only transcribed graphs from 2-D images, however. It would be interesting to see if the small-world hypothesis is still confirmed for 3-D networks.

Another potential application is fluid-flow modelling. For example, a 2-D aperture map was used to model Navier-Stokes flow (Cardenas et al., 2007). Different sections of the fracture were partitioned to observe flow and transport properties under different aperture conditions. It was observed that small-scale changes in fracture geometry greatly impact the flowlines of solute transport through the fracture. It is shown that if geometry permits, eddies will form that produce tailings in flow, which can be described by power laws. This implies that residence times for solutes within a particular area of a fracture are often dependent on small changes in geometry. Flow rate is consistently connected to the fracture aperture geometry (Dijk and Berkowitz, 1999),

again demonstrating the need for accurate aperture information. It is reasonable to expect 3-D geometry to impact solute residence time, and being able to investigate solute residence for 3-D networks would be a useful advancement. In a follow-up study, the major flow direction was reversed along the same fracture to show differences in eddy geometry and residence time, which varied as a result (Cardenas et al., 2009). In another 2-D study, non-Fickian transport was demonstrated to vary based on the heterogeneities of a fracture (Wang and Cardenas, 2014). With a 3-D network sample, there are many more directions for fluids than just back and forth, adding a degree of anisotropic complexity to how directional flow is conceptualized. Prediction of transport will be much more complicated in a full 3-D network. A modified Local Cubic Law has also been developed to quantify hydrologic properties of rough fractures, and it would be useful to see if the approach works on real 3-D fracture data for more complex transport understanding (Wang et al., 2015).

Further complexities can be later added to a fluid-flow model, including the addition of fracture skins (Robinson et al., 1998) or chemical dissolution and precipitation. CT was also used to study diffusivity within cracked concrete samples to better determine the durability of concrete (Darma et al., 2013). Not only are the fractures impacting flow, but capillary absorption into the concrete does as well. Ultimately, the big-picture question is to what degree results from CT labs can be upscaled to situations encountered in the field: from the lab scale all the way up to the reservoir scale. Preliminary research suggests that upscaling works for some hydraulic properties, while others cannot reliably make the jump from lab to field (Sharp et al.,

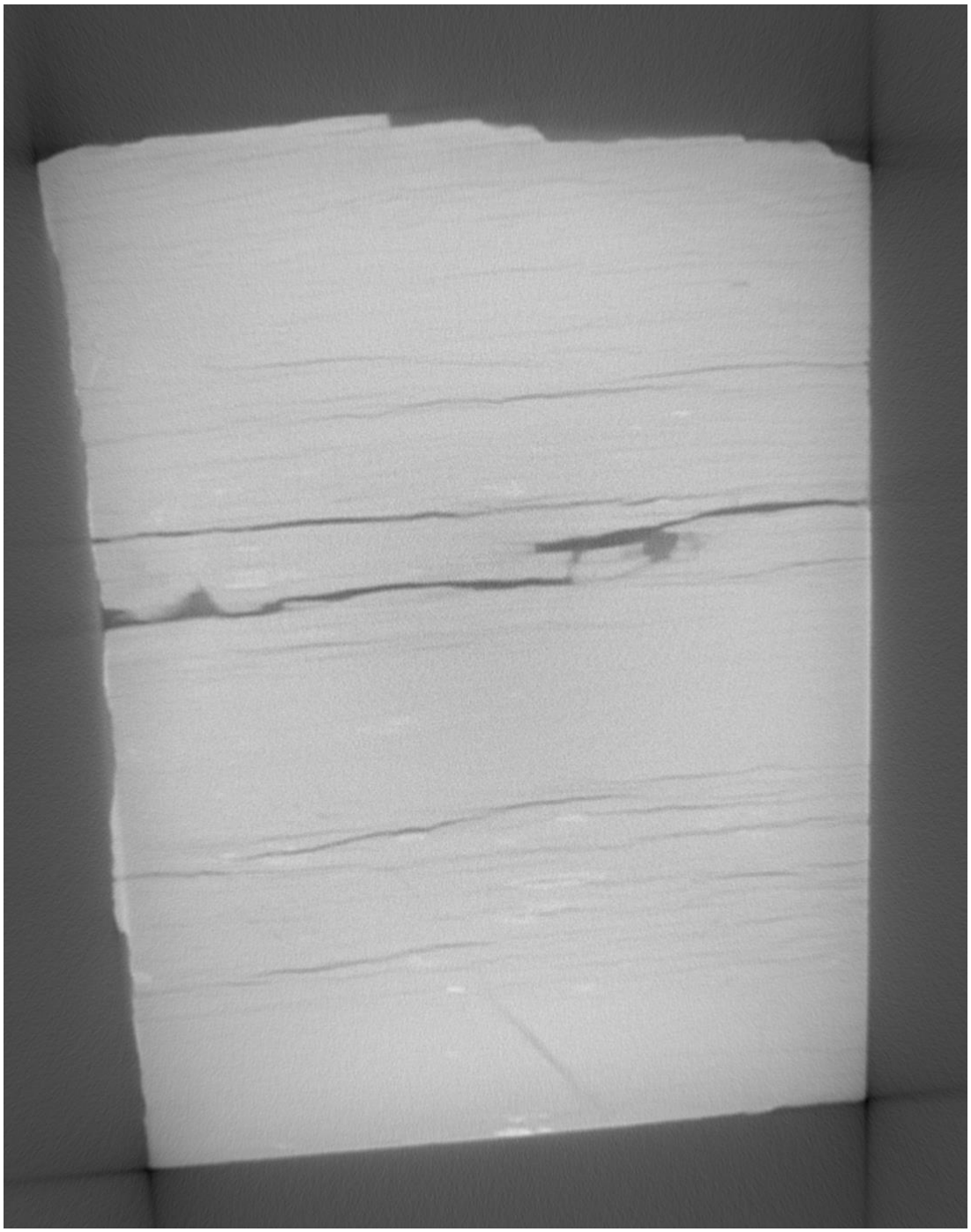
2014). The hydraulic aperture, for example, was not a confident candidate for upscaling, whereas surface roughness was performed reasonably well. Upscaling using statistical modeling has also been attempted on various parameters associated with real fracture networks at Yucca Mountain (Liu et al., 2009) and purely mathematically (Kfoury et al., 2004).

Chapter 5: Conclusions

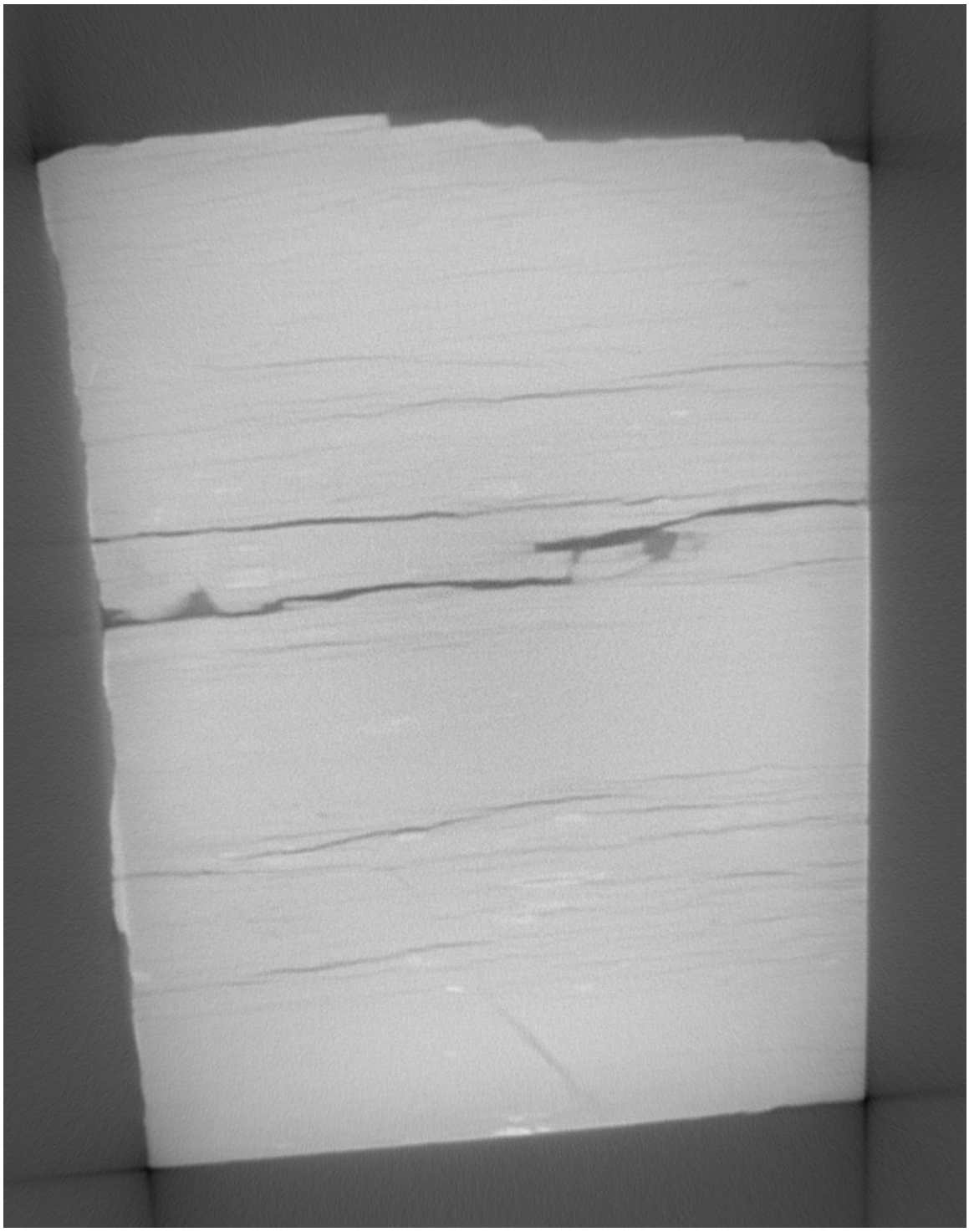
A fracture tracing algorithm was developed for use in analyzing CT data to identify fracture networks and extract information on the location of the fracture walls in order to determine the network topology and calculate various characteristics such as aperture and roughness. The algorithm uses information about the blurring inherent in a CT scan, which is represented by the PSF, to increase the accuracy of the measurements taken. Underlying the algorithm is an object-oriented model of describing a fracture network as a set of sheets, composed of smaller tiles, defined by a set of linear traverses. A test was performed on a real fracture network in a sample of Packsaddle schist, which has a challenging arrangement of thin fractures with anastomosing bifurcations. Some of the network bifurcations were identified by the algorithm. The iterative, local approach the algorithm takes to determine the fracture boundaries provides more robust information, though at a higher computational cost. Future improvements will make this a more viable research tool, as the current implementation does not account for all of the complicated features exhibited by real fracture networks. The current state though is a positive advancement toward characterizing real fracture networks imaged with CT that are more than just planar.

Appendix A: Packsaddle Test Images

"Slice 12" Packsaddle Schist



“Slice 13” Packsaddle Schist



Appendix B: Pseudocode

3-D Fracture Tracing Outline

This pseudocode is a general, high-level flow of the fracture tracing logic, color-coded and commented to help follow the method calls.

```
%At the highest level, the program is meant to obtain a user traverse, map the fracture, then report the data
------------------
AnalyzeFracture(User Traverse)
        StorageUnit = MapFracture3D(UserTraverse)
        ReportFractureData(StorageUnit)
End


%To map a fracture in a volume, map it in a single slice, then go to each other slice
------------------
MapFracture3D(UserTraverse)
        %stores all sheets
        StorageUnit = CreateContainer()
        MapFractureInSlice(UserTraverse, StorageUnit)
        %map for all the higher slices
        NextSlice(+1)
        %restart at the original slice and go in the opposite direction through the lower slices
        NextSlice(-1)
        Return StorageUnit
End
```

%For a single slice, make a Tile around the original user traverse, then follow the fracture within the slice

---

MapFractureInSlice(UserTraverse, StorageUnit)

    MakeTile(InitPos From User traverse, Sheet)

    %go forward and then backward along the fracture length

    PropagateFracture(+1)

    PropagateFracture(-1)

    If bad, return fail

End

%To make a tile, first make the central traverse. If it's valid, continue to create the tile and place it in its sheet

---

MakeTile(nextPos, Sheet, dir)

    Trav = MakeTrav(nextPos)

    EvalTrav(trav)

        -If bad, return fail

    Tile = FormTile(trav)

    EvalTile(tile, dir)

        -if bad, return fail

    AddTileToSheet(Sheet)

End

%Trace the fracture in the dir(ection) given until it meets a data boundary or other fail condition

---

PropagateFracture(dir, dataBoundary, tile)

    While not fail

        nextPos = FindNextCentralTrav(orientation, stepDirection, nextPos, dataBoundary)

        MakeTile(nextPos, StorageUnit, dir)

End

MakeTrav(initPos)

    GetVoxelData(initPos)

    FindFractureAperture()

    Return Trav

End

EvalTrav(trav)

    DoBothEndsInMatrixText()

    DoFracturePresentTest()

    Return T/F

End

FindNextCentralTrav(orientation, stepDirection, initPos, dataBoundary)

    [newStartPt, newEndPt] = CalculateStep()

    If not within dataBoundary, return fail

```
        Return newStartPt and NewEndPt
End


%A tile is formed with 8 additional traverses surrounding a central traverse
-----------------
FormTile(trav)
        tile = CreateTile(trav)
        Locations = CalculateLocationsForOther8()
        For Locations:
                MakeTrav(Location)
                EvalTrav
                Add to Tile
        CalcBestFitPlane(tile)
End


-----------------
EvalTile(tile, dir)
        DoMidpointOutlierTest()  ;best-fit plane confidence interval
        DoTerminationTest()
        DoBifurcationTest()
End
```

References

1. Andresen, C., Hansen, A., Le Goc, R., Davy, P., and Hope, S., 2012, Topology of Fracture Networks, *Physics and Society,* doi: 10.3389/fphy.2013.00007.

2. ASTM, 1992, Standard guide for computed tomography (CT) imaging, ASTM designation E1441-92a.

3. Baveye, P., Laba, L., Otten, W., Bouckaert, L., Sterpaio, P., Goswami, R., Grinev, D., Houston, A., Hu, Y., Liu, J., Mooney, S., Pajor, R., Sleutel, S., Tarquis, A., Wang, W., Wei, Q., Sezgin, M., Observer-dependent variability of the thresholding step in the quantitative analysis of soil images and X-ray microtomography data, *Geoderma*, v. 157, Issues 1–2, 15 June 2010, p. 51-63, ISSN 0016-7061, http://dx.doi.org/10.1016/j.geoderma.2010.03.015.

4. Bertels, S., DiCarlo, D., and Blunt, M., 2001, Measurement of aperture distribution, capillary pressure, relative permeability, and *in situ* saturation in a rock fracture using computed tomography scanning, *Water Resources Research*, v. 37, p. 649-662.

5. Bresenhem, J., 1965, Algorithm for computer control of a digital plotter, *IBM Systems Journal*, v. 4, p. 25-30, doi:10.1147/sj.41.0025.

6. Cardenas, M., Slottke, D., Ketcham, R., and Sharp, J., 2007, Navier-Stokes flow and transport simulations using real fractures shows heavy tailing due to eddies, *Geophysical Research Letters*, v.34, L14404, doi:10.1029/2007GL030525.

7. Cardenas, M., Slottke, D., Ketcham, R., and Sharp, J., 2009, Effects of inertia and

directionality on flow and transport in a rough asymmetric fracture, *Journal of Geophysical Research*, v. 114, B06204.

8. Clark, D., 2013, Beginning C# Object-Oriented Programming, Apress.

9. Cloetens, P., Barrett, R., Baruchel, J., Guigay, J.-P., and Schlenker, M., 1996, Phase objects in synchrotron radiation hard x-ray imaging, J. Phys. D, 29, p. 133-146.

10. Darma, I., Sugiyama, T., and Promentilla, M., 2013, Application of X-ray CT to study diffusivity in cracked concrete through the observation of tracer transport, *Journal of Advanced Conrete Technology*, v. 11, p. 266-281.

11. Dijk, P. and Berkowitz, B., 1999, Three-dimensional flow measurements in rock fractures, *Water Resources Research*, v. 35, n. 12, p. 3955-3959.

12. Erhel, J., De Dreuzy, J., and Poirriez, B., 2009, Flow simulation in three-dimensional discrete fracture networks, Siam J., *Sci. Comput*, v. 31, n. 4, p. 2688-2705.

13. Holtzman-Gazit, M., Kimmel, R., Peled, N., and Goldsher, D., 2006, Segmentation of thin structures in volumetric medical images, *IEEE Transactions on Image Processing*, v. 15, n. 2, p. 354-362.

14. Huseby, O., Thovert, J.-F., and Adler, P., 1996, Geometry and topology of fracture systems, *J. Phys. Math A*: Gen. 30, p. 1415-1444.

15. Iassonov, P., T. Gebrenegus, and M. Tuller (2009), Segmentation of X-ray computed tomography images of porous materials: A crucial step for characterization and quantitative analysis of pore structures, *Water Resour. Res.*, 45, W09415, doi:10.1029/2009WR008087.

16. Johns et al, 1993, Nondestructive measurements of fracture aperture in crystalline rock cores using X-ray computed tomography, *Journal of Geophysical Research*, v. 98, i. B2, p. 1889-1900.

17. Ketcham, R. and Carlson, W., 2001, Acquisition, optimization and interpretation of X-ray computed tomographic imagery: applications to the geosciences, *Computers and Geosciences* 27, p. 381-400.

18. Ketcham, R., 2006, Accurate three-dimensional measurements of features in geologic materials from X-ray computed tomography data, Advances in X-ray Tomography for Geosciences, 2nd International Workshop on X-ray CT for Geomaterials, p. 143-148.

19. Ketcham, R., and Ryan, T., 2004, Quantification and visualization of anisotropy in trabecular bone, *Journal of Microscopy*, v. 21, n. 3, p. 158-171.

20. Ketcham, R., Slottke, D., and Sharp, J., 2010, Three-dimensional measurement of fractures in heterogeneous materials using high-resolution X-ray computed tomography, *Geosphere*, v. 6, n. 5, p. 499-514.

21. Ketcham, R., 2005a, Computational methods for quantitative analysis of three dimensional features in geological specimens, *Geosphere*, v. 1, n. 1, p. 32-41.

22. Ketcham, R., 2005b, Three-dimensional grain fabric measurements using high resolution X-ray computed tomography, *Journal of Structural Geology* 27, p. 1217-1228.

23. Ketcham, R., Sharp, J., Hildebrandt, J., and Slottke, D., April 2013, Measurements at the edge of resolution: Characterizing fractures in 3D using high-resolution X-ray

computed tomography, GSA South-Central Section Annual Meeting, AT&T Center, Austin, TX, Talk.

24. Ketcham, R., and Hanna, R., June 2014, Beam hardening correction for X-ray computed tomography of heterogeneous natural materials, *Computers & Geosciences*, v. 67, p. 49-61.

25. Ketcham, R. and J. Hildebrandt, 2014, Characterizing, measuring, and utilizing the resolution of CT imagery for improved quantification of fine-scale features, *Nucl. Instr. Meth. B* 324, p.80-87.

26. Kfoury, M., Ababou, R., Noetinger, B., and Quintard, M., 2004, Matrix-fracture exchange in a fractured porous medium: stochastic upscaling, *C.R. Mechanique* 332, p. 679-686.

27. Liu, X., Zhang, C., Liu, Q., and Birkholzer, J., 2008, Multiple-point statistical prediction on fracture networks at Yucca Mountain, *Environ. Geol.* 57, p. 1361 1370.

28. Olson, J., 2003, Sublinear scaling of fracture aperture versus length: An exception or the rule?, *J. Geophys. Res*., v. 108, n. B9, 2413.

29. Otsu, N., 1979, "A threshold selection method from gray-level histograms". IEEE Trans. Sys., Man., Cyber., v. 9 (1), p. 62-66. doi:10.1109/TSMC.1979.4310076.

30. Park, R.G., 1997, Foundations of Structural Geology, Chapman & Hall, p. 9.

31. Ramachandran, G.N., and Lakshminarayanan, A.V., 1970, Three-dimensional reconstruction from radiographs and electron micrographs: Application of convolutions instead of Fourier transforms, Proceedings of the National Academy

of Science 68, p. 2236-2240.

32. Reese, J., Mosher, S., Connelly, J., and Roback, R., 2000, Mesoproterozoic chronostratigraphy of the southeastern Llano uplift, central Texas, *GSA Bulletin*, v. 112, n.2, p. 278-291.

33. Robinson, N., Sharp, J., and Kriesel, I., 1998, Contaminant transport in sets of parallel finite fractures with fracture skins, *Journal of Contaminant Hydrology* 31, p. 83-109.

34. Sezgin, M. and Sankur, B., 2004, "Survey over image thresholding techniques and quantitative performance evaluation". *Journal of Electronic Imaging* 13 (1): 146–165. doi:10.1117/1.1631315.

35. Sharp, J., 1993, Fractured aquifers/reservoirs: approaches, problems, and opportunities, Memoirs of the XXIVth Congress of IAH, As, Oslo.

36. Sharp, J., Al-Johar, M., Slottke, D., and Ketcham, R., 2014, Prediction of fracture roughness and other hydraulic properties: is upscaling possible?, *Fractured Rock Hydrogeology*, p. 167-179.

37. Slottke, D., 2010, Surface roughness of natural rock fractures: implications for prediction of fluid flow (Doctoral dissertation).

38. Valentini, L., Perugini, D., and Poli, G., 2007, The "small-world" topology of rock fracture networks, *Physics A* 377, p. 323-328.

39. Vandersteen, K., Busselen, B., Van Den Abeele, K., and Carmeliet, J., 2003, Quantitative characterization of fracture apertures using microfocus computed tomography, Geological Society, London, Special Publications, v. 215, p. 61-68.

40. Voorn, M., Exner, U., and Rath, A., 2013, Multiscale Hessian fracture filtering for the enhancement and segmentation of narrow fractures in 3D image data, *Computers and Geosciences* 57, p. 44-53.

41. Wang, L., and Cardenas, M.B., 2014, Non-Fickian transport through two-dimensional rough fractures: assessment and prediction, *Water Resources Research*, 50, 871-884.

42. Wang, L., Cardenas, M.B., Slottke, D.T., Ketcham, R.A., and Sharp, J.M. Jr., 2015, Modification of the Local Cubic Law of fracture flow for weak inertia, tortuosity, and roughness. *Water Resources Research*, 51, 2064-2080.

43. Watanabe, N., Ishibashi, T., Ohsaki, Y., Tsuchiya, Y., Tamagawa, T., Hirano, N., Okabe, H., and Tsuchiya, N., 2011, X-Ray CT based numerical analysis of fracture flow for core samples under various confining pressures, *Engineering Geology*, v. 123, i. 4, p. 338-346.

44. Wellington, S. and Vinegar, H., 1987, X-ray computerized tomography, *Journal of Petroleum Technology*, v. 39, p.885-898.