

INVESTIGATING THE VOLUME AND STRUCTURE OF POROSITY IN
FRACTURED AND UNFRACTURED ROCKS FROM THE NEWBERRY
VOLCANO, OREGON:
AN EVALUATION AND COMPARISON OF TWO- AND THREE-
DIMENSIONAL METHODS

A Thesis
Submitted to
The Temple University Graduate Board

In Partial Fulfillment
of the Requirements for the Degree
MASTER OF SCIENCE

By
Justin M. Roth
January 2014

Thesis Approvals:

Dr. Nicholas C. Davatzes, Thesis Advisor, Earth and Environmental Science
Dr. Alexandra Krull Davatzes, Thesis Advisor, Earth and Environmental Science
Dr. David E. Grandstaff, Earth and Environmental Science

UMI Number: 1552297

All rights reserved

INFORMATION TO ALL USERS

The quality of this reproduction is dependent upon the quality of the copy submitted.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if material had to be removed, a note will indicate the deletion.



UMI 1552297

Published by ProQuest LLC (2014). Copyright in the Dissertation held by the Author.

Microform Edition © ProQuest LLC.

All rights reserved. This work is protected against unauthorized copying under Title 17, United States Code



ProQuest LLC.
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106 - 1346

ABSTRACT

Porosity is a fundamental characteristic of rock critical to its mechanical and hydrologic behavior, yet a study of the open and accumulated healed porosity of nine core samples from Newberry Volcano shows that different measurement methods produce significantly different estimates of pore volume and structure. This study compares traditional 2D point count, petrographic image analysis, and 3D x-ray Micro Computed Tomography (micro CT) measurement of porosity primarily derived from fracture slip and dilation. The set of measurements quantifies the discrepancy among measurement methods and provides a basis for assessing how this uncertainty depends on geologic factors including the stage of fracture development, and the size and connectivity of the pores. This comparison reveals that detailed petrographic mapping provides the most accurate characterization of fracture porosity, and its history of development, owing to its high spatial resolution and accuracy of phase identification as well as insights afforded from mineralogic and textural relationships. However, this analysis lacks the three-dimensional characterization necessary to determine pore shape and interconnectedness, especially in highly anisotropic and heterogeneous fracture porosity. Micro CT does characterize the three dimensionality of pores, and thus although it consistently underestimates porosity due to non-uniqueness of phase densities and limitations in resolution, and is difficult to post process, this method can usefully augment the petrographic analysis.

High resolution mapping of petrographic thin sections also provides a means to characterize the roughness of fracture surfaces across multiple cycles of slip, related dilation, and healing. Analysis of 19 slip events on a small, early stage fracture experiencing less than mm-scale slip, indicates that this roughness is preserved across

multiple slip events and is consistently associated with dilation recorded by the accumulation of layers of precipitated cement. Initially, characteristic length scales intrinsic to rock such as the primary grain and pore size distribution of the > 0.2 mm size fraction significantly influence the roughness of fractures, until the dominant mechanism of fracture growth becomes linkage among macroscopic fractures. This correlation among primary rock characteristics such as grain size, fracture roughness, repeated fracture slip, and dilation provides a potential method to assess the key attributes promoting dilatant, self-propagating fracture slip necessary for successful stimulation to generate an Enhanced Geothermal System. Comparison to more developed fractures characterized by the development of fault rock suggest such stimulation is most successful for fractures sustaining small slip of a few millimeters or less during single slip events.

ACKNOWLEDGEMENTS

I would like to thank the EGI core repository at the University of Utah, Salt Lake City for providing the Newberry Core used in this study, Mamta Amin and Mary Barbe for granting me access to the micro CT scanner at the Temple University School of Medicine, James Ladd for his technical support and services, Shelah Cox for her endless help and advice, David Grandstaff for serving on my advisory committee, Alix Davatzes for serving as an advisor of this research, keeping my progress in check, and providing useful insight to the study, Nick Davatzes for serving as an advisor to this research and using his grants to fund this research, for the countless hours he spent with me—knowingly and unknowingly—showing me what it means to have a strong work ethic, discussing with me ideas about this study, and helping me get through problems encountered in MATLAB. Lastly, I'd like to thank my family, my friends, and my fellow graduate students for all of their help and support over the past few years. This study is based upon work supported by a grant from the Department of Energy, Award Number: DE-EE0002757.

TABLE OF CONTENTS

	PAGE
ABSTRACT	ii
ACKNOWLEDGEMENTS	iv
LIST OF TABLES	viii
LIST OF FIGURES	ix
CHAPTER	
1. INTRODUCTION	1
1.1. Purpose and Goals	3
1.2. Background	6
1.2.1. Survey of Techniques for Measuring Porosity	6
1.2.2. Fracture Roughness	13
1.2.3. Definition of Terms	14
1.3. Geologic Setting	15
1.3.1. Structure of the Newberry Volcano	16
1.3.2. Geothermal Potential of Newberry Volcano	20
1.3.3. Lithology of the Geo-N2 Well	22
2. METHODS	23
2.1. Overview	23
2.2. Sample Preparation	27

2.3. Point Counts	28
2.4. Automated Image Analysis	30
2.5. High Resolution Image Analysis	31
2.5.1. Thresholding High-Resolution Images	32
2.5.2. Determining Porosity Amounts	33
2.6. 3D Micro Computed Tomography Scans	38
2.6.1. Reconstruction and Post-processing	40
2.6.2. Determining Porosity Amounts	42
2.6.3. A Note About Thresholding Categories	45
2.7 Measurements of Fracture Roughness	46
2.7.1. Identifying, Digitizing, and Assessing Fracture Surfaces	47
2.7.2. Determining Grain and Pore Size Distribution	51
3. RESULTS	52
3.1. Description of Porosity in Samples	52
3.2 Point Counts and Automated Image Analyses	52
3.3. High-Resolution Images and Micro CT Scans	55
3.4. Multi-Transect Analysis to Asses Statistical Robustness of Measured Porosity	61
3.4.1. High-Resolution Image Transect Variability	61
3.4.2. Micro CT Transect Variability	64
3.5. Fracture Surface Roughness	69

4. DISCUSSION	72
4.1. Method Comparison	72
4.1.1. Point Counts versus Automated Image Analyses	72
4.1.2. High Resolution Images vs. Micro CT Scans	73
4.2. Geologic Context	79
4.2.1. Key Attributes of Developing Fracture Porosity Revealed by Analyses	79
4.2.2. Fracture Surface Roughness	84
4.2.3. History of Dilatancy and Implications for EGS	85
4.2.4 Correlations to EGS Induced Seismicity	92
5. CONCLUSION	95
REFERENCES CITED	98
APPENDIX	105
A. MATLAB SCRIPTS AND FUNCTIONS	105
B. AVERAGE POROSITY VALUES AND METHOD CHECKLIST FOR EACH SAMPLE	181
C. POROSITY MAPS AND PLOTS FOR POINT COUNTS AND AUTOMATED IMAGE ANALYSIS	183
D. RESULTS OF HIGH RESOLUTION IMAGE ANALYSES	192
E. MICRO CT SCAN RESULTS	198

LIST OF TABLES

Table	Page
1.2.1 Summary of porosity measurement techniques	7
2.1.1 Summary of thin section/core classification and fracture stage	27
4.1.1 Brief summary of the advantages, disadvantages, and pixel size of each method evaluated in this study	78

LIST OF FIGURES

Figure	Page
1.2.1 Thin Section 4303 PFWA and generalized porosity map	12
1.2.2 Cartoon showing different orientations of a 2D slice of rock	12
1.2.3 Thin section displaying the various types of porosity	15
1.3.1 Regional view of Newberry Volcano showing the effects of Cascade volcanism	17
1.3.2 Cross section showing the major stratigraphic units, structural boundaries, and geothermal temperature profile	18
1.3.3 Satellite view showing the faulting regime at Newberry Volcano	19
1.3.4 Topographic map of Newberry National Volcanic Monument	21
2.1.1 Components of a fault	26
2.2.1 Schematic showing the relative locations of thin section slices and sub-cores	28
2.3.1 Plot of number of point counts vs. porosity percent	30
2.5.1 Summary of high resolution image analysis	35
2.5.2 Window sensitivity analysis for high resolution images	36
2.5.3 Workflow diagram conceptualizing the methods and results for high resolution image analysis	37
2.6.1 Schematic of how x-rays pass through a specimen during a micro CT scan	38
2.6.2 Image of the Temple University School of Medicine's SkyScan 1172 high-resolution micro CT scanner	39
2.6.3 Cross-sectional slice of micro CT scan	41
2.6.4 Pixel grayscale value vs. material density	42
2.6.5 Window sensitivity analysis for micro CT scans	44

2.6.6	Micro CT method	45
2.6.7	Flow diagram conceptualizing the methods involved in micro CT scans	48
2.7.1	Photograph of the Luminoscope [®] Model ELM-2A used for cathodoluminescence analysis	50
3.1.1	Thin Sections showing types of porosity in Newberry Rock	53
3.2.1	Thin section analysis results	55
3.3.1	High resolution image and micro CT results	58
3.3.2	Anomalous peaks in micro CT data	59
3.3.3	Edge effects of micro CT scans	60
3.3.4	Areas of low attenuation found near pyrite grains	61
3.4.1	Perspective view of results of open and healed porosity multi-transect analysis for high resolution image analysis	62
3.4.2	High resolution image based analysis of open and healed porosity along multiple transects parallel to the y-axis of the sample	63
3.4.3	Statistical variability in the open and healed porosity along the family of 13 high resolution image transects	64
3.4.4	Visual of multi-transect analysis for micro CT scans	65
3.4.5	Micro CT based analysis of open and healed porosity along nine transects parallel to the z-axis of the sample	66
3.4.6	Statistical variability in the open and healed porosity along the family of nine Micro CT transects	67
3.5.1	Fracture roughness results	70
3.5.2	Plots from fracture surface data	71
4.1.1	Thresholding and despeckling images from micro CT post-processing	75
4.1.2	Further investigation of unexplained porosity peaks	77
4.2.1	Fracture stage vs. percent skeletal porosity	81

4.2.2	High resolution image scan showing a completely healed fracture	82
4.2.3	High resolution image scan of a fracture that contains open porosity	82
4.2.4	Micro CT scans showing low open porosity	84
4.2.5	Model of tendency toward dilation, simple shear, or compaction due to stress boundary conditions	87
4.2.6	Induced seismic events in the vicinity of Newberry Core	94

CHAPTER 1

INTRODUCTION

Porosity in rock is a critical control on the storage and flow of fluid in the subsurface and is thus fundamental to the study of geothermal, petroleum, and hydrologic systems (e.g., Ingebritsen and Sanford, 1998). Accurate measurements of porosity volume and structure, defined as the shape, size, orientation, and distribution of pores, are therefore necessary to model fluid storage, fluid flow, and the surface area exposed to fluids during flow under natural flow, production, injection, and stimulation conditions. A better understanding of transport through rock pore space is critical to both the geothermal and petroleum industries, in which natural circulation of hot water or petroleum controls accumulation and access to energy resources. Similarly, water resources and the evaluation and remediation of contaminant transport critically depends on porosity. In these cases, accurate modeling of fluid circulation must include properties that explicitly depend on porosity and pore structure (e.g., Ingebritsen and Sanford, 1998). In addition, porosity plays an important role in determining the failure mode of geologic materials (e.g., Paterson and Wong, 2005). For instance, large pore volume or size tends to produce porosity loss during shearing, whereas very low porosity can lead to high strength and low permeability, but is shown to promote dilatant failure (Zhu and Wong, 1997). Similarly, porosity between fracture surfaces is related to surface roughness and contact area and could impact the slip behavior of the fracture including its potential for dilation (e.g., Barton, 1973, 1986; Brown, 1987). In addition, the pore structure in the volume adjacent to these surfaces can influence the stiffness of asperities

and local stresses (e.g., Paterson and Wong, 2005). Since porosity and pore connectivity, including the regeneration of porosity lost due to sediment compaction or mineral precipitation, is often controlled by fractures, especially at several kilometers depth (e.g., Committee on Fracture Characterization and Fluid Flow, 1996; Currewitz and Karson, 1997), it is necessary to characterize the true porosity and the structure of pores to fully understand groundwater, petroleum, and geothermal systems.

Porosity is derived from and modified by a wide variety of processes including sedimentation, compaction, cementation, dissolution, alteration and deformation that result in a variety of pore shapes and size distributions, as well as textural and mineralogic changes (Crawford et al., 2002; Krauskopf and Bird, 1995). In geothermal systems in particular, porosity is primarily added through fracture formation and slip (e.g., Davatzes and Hickman, 2010). The history of recurrent fracture slip and porosity production in fractures is evident from the varying amounts of open, healed, and skeletal (open + healed) porosity present (e.g., Antonellini and Ayden, 1994; Antonellini et al., 1994; Lee and Wiltschko, 2000; Fetterman and Davatzes, 2011; Facca and Tonani, 1967).

A variety of techniques including thin section analysis and high resolution Micro Computed Tomography (CT) are currently used to measure porosity structure. These techniques balance practicality against the goal of accuracy and precision in an attempt to characterize the total three-dimensional porosity volume and, perhaps more importantly, the structure of those pores. Though various techniques are used to study porosity in the geosciences, a detailed, direct comparison of methods has yet to be completed.

1.1 Purpose and Goals

The first part of this study evaluates and compares two-dimensional and three-dimensional techniques used to quantify the volume and structure of pores in rocks from the Newberry Volcano, Oregon, and investigates how the measured values of volume differ as function of the technique being used. In the measurement of porosity, there is a trade-off between the time it takes to perform the analysis, the cost of the analysis, and the resolution of the method, which can, in turn, limit the number of analyses that can be performed. In addition, the fracturing and deformation of rock introduces anisotropic grain and pore geometry and heterogeneous pore distribution. The result is preferentially oriented, highly elongated pores whose volume and structure will be portrayed differently depending on which technique is chosen for analysis, the plane of observation, and the relative size of the volume measured. Thus, a careful study of the relative differences between 2D and 3D techniques in intact rocks as well as rocks fractured to different levels of development needs to be conducted. Porosity-measuring techniques have been employed over decades, generating large amounts of data. In order to use these data in an appropriate and self-consistent manner, this study establishes how these different measurements correspond in the case of fracture porosity. In particular, the first part of this study compares point counts and image analyses of thin sections (2D techniques) along with Micro Computed Tomography (CT) scans (3D technique) in an attempt to investigate and classify the precision, variance, and introduced error of these two-dimensional and three-dimensional methods. The porosity in rock volumes containing a range of developed fractures is measured. This range, increasing with fracture development, consists of simple fractures, linked fractures, proto-breccia, and lastly,

developed fault breccia (Fetterman and Davatzes, 2011). Both the reliability of the techniques in measuring porosity across fractures characterized by differences in the development of fault rocks and the characterization of the change in porosity associated with fracture development can therefore be assessed.

In the second part of this study, the relationship between fracture dilation during slip, host rock properties, and the evolving roughness of multiply-slipped fractures is investigated. High resolution mapping of petrographic thin sections is utilized to investigate the history of fracture surface roughness and the corresponding history of dilation and healing in a natural fracture experiencing repeated failure in an attempt to relate roughness features to intrinsic properties (e.g. grain and pore size distribution) of the host rock. In general, the size, shape, and curvature of asperities in contact across fracture surfaces affect mechanical properties of the rock mass (e.g. Brown and Scholz, 1985, 1986; Barton, 1986; Power and Tullis, 1992; Jaeger et al., 2007; Barton, 2007). If roughness is influenced initially by the characteristic length scales of the rock mass associated with grain and primary pore size, and then by the evolving distribution of cracks, the potential for dilation during fracture slip, and in particular slip induced by injecting water, could be partly assessed *a priori* through characterizing these attributes. The corollary to this hypothesis is that if these basic attributes evolve as fractures slip (new pores are introduced or the grain size distribution within and adjacent to the fracture walls is modified), the fracture's capacity for dilation will change.

Documentation of the history of porosity production establishes the most basic necessary criteria for developing an Enhanced Geothermal System (EGS) in hot but otherwise low permeability rocks. The basis for EGS involves the injection of water to

induce self-propping shear failure, a process known as hydro-shearing. In addition, assessment of porosity associated with different levels of fault zone development, which is evident from the formation and accumulation of fault gouge, provides a 1st order insight into the ability of faults and fractures to dilate during natural or induced slip (e.g., Fetterman and Davatzes, 2011). Careful measurements of surface roughness in an early stage fracture that has sustained multiple dilation events are also conducted. The documentation of repeated failures provides the opportunity to assess: (1) dilation accompanying multiple slip events, (2) how dilation is related to roughness, (3) how roughness is related to initial rock characteristics, such as primary grain and pores size, and (4) how surface roughness evolves over the short slip distances expected in most of the fractures stimulated during a low pressure EGS injection stimulation.

This investigation is conducted on naturally fractured rocks obtained from core at depths from 1103 to 1313 m at Newberry Volcano, Oregon. Core is obtained from well GEO N-2, which is within 0.5 km of the Enhanced Geothermal Systems (EGS) demonstration well NWG 55-29. Well NWG 55-29 was initially stimulated by the injection of relatively cold water in October through December of 2012, resulting in earthquakes and increased transmissivity (Cladouhos et al., 2013; Petty et al., 2013). The seismic events and increased transmissivity are interpreted as a new cycle of porosity generation associated with slip on natural fractures, consistent with the history of porosity generation previously documented in the natural fractures in the core.

1.2 Background

1.2.1 Survey of Techniques for Measuring Porosity

Table 1.2.1 provides a brief survey of the most common techniques used to measure porosity, along with a summary of their key attributes. Although many techniques are available to measure porosity (Table 1.2.1), the analysis time, number of measurements that can be produced, and cost vary significantly between methods and thus limit use of these methods. Thin sections allow direct measurement of porosity with textural and mineralogic evidence of porosity evolution. However, critical properties of pores, including their connectedness and their 3D size and shape, are not completely revealed by thin section analysis in 2D. Conversely, micro CT characterizes a volume, providing insights into the full 3D structure and connectedness of pores, but is limited because minerals cannot be uniquely identified by x-ray attenuation, especially if pores are very small ($<\sim 1 \mu\text{m}$) or very thin (micro-crack pores). These differences lead to potential inconsistencies in characterizing the porosity structure and its development that are necessary to understand the permeability of fractured rock, its ability to be stimulated by increasing fluid pressure to induce shearing accompanied by dilation, and the likelihood of seismic energy release. Although this study focuses on lab-based measurements of porosity, and in particular thin sections because of the context they provide, there are also a variety of techniques for *in situ* measurements of porosity, many of which are adapted to provide continuous measurements of porosity in boreholes.

Table 1.2.1

Summary of Porosity Measurement Techniques

Technique	Dimensions	Pore Length-Scale	REV Length-Scale	Pore Properties	Types of Porosity
Laboratory					
Gas Porosimetry	3	10^{-4} to 10^{-2}	10^{-2} to 10^{-1}	Magnitude	Open, effective
Fluid Porosimetry	3	10^{-4} to 10^{-2}	10^{-2} to 10^{-1}	Magnitude	Open, effective
NMR	3		10^{-2} to 10^{-1}	Magnitude Pore Size Spatial Distribution	Water-filled Total Open Effective
Micro CT	3	10^{-3} to 10^{-1}	10^{-2} to 10^{-1}	Magnitude Pore Size Pore Shape Spatial Distribution	Total Open Healed (case-by-case; simple, known mineralogy only) Effective
Thin Section (Optical) (Image Analysis)	2	10^{-5} to 10^{-1}	10^{-2} to 10^{-2}	Magnitude Apparent Pore Size Apparent Pore Shape Spatial Distribution Mechanism of Porosity Formation	Total Open Healed (case-by-case; Skeletal)
Thin Section (Optical) (Point Counting)	2	10^{-5} to 10^{-1}	10^{-2} to 10^{-2}	Spatial Distribution Mechanism of Porosity Formation	Total Open Healed (case-by-case; Skeletal)
Thin Section (SEM QEMScan)	2	10^{-6} to 10^{-1}	10^{-3} to 10^{-2}	Spatial Distribution Mechanism of Porosity Formation	Total Open Healed (case-by-case; Skeletal)
Laser Scans	2 3	10^{-4} to 10^{-1}	10^{-3} to 10^{-1}	Pore Shape Pore Size Spatial Distribution	Total Open
Borehole					
Neutron Porosity (NPHI)	3	10^{-7} to 10^0	10^{-1} to 10^0	Magnitude	Total Water-filled (including bound water)
Density Porosity (DPHI)	3	10^{-6} to 10^0	10^{-1} to 10^0	Magnitude	Total Open
Sonic Porosity (SPHI)	3	10^{-4} to 10^{-2}	10^{-1} to 10^0	Magnitude	Total Open
Resistivity Porosity	3	10^{-4} to 10^{-2}	10^{-1} to 10^0	Magnitude	Total Open
NMR	3	10^{-7} to 10^{-2}	10^{-1} to 10^0	Magnitude	Total Open

Thin section point counts, thin section image analyses, and micro CT scans are the laboratory techniques used in this study. The following is a brief description of the remaining porosity-measuring techniques found in Table 1.2.1. In gas porosimetry, a pressurized gas is introduced into an airtight chamber that contains a rock sample (e.g., Constantz et al., 1995). When equilibrium between the gas chamber and the sample chamber is reached, the new pressure of the system, in conjunction with Boyle's law, is used to estimate the effective porosity of the sample. In fluid porosimetry, a non-wetting fluid (usually mercury) is introduced into an evacuated chamber containing a rock sample (e.g., Cook and Hover, 1999). Pressure on the fluid is gradually increased until it is forced into the pores of the sample. Effective porosity can therefore be estimated by measuring the pressures and intrusion volumes during the experiment. Nuclear Magnetic Resonance (NMR) uses a magnetic field to induce alignment of dipolar water molecules (e.g., Grunewald and Knight, 2011). Once the magnetic field is relaxed, water molecules rotate back out of alignment, leading to a change in the natural magnetic field that can be measured and attributed to pore filling water. Thin section QEMScans use a scanning electron microscope (SEM) in combination with an energy dispersive detector to acquire an x-ray spectrum at points along a rectangular grid superimposed on the sample (e.g., French et al., 2008). Both backscattered electron and energy dispersive x-ray signals are used to create digital images in which each pixel corresponds to a mineral species, or phase in a region under the electron beam (French et al., 2008). In this manner, open and healed porosity maps can be generated. Laser scans use a pinhole of light (laser source) to create a 2D image of the sample by scanning in the x- and y- directions. Because a laser scanning microscope's stage can move in the z- direction, 3D data can be generated

by analyzing 2D images at various depths. Open porosity can be identified with penetration depths usually on the order of tens to hundreds of microns (e.g., French et al., 2008).

In the case of borehole techniques, several geophysical proxies for porosity are exploited measuring the porosity of volumes of centimeters to meters as this is not the focus of this thesis, only brief descriptions are provided here. (Details are summarized in Hearst et al. (2000) and Mavko et al. (2009)). Neutron porosity logs use an active neutron source to probe the rock adjacent to the borehole. These neutrons interact with in-place formation water, which is taken to be a function of porosity. Density logs probe the rock with γ -rays that interact with the atoms of a formation, from which a measurement of the bulk density is derived. In this case porosity is determined by referencing the measured density to a model density. The deficit density is taken to indicate porosity. Sonic porosity is similarly derived from passing a sonic pulse into the formation. Since pores tend to slow the travel of the sonic wave, porosity is derived from the relative speed of the measured travel time of the sonic wave from a source to a receiver as compared to a model of sonic velocity. Resistivity porosity is estimated using active electrical soundings between a source and a receiver either by passing direct current or inducing current in the surrounding formation. In the simplest approach, rock is treated as a relative insulator and the fluid filling pores as a relative conductor. Archie's law is then used to estimate porosity. Borehole NMR works in the same fashion as described in the laboratory section above, though in this case in boreholes for *in situ* measurements rather than in a laboratory setting. These proxies are all susceptible to errors that violate the assumptions such as unusual mineral density, conductivity, or pore

filling fluids other than water (such as natural gas or oil) and environmental conditions such as borehole conditions, tool position, and temperature.

For laboratory techniques mentioned above, thin sections are a common means of 2D porosity analysis. In general, a thin section slice of a rock will never cut through the exact center of each pore, nor the grains that bound the pore, since the pore centers do not necessarily lie within a flat plane. The representation of pore volume and structure can be further complicated by anisotropy of the pore shape. The result is an apparent distribution of pore size, shape, and position (Krumbein, 1935). Figure 1.2.1 shows a representation of apparent pore size, shape, and position in a 2D thin section. Without any 3D porosity data, the true structure of these pores (represented as white shapes) is unknown. Figure 1.2.2 is a cartoon that further demonstrates difficulty associated with capturing the true nature of rock pores from two-dimensional sections. The orientation of the green slice represents the pores as more circular, whereas the orientation of the red slice represents the pores as much more elliptical. Since 2D porosity data rely on slices of a rock sample, they introduce a poorly characterized bias that hinders quantifying the volume, size, and shape of pores in a rock. This bias can be at least qualitatively defined in relation to the source of porosity. For instance, faulted rocks display anisotropic grain and pore geometry attributed to the fracturing process. The resulting porosity is therefore preferentially oriented and highly elongated in shape. To characterize these pores, thin-section orientations are chosen that are assumed to align with the principal axes of the anisotropy; however if those assumptions are not independently evaluated, they may introduce bias.

Direct 3D methods can be used in an attempt to overcome the bias associated with 2D porosity measurements. Initially, 3D data in the geosciences were generated by destructive and tedious methods. The processes involved drawing or scanning numerous images of 2D slices, each generated after a few microns of material had been sequentially ground away (e.g., Fourie, 1974; Kretz, 1993), thereby destroying the specimen during the process. In the early 1980s a nondestructive means of obtaining 3D data was applied to the geosciences. Micro CT scans helped to reveal pore shape, size, and magnitude, as well as the spatial distribution of those pores, all while keeping the specimen intact during and after analysis. The first application of micro CT scans to the geosciences was used to image matrix-filled fossil skulls (Conroy and Vannier, 1984) and quickly grew from there. As the specifications of the technique in the medical field advanced, so too did its application to the geosciences. For example, micro CT imaging has since been used to study one-of-a-kind meteorite specimens (Arnold et al., 1982), to image fluid flow experiments (Anderson et al., 1992; Heijs et al., 1995), to characterize soil and pore-space morphology (Peyton et al., 1992; Zeng et al., 1996), and to employ real-time analysis of scale models of faulting (Schreurs and Hanni, 1998).

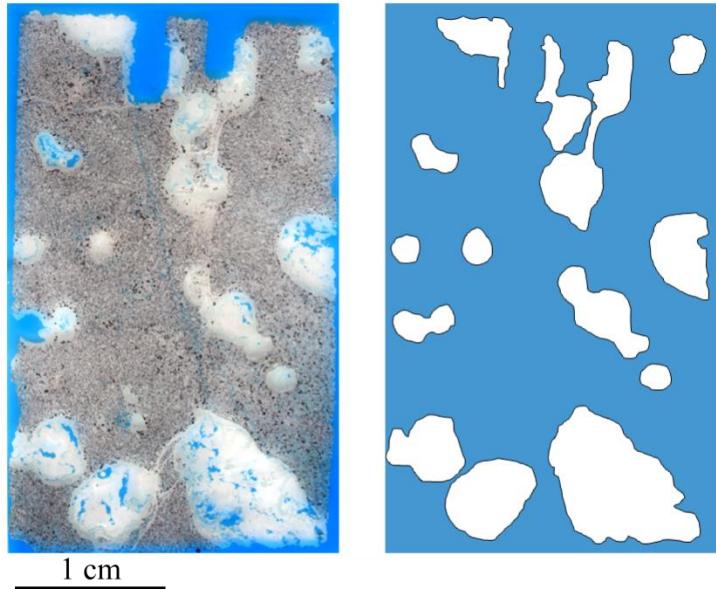


Figure 1.2.1: Thin section 4303 PFWA and generalized porosity map. The structure of the pores (seen in white) is derived from a 2D slice of a rock and therefore represents an apparent distribution of pore shapes and sizes.

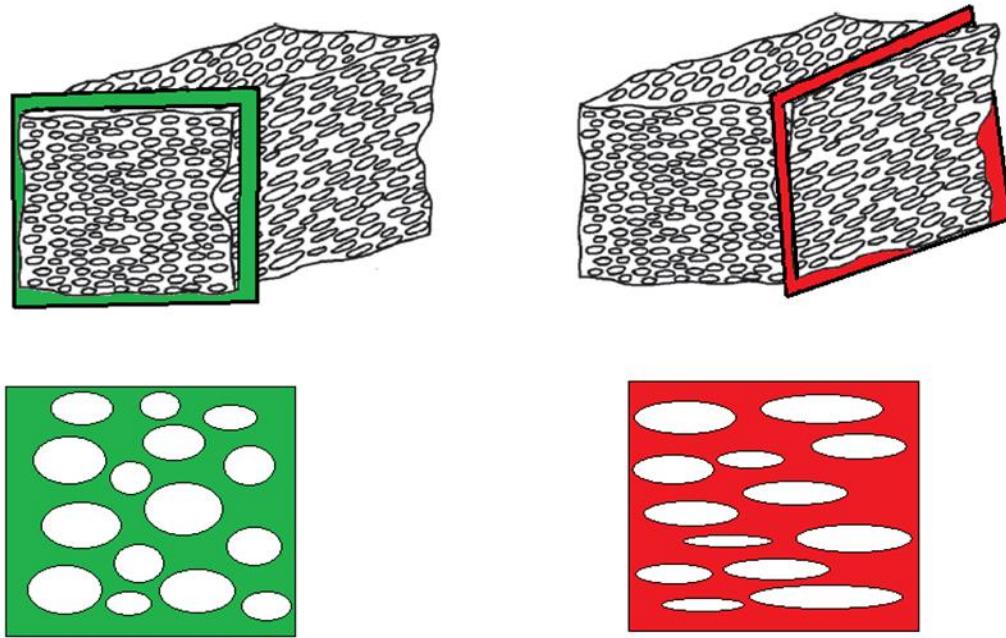


Figure 1.2.2: Cartoon showing different orientations of a 2D slice of rock. This demonstrates the difficulty associated with capturing the true nature of rock pores as a result of 2D orientation. The green slice (left) represents the pores to be more circular, whereas the red slice (right) represents the pores to be much more elliptical. Modified

from Rowland et al. (2007).

1.2.2 Fracture Roughness

Fracture surface roughness can be defined in terms of the topographical relief of a fracture surface relative to a flat line or plane fit to the surface. Roughness can be measured across different length- scales through use of surveying equipment, contact profilometers, LIDAR, and more recently, two- and three- dimensional imaging of thin sections and sub-cores. The character of surface roughness is important to geologists for a number of reasons. The size, shape, and curvature of asperities in contact across the surfaces affect the stiffness and strength of the fracture (Brown and Scholz, 1985, 1986; Barton, 1986), influencing the bulk mechanical properties of the rock mass (e.g., Power and Tullis, 1992; Jaeger et al., 2007; Barton, 2007). The topography of the fracture surface and mismatch of asperities on the faces of the fracture provide a mechanism to prop the fracture open and maintain open porosity. During slip, the shape of such asperities and their strength influence dilation potential as asperities are forced to pass and ride up over each other or break. Such surface textures can be modified by secondary processes such as grinding, mineral precipitation, dissolution, mineral alteration, and repeated opening or slip that modify the surface topography, alter the strength of asperities, and which can develop cohesion between the fracture surfaces (Crawford et al., 2002). The size and complexity of the apertures between the surfaces affects the ability of a fracture to transport fluids (Long et al., 1985; Brown, 1987), thus controlling the transport properties of the fracture, and both the permeability and storativity of the rock mass in otherwise low porosity rock. High porosity in slipped fractures (and therefore the potential for natural or stimulated permeability) can be

maintained when the surface roughness is high enough, in combination with sufficient strength, to actually prop the two fracture surfaces apart during slip (Committee on Fracture Characterization and Fluid Flow, National Research Council, 1996).

1.2.3 Definition of Terms

In general, *porosity* refers to the open space within a rock volume. Varying geologic conditions, however, can give rise to different types of porosity, and if the rock is crushed, can even lead to porosity loss (Paterson and Wong, 2005). *Primary porosity* refers to porosity attributed to the formation of the rock (i.e., intergranular pores, vugs and vesicles). *Secondary porosity* refers to any porosity generated after the rock has formed due to chemical processes such as dissolution of minerals, or mechanical processes such as fracturing. *Open porosity* refers to the open space that currently exists in a rock. *Healed porosity* refers to porosity that has been filled-in with secondary minerals since its formation. Lastly, *skeletal porosity* refers to the sum of both the open and healed porosities in a rock (Figure 1.2.3), and provides some measure of the accumulation of porosity in the rock through time. In each of these cases, the type of porosity refers only to the volume of the rock occupied by open or formerly open space. Although open pores can store fluids, their connectedness is a separate property that determines if they can allow fluid transmission across the sample. This type of porosity is the *effective porosity*. The minimum width of open-space between two larger connected pores is termed the *pore throat*. *Pore structure* is characterized by, but not limited to, the following characteristics: pore connectivity, pore position, pore ellipticity, and pore tortuosity, that is, the path that fluids may take throughout the pore space.

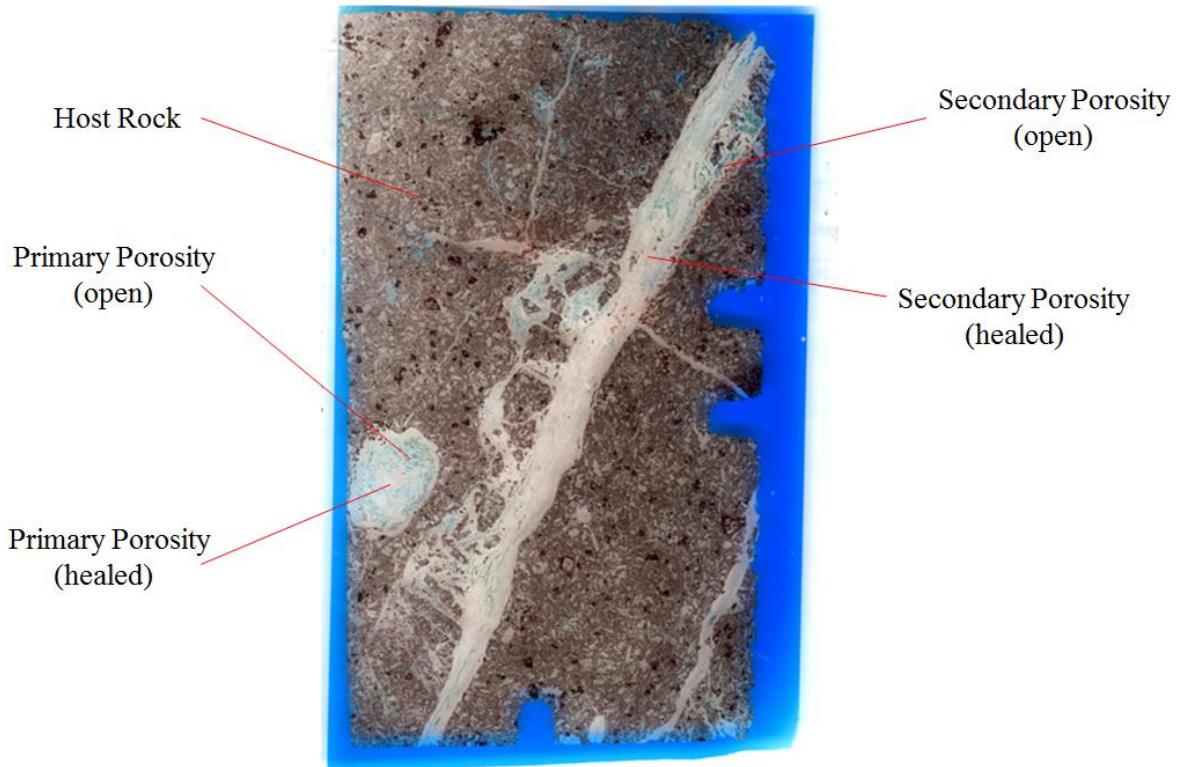


Figure 1.2.3: Thin section displaying the various types of porosity. Thin section dimensions are 20 x 30 mm. The single- and double- notched sides signify the formation dip and up-well direction, respectively.

1.3 Geologic Setting

Newberry Volcano is located in Oregon, approximately 60 km east of the north-south trending crest of the Cascade Range (Figure 1.3.1) (Bargar and Keith, 1999). Newberry is the largest volcanic edifice in the contiguous United States and contains a wide range of volcanically derived rock types, ranging from common basaltic and andesitic lava flows, to silicic lavas and rhyolite, as well as associated intruded dikes (Jensen et al., 2009). Also present are pyroclastic flow deposits occurring with tephras and other sediments, obsidian flows, ash flow tuffs, and pumice and lapilli beds (Jensen

et al., 2009). The volcanism associated with Newberry is attributed to the Cascades, and has occurred as recently as 1300 years ago (Fitterman, 1988). Paleomagnetic studies (MacLeod and Sammel, 1982) of core taken from the volcano show only normal polarities, suggesting the volcano itself is younger than 0.7 Ma. Figure 1.3.2 shows a cross section of the Newberry volcano. The edifice is comprised of intrusive rocks, the John Day Formation, the Newberry/Deschutes Formation, and various intracaldera units of which the lithology has been mentioned above. The core used in this study is obtained from the Newberry/Deschutes Formation (Figure 1.3.2).

1.3.1 Structure of Newberry Volcano

Newberry's shield-like structure is the result of extensional features of the Basin and Range province (Fitterman, 1988). The rocks' several stages of deformation can be attributed to the region's volcanism and the three regional fault systems that converge at the volcanic edifice (Fitterman, 1988). Three extensional faults, as well as various intrusives and feeder dikes attributed to the region's volcanism, cross-cut the John Day and Newberry/Deschutes Formations (Figure 1.3.2). A series of ring fractures associated with caldera collapses also surrounds the outer edges of the Newberry Volcano. These collapses are thought to have occurred between 0.3 and 0.5 Ma and are attributed to two large ash-flow tuff eruptions (Fitterman, 1988; MacLeod and Sammel, 1982). The fault systems that converge in the region have different orientations. Basin and Range normal faults trend NNE, Cascades Graben normal faults are N-trending, and the Brother's Fault Zone trends WNW (Figure 1.3.3) (Cladouhos et al., 2011). These different units contain varying amounts of open, healed, and skeletal porosity that relate to the recurrent fracture

slip in the area (Fetterman and Davatzes, 2011).

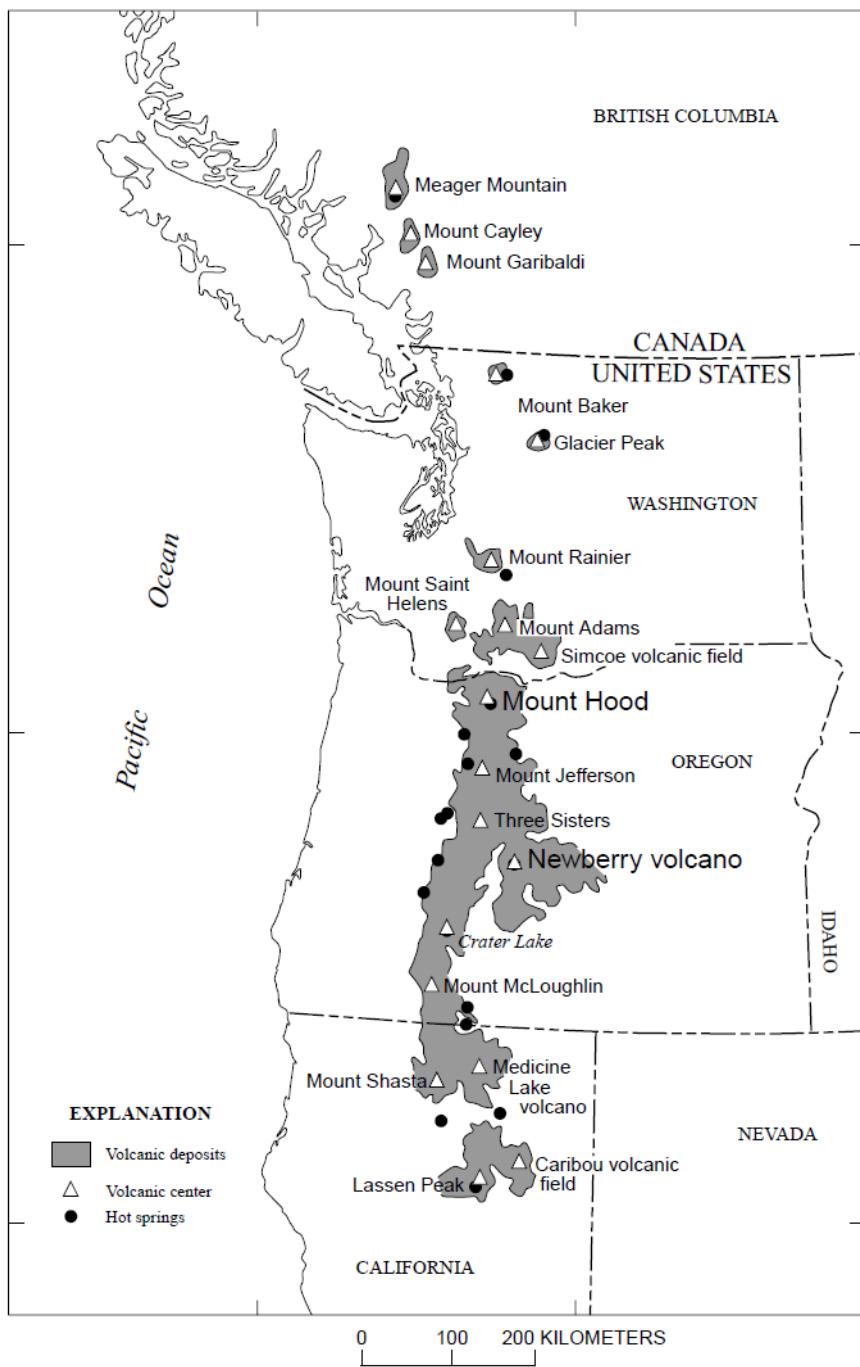


Figure 1.3.1: Regional view of Newberry Volcano showing the effects of Cascade volcanism. Newberry Volcano is circled in red. Modified from Bargar and Keith (1999).

EGS 55-29

GEO N-2

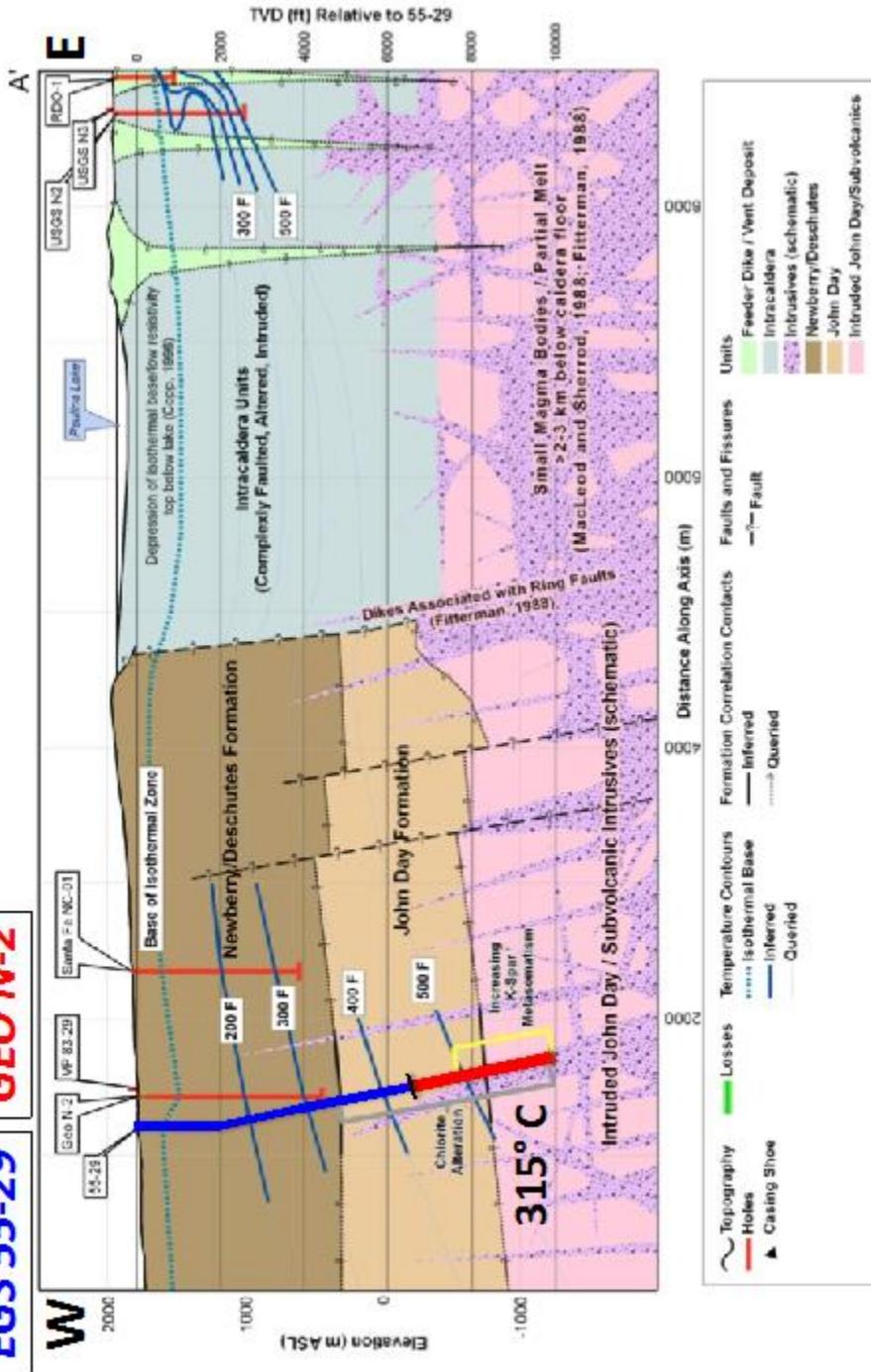


Figure 1.3.2: Cross section showing the major stratigraphic units, structural boundaries, and geothermal temperature profile. The GEO N-2 core hole is in the west flank of the Newberry Volcano and outside the zone of hydrothermal activity, which is largely confined to the collapsed caldera. The EGS demonstration well, NWG 55-29, is located approximately 0.5 km to the west. Modified from the Newberry Demonstration Project

Phase I, Stage Gate Report (AltaRock, 2011) and Sonnenthal et al. (2012).



Figure 1.3.3: Satellite view showing the faulting regime at Newberry Volcano. Fault systems are color-coded by age. Orange faults represent Holocene to Pleistocene, green faults represent late-Quaternary, blue faults represent mid-Quaternary, and purple faults represent unspecified Quaternary. The yellow circle and red diamond represent the Geo-

N2 and NWG 55-29 geothermal wells, respectively. Modified from Cladouhos et al. (2011).

1.3.2 Geothermal Potential of Newberry Volcano

The bedrock at Newberry Volcano is subject to a high, conductive temperature gradient, thus reaching high temperatures at relatively shallow depth. Figure 1.3.2 shows the variation in temperature as a function of depth (blue temperature contours). Thus, the region has the potential to be geothermally viable, but although naturally fractured (Davatzes and Hickman, 2011), the bedrock surrounding the central caldera has low permeability. As early as 1976, twenty cores were drilled at Newberry (Figure 1.3.4) to study the area's geothermal potential (Olmstead and Wermiel, 1988). Currently, a multi-million dollar Enhanced Geothermal Systems (EGS) demonstration project is underway by AltaRock supported by private investment and the Department of Energy (DOE) in an attempt to stimulate fractures and improve the permeability of the subsurface (Cladouhos et al., 2012). The Northwest Geothermal (NWG) well 55-29, in particular, spans 10,060 ft (\sim 3066 m) in depth and is the well directly associated with the EGS project. Following a long period of pre-stimulation characterization, the first step in the stimulation of EGS well NWG 55-29 was implemented in October 2012 (Cladouhos et al., 2013; Petty et al., 2013). The core used in this study comes from the Geo-N2 well, located \sim 0.5 km east of the NWG 55-29 stimulation well (Figure 1.3.2).

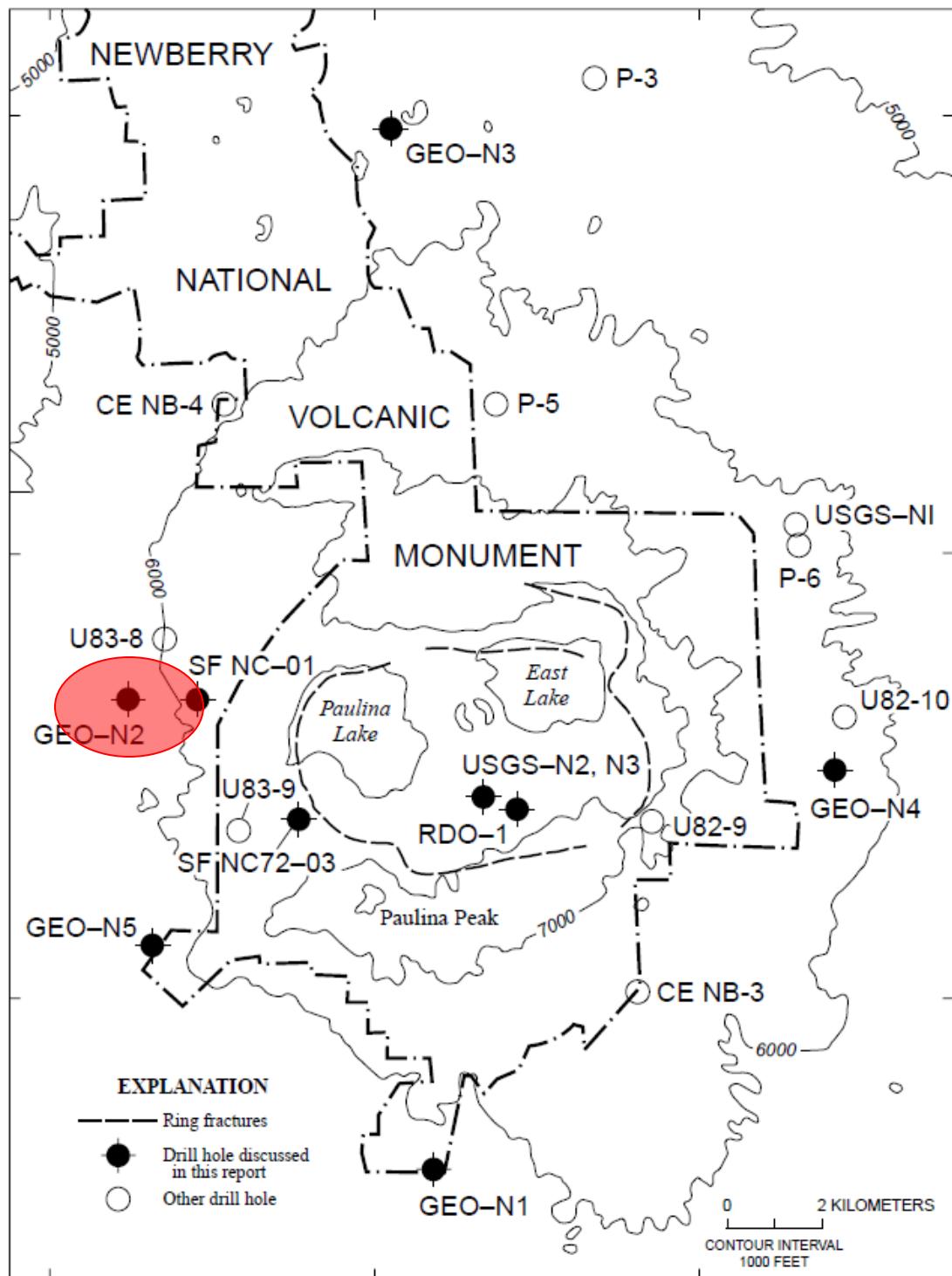


Figure 1.3.4: Topographic map of Newberry National Volcanic Monument. Map shows drill holes associated with the geothermal investigation. The Geo-N2 well is circled in red. Modified from Bargar and Keith (1999).

1.3.3 Lithology of the Geo-N2 Well

Drilled in 1986, the Geo-N2 well is the deepest core in Newberry, reaching a depth of 4,386 ft (1,337 m) and consisting of basaltic to rhyolitic lava flows with intervening flow breccias, lithic tuff, and volcanic sandstone (Bargar and Keith, 1999). At a depth of 3,891 ft (~1,186 m) at least one thin basaltic dike intrudes a vesicular mafic lava flow. As confirmed by powder x-ray diffraction (XRD) and petrography, the mineralogy of the core is dominated by plagioclase; also present are magnetite, clinopyroxene, quartz, potassium feldspar, and chalcopyrite (Bargar and Keith, 1999; Fetterman and Davatzes, 2011). Many primary and secondary features are contained within the rock, including vesicles (spherical to elliptical) and fractures, respectively (Bargar and Keith, 1999). Phenocrysts of plagioclase are also abundant. Pore-filling minerals are dominantly chalcedony, opal, quartz, and calcite, with minor amounts of chlorite (Fetterman and Davatzes, 2011; Bargar and Keith, 1999). Because the well is located only ~0.5 km away from the NWG 55-29 stimulation well, the rocks in this study are representative of the rocks that will be stimulated for the EGS project. Both wells contain the same rock types and have undergone the same stresses and deformation history.

CHAPTER 2

METHODS

2.1 Overview

To investigate both the modern porosity and its evolution through geologic time, the current open porosity, the healed porosity, and the total skeletal porosity structures are carefully distinguished. 2D image analyses and point counts of thin sections, as well as 3D measurements are integrated to (1) evaluate the differences in porosity estimated through these techniques, including sensitivity to the types of pores associated with different stages of fracture development, and (2) interpret the process of porosity creation and destruction in the volcanic rocks at Newberry. Whereas the connected porosity is critical for fluid flow and might be measureable in core plugs through porosimetry techniques, the total porosity, including isolated pores, is critical for mechanical failure and accompanying dilation relevant to EGS applications. 2D methods lack the ability to assess such connections, but the three dimensionality of the pore structure can be investigated using 3D micro CT imaging. In addition, the healed pore structure provides insights into the *in situ* accumulation and loss of porosity due to dilation, dissolution, and healing. Both 2D and 3D methods are applied to assess the history of fracture healing through mineral precipitation.

First, measurements of the porosity types were obtained using point counts of thin sections and automatically thresholded images of thin sections to explore differences, advantages, and disadvantages between the two techniques. Second, the pore structure, within sub-regions of the thin sections, was carefully mapped in high resolution images having pixel dimensions of $\sim 1.04 \mu\text{m}$. This approach provides both higher resolution

imaging and critical qualitative contextual information on the mineral structure, mineral dissolution and replacement textures, and consequently the history of healing in the sample. Third, the pore structure was investigated using non-invasive, 3D micro CT imaging with a resolution of $\sim 26.7 \mu\text{m}$. The 3D measurement technique should have the potential to better characterize anisotropic porosity structure that is expected of fractured rock volumes. Finally, the differences in porosity measurements between these techniques, including the dependence of these measurements on the relative maturity of the fracture, were evaluated. This data set, along with measurements of fracture roughness, also provides the opportunity to assesses how the potential for dilation accompanying repeated slip depends on the fracture stage and fracture characteristics such as the roughness of the fracture walls. Fracture roughness measurements were obtained by digitizing the topography of the fracture surfaces seen in a scanned image of a 2D thin section. Petrographic analysis and cathodoluminescence were used to aid in distinguishing surfaces of past healing and slip events within the fracture based on subtle differences in mineral texture, layer continuity, and luminescence due to cathode ray excitation, respectively. For a checklist of methods applied to each sample, see Appendix B.

Samples spanning the depths of 3,523 to 4,306 ft ($\sim 1,074$ to $\sim 1,312$ m) measured depth that include natural fractures and surrounding adjacent sections of unfractured/undisturbed rock were studied. This configuration enables assessment of the characteristics of the initial, primary porosity and its subsequent modification by fracturing. To isolate the ability of the different techniques to measure fracture porosity and to assess the impact of fracture formation and repeated slip on the development of

secondary porosity, the samples analyzed were restricted to andesite and basalt from a narrow depth range in the GEO-N2 corehole. In the GEO-N2 well, these rock types have similar primary porosity structure and grain size characteristics as well as only a narrow range of mineral composition. The narrow range in depth from 3617 to 4306 feet (~1,074 to ~1,312 m) makes the stress conditions experienced by these natural fractures as similar as practically possible given the core available.

These samples comprise the deepest core available in the Newberry Geothermal system and the closest to the stimulation well 55-29. Nine samples were chosen to represent fractures at five different stages of development (see discussion by Fetterman and Davatzes, 2011) as assessed through the accumulation of fault rocks (rock so altered by deformation that the original primary host rock characteristics have been lost). *Stage 1* represents a rock that is a controlled, unfractured and cohesive mass. *Stage 2* represents a rock that contains a single fracture, or a macroscopic structure representing a discontinuity in the rock defined by two distinct surfaces and lacking fault rock or closely spaced, linked macroscopic fractures. *Stage 3* represents a rock that contains linked fractures, or a set of interconnected but mostly parallel fractures forming a network that still lack fault rock in the host rock. Fault rock and/or vein material between the two contact surfaces might be slickensided. *Stage 4* represents a rock that contains proto-breccia. Macroscopic splay fractures begin forming in this stage and large, isolated rock clasts are created adjacent to the central fault zone. Lastly, *stage 5* represents rock that has developed fault breccia, or macroscopic features that form a network of interconnecting cracks and breccia. Figure 2.1.1 shows a schematic of the components of a fully developed fault. Orange represents the fault core, consisting of fault gouge,

cataclasite, and fault breccia. Green represents the damage zone associated with the fault, as will be discussed in detail in subsequent sections of this paper. The damage zones at Newberry consist of small faults, fractures, veins, and folds as a result of the deformation process. Grey represents the undisturbed protolith, containing regional structures of the host rock.

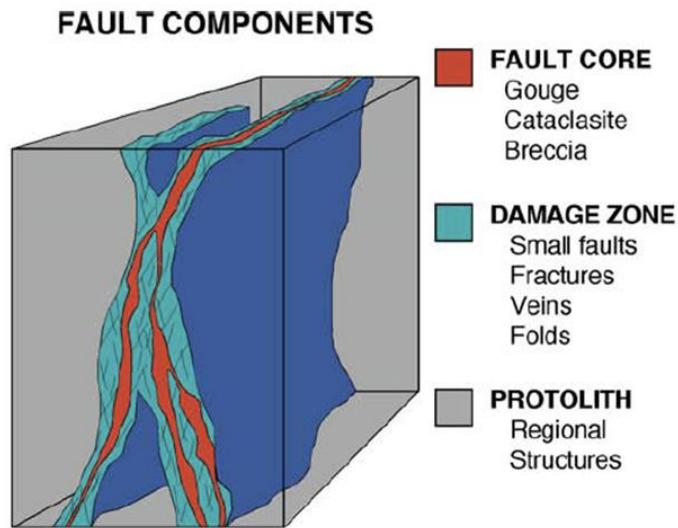


Figure 2.1.1: Components of a fault. Modified from Caine et al. (1996).

Nine samples provided a balanced number of examples allowing the dependence on fault stage and the repeatability of pore characteristics to be assessed as well as allowing complete suites of analyses to be completed in a reasonable period of time. Table 2.1.1 shows the sample thin sections and associated cores along with their classification (Fracture, Footwall, etc.) and fracture development stage (1, 2, 3, etc.).

Table 2.1.1

Summary of thin section/core classification and fracture stage

Stage:	1	2	3	3 to 4	4	4	4	5
Fracture								
Footwall (A)								
Footwall (B)								
Sample:	N2-3617	N2-3937	N2-4152	N2-4267	N2-3523.5	N2-4302	N2-4303	N2-4306

2.2 Sample Preparation

For 2D analyses of both porosity and fracture roughness, 20 x 30 mm thin sections were cut from the Newberry core and impregnated with blue epoxy so that any open pore space could be easily distinguished from the surrounding materials. For 3D porosity analyses, adjacent sub-cores were cut from the same location of the thin sections using a rock saw and drill press. Both 2D and 3D samples were obtained from the center of the cores so that they were furthest from disturbance and thermal shock expected from drilling and extraction to the surface. The sub-cores had dimensions no greater than 2.5 cm in diameter and 6 cm in height. Figure 2.2.1 shows a cartoon illustrating the general location of thin section cuts and sub-core locations. Despite being adjacent to the thin section, the sub-cores encapsulate the same features as the thin sections, allowing for useful comparisons of patterns in porosity across the samples.

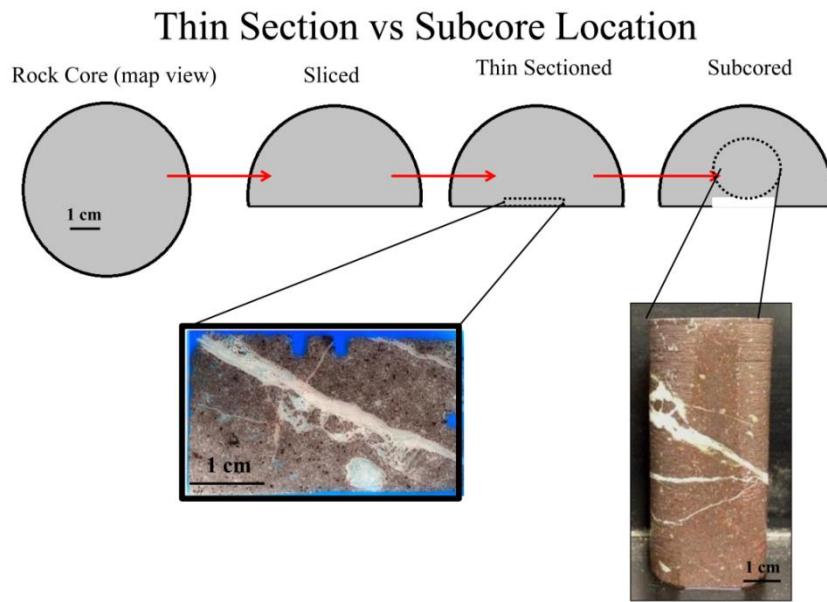


Figure 2.2.1: Schematic showing the relative locations of thin section slices and subcores. Notice the sub-core location is adjacent to the thin section location, but still manages to capture the same feature (in this case a fracture) of interest.

2.3 Point Counts

Point counts were conducted on each thin section using a Nikon Eclipse LV 100 petrographic microscopic in combination with a Pelcon Automatic Point Counter. The point counter not only allowed for automatic movement along the thin section, but also kept track of the designated phase (category) and Cartesian coordinates of each point, allowing the construction of porosity maps. Each observation point was classified as either: (1) groundmass (original rock), (2) healed porosity, or (3) open porosity. For the samples used in this study, the blue color of the impregnated epoxy clearly identifies the open porosity. Healed porosity, whether from crack fill or dissolution re-precipitation, is interpreted from mineralogy, texture, and relative age relationships.

While different recommendations exist for the desired number of point counts necessary to maintain precise measurements (e.g., Plas and Tobi, 1965), most recommendations deal with sedimentary rock matrix or pores and fail to take into account the effect of fractures within an igneous rock matrix. The point count density in this study, therefore, was checked for quality assurance by continuously sub-selecting a very dense dataset until the porosity result began to vary. An initial count of 1000 points was conducted on a thin section and the percentage of different porosity phases were plotted (Figure 2.3.1). Subsets of this data were then sampled and porosity was calculated, providing a map of the repeatability of the porosity measurement as a function of the number of point counts. The fewest number of counts that yield results that show no substantial error from the maximum count density of 1000 constitutes the preferred number of counts because it provides a repeatable measurement of each of the phases distinguished with the least effort and time (Figure 2.3.1), while guaranteeing accuracy when compared to the maximum count results. For point counts in this study, the required representative point count density was 700, as signified by the red arrow in Figure 2.3.1. 700 point counts is exceptionally high when compared to other studies, but is acceptable in this study because (1) each method compared is maximized to the highest resolution settings and (2) use of an automatic point counter increases the speed of the point count. Any error seen with this high of a count density therefore, is negligible.

Once the thin sections were point counted, the data were uploaded into Microsoft Excel where the percentages of each category could be obtained and a porosity map could be generated. Relative porosities were obtained by dividing the number of counts of each

porosity type by the total number of counts for the entire thin section. Porosity maps were then created in Microsoft Excel by color-coding each phase and plotting the data against the Cartesian coordinates for each point for comparison to other thin section and micro CT maps.

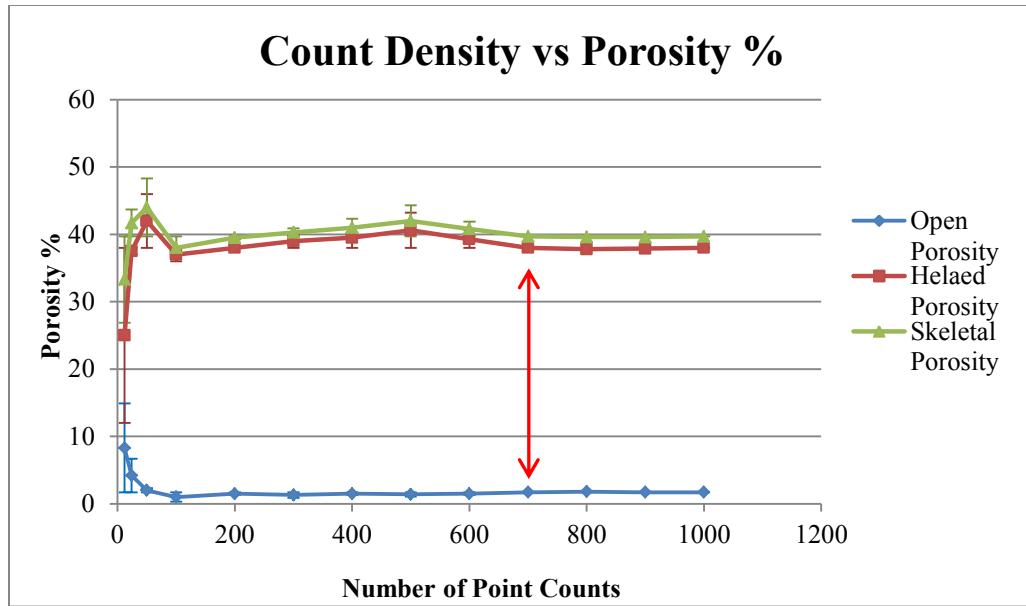


Figure 2.3.1: Plot of number of point counts vs. porosity percent. At densities less than 700 point counts (red arrow) the skeletal porosity begins to vary significantly, as represented by error bars.

2.4 Automated Image Analysis

Scans of thin sections were obtained using a flatbed scanner. The resulting TIF files were imported into MATLAB[®] for automated analysis. Each pixel in the images is 20.7 μm and contains combinations of RGB color values between 0 and 255. Pixels correlating to the colors of the range of blue epoxy and to the range of colors indicating

the healing minerals were isolated for analysis. For each thin section, the lower and upper limit of the range of these colors was manually determined by a trial-and-error process that ended when a satisfactory porosity map was generated. In general, the blue of the open pore space is unequivocal; however the color ranges corresponding to the minerals healing the fracture can be non-unique on a sample-by-sample basis. For instance, plagioclase, abundant in these rocks as a primary rock-forming mineral, and unstained in the thin sections analyzed in this study, can share a similar range of white with the healed minerals from Newberry (predominately chalcedony, opal, quartz, and calcite). Consequently, it was avoided as much as possible when determining the threshold range.

A MATLAB[®] algorithm was written which automatically imported the thin section images and thresholded for the desired color ranges of both open and healed porosity for each sample via user input. Appendix A contains the MATLAB[®] script used with this method. To determine the relative abundances of open and healed porosity, the algorithm divided the number of pixels selected for each category by the total number of pixels in the image. Similarly, a porosity map was generated by replacing the thresholded pixels representing open and healed porosity with solid colors of blue and green, respectively.

2.5 High-Resolution Image Analysis

Photomicrographs of thin sections magnified at 4x were taken in plane-polarized light using a Nikon Eclipse LV 100 petrographic microscopic in combination with a high-resolution digital camera attachment. Each picture contained approximately 2 x 2.5

mm of thin section area and had a resolution of 2560 x 1920 pixels and a pixel size of ~1.04 µm. Overlapping pictures were taken in 1 mm increments to form transects across fractures that included undisturbed host rock on either side (sometimes requiring multiple sections) with the aid of a thin section stage mount. The photographed transect location was chosen such that it encompassed varying porosity types (Figure 1.2.3) as well as at least one of the following: host rock, damage zone, and/or fracture. The series of photographs were then stitched together using Adobe® Photoshop® and saved as TIF files. Once the transects were created, a combination of the software Adobe® Photoshop®, ImageJ, and MATLAB® were used to generate high-resolution porosity maps, determine the relative porosity abundances, and create a moving window that calculates fractional porosity along increments of the transect at different starting positions. Appendix A contains the MATLAB® scripts used with this method.

2.5.1 Thresholding High-Resolution Images

The TIF images of magnified thin section transects were uploaded into Adobe® Photoshop® in order to isolate pixels that represented open and healed porosity (Figure 2.5.1a, b). Again the impregnated blue epoxy was represented by pixels containing open porosity, which could be automatically selected using tools in Photoshop® to create a blue-color overlay that represents a high-resolution map of open porosity. In plane-polarized light, mineral colors that represent healed porosity do not strongly contrast the mineral color of the host rock because they each share many of the same shades of grey, so healing minerals were manually thresholded and mapped in detail. These interpretations were benchmarked petrographically for quality assurance. Thus, the

Photoshop® tools, in combination with petrographic data and researcher-decision, generated a high-resolution map of the pixels that correlate directly to healed porosity. Once completed, a green color overlay was constructed that represents a high-resolution map of healed porosity in the image. These overlays provide the basis for binary image maps of the open and healed porosity (using either Photoshop® or ImageJ), which can be subsequently analyzed via scripts in MATLAB®, as discussed in the following section (Figure 2.5.1c, d).

2.5.2 Determining Porosity Amounts

To evaluate the amount of porosity present in the sample, as well as spatial variation in pore structure and size, the fractional porosity was evaluated within the whole image and within image sub-regions. To determine the relative amounts of open and healed porosity within the entire photo transect, the porosity maps were first made binary in the photo-editing software ImageJ. A MATLAB® algorithm (Appendix A) was then written that uploaded the binary maps and divided the number of black pixels in the binary porosity map by the total number of pixels in the image, yielding the fractional porosity of the entire transect. This method was applied to both open and healed porosity.

To evaluate porosity at points along the transect, a MATLAB® algorithm (Appendix A) was written that established a moving window of set size that traversed the length of the porosity maps. This moving window measures porosity along a transect across the image. The window size for these samples needed to be small enough to not overlap the outer bounds of the images, but big enough to capture a sufficient number of

pixels. To investigate the window size appropriate for this study, a window sensitivity analysis was performed. This method of sensitivity analysis described below determined an appropriate window size that generated results for this method. Figure 2.5.2 details the window sensitivity analysis. The centered transect ran parallel to the z-axis along the open and healed porosity maps generated from the stitched thin section images (Figure 2.5.2a, b, and c). In Figure 2.5.2c, two window sizes (red being 0.01 cm and black being 0.05 cm) were initially chosen to investigate healed porosity representation as a function of window size. The numbers next to each window represent the number of pixels that lie within the respective window sizes. When the window size is 0.05 cm (Figure 2.5.2d) instead of 0.01 cm (Figure 2.5.2e), the fine details of healed porosity across the fracture are lost, however so too were the number of pixels being analyzed, dropping from 230,880 pixels to 9,216 pixels and decreasing the resolution of the measurement. To investigate window sensitivity in more detail, a MATLAB[®] algorithm (Appendix A) was written that plotted a summary of the variation in the profile of healed porosity for different sized windows (Figure 2.5.2f), box plots illustrating the variation in porosity within the family of windows for each window size tested (Figure 2.5.2g), and the mean and standard deviation of porosity along the sample profile as a function of the window size (Figure 2.5.2h). The box plots in Figure 2.5.2g show that as window size increases, the median of the data begins to stabilize and outliers decrease. With very small window sizes, the box plots usually hit or miss capturing porosity measurements, whereas the largest window sizes are rarely porosity-free. Figure 2.5.2h shows that as window size increases, the standard deviation and mean decrease. Because this study needs (1) a sufficient number of pixels in each window (generally much greater than pixel size), (2) a

window that does not lie outside the bounds of the image data, and (3) a window size that can be directly compared to a three dimensional cube size (as discussed in the next method), the desired window size and pixels captured were chosen to be 0.1 x 0.1 cm and 922,560 pixels, respectively.

Once the window size was established, the MATLAB[®] algorithm moved the window in set-increments along the transect, calculating porosity at each point by dividing the number of colored pixels by the total number of pixels contained in the window. The fractional porosity at each step was then plotted against position along the transect, revealing the variation of porosity in the vicinity of a fracture, including damage adjacent to the fracture and the current open and healed porosity of the fracture itself (Figure 2.5.1e). Figure 2.5.3 is a workflow diagram that conceptualizes the steps necessary for this method, showing visual results of stitching images, thresholding porosity, making images binary, and analyzing transect porosity.

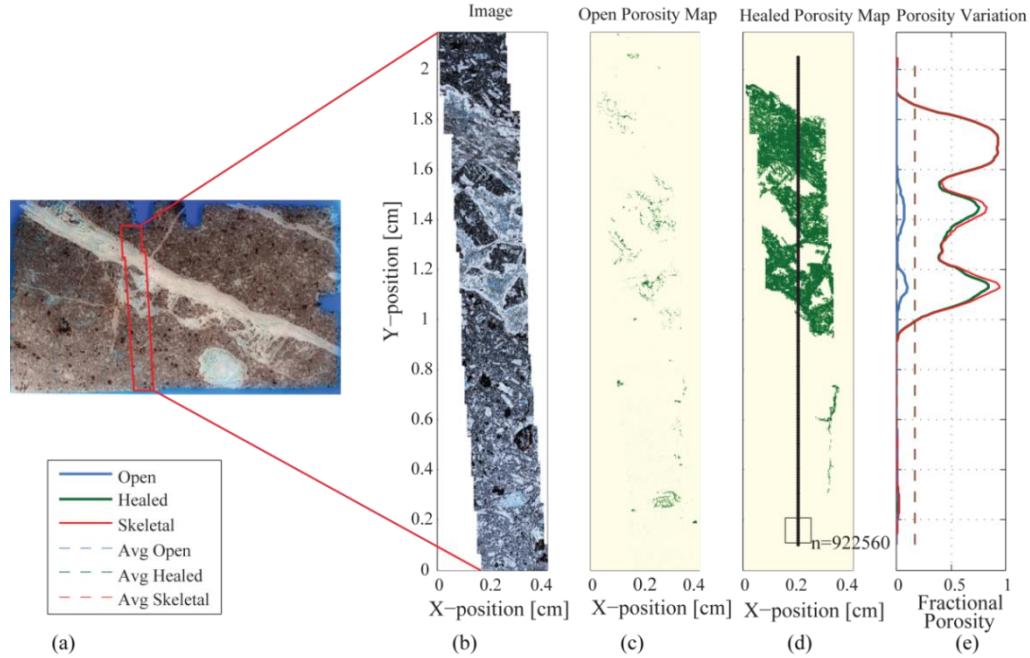


Figure 2.5.1: Summary of high resolution image analysis. (a) Thin section scan (30mm in length); (b) High resolution thin section photo montage in plane-polarized light with

~1 micron pixel resolution at 4x magnification; (c) Map of open porosity; (d) Map of healed porosity; (e) Plot of open, healed, and skeletal porosity along transect of window size shown in (d).

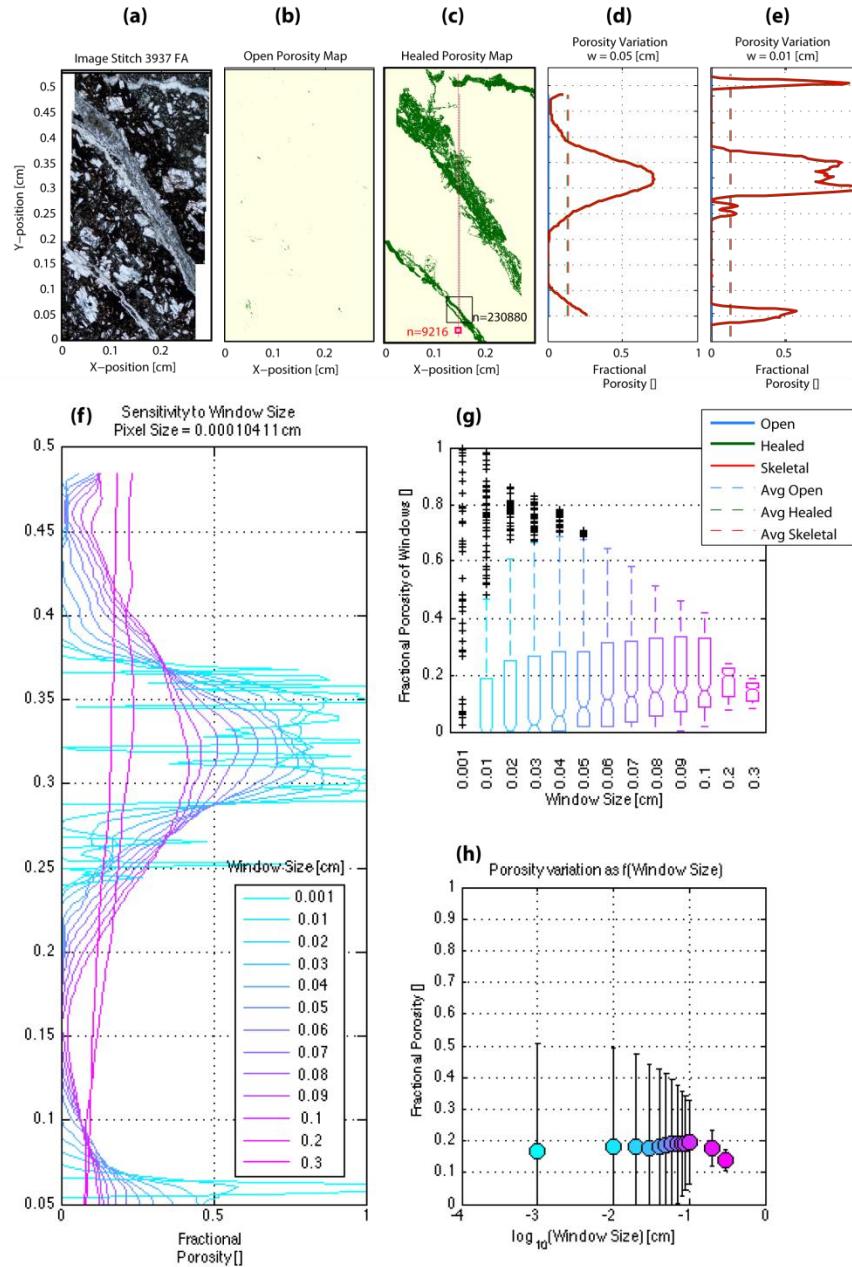


Figure 2.5.2: Window sensitivity analysis for high resolution images. (a) High resolution thin section image, (b) binary open porosity map, (c) binary healed porosity map, (d) transect parallel to z-axis with sample window 0.05 cm, and (e) transect parallel to the z-

axis with sample window 0.01 cm. Note that the sample positions and window size for each transect are illustrated in (c) including the number of pixels that lie within the respective windows. (f) Summary of the variation in the profile of the healed porosity for different sized windows. (g) Box plots illustrating the variation in porosity within the family of windows for each window size tested. (h) mean and standard deviation of porosity along the sample profile as a function of the window size.

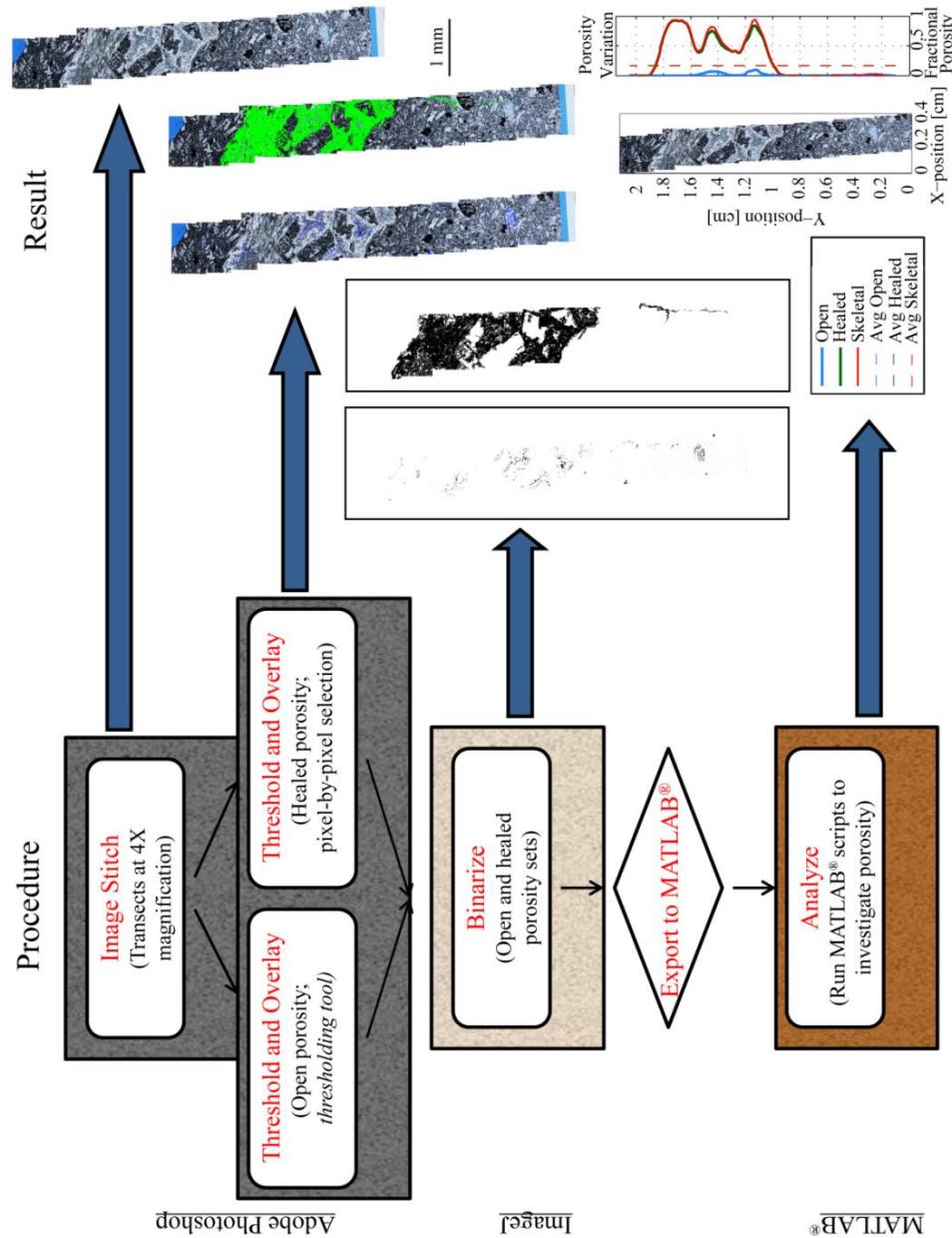


Figure 2.5.3: Workflow diagram summarizing the workflow of the methods and results for high resolution image analysis.

2.6 3D Micro Computed Tomography Scans

Micro Computed Tomography (CT) scanning allows for the visualization of a three-dimensional porosity structure that would otherwise be omitted from traditional two-dimensional techniques. This method passes x-rays through samples at different angles as the samples rotate in the machine (Figure 2.6.1). The x-rays attenuate as a function of density and create thousands of grayscale horizontal 2D slices that comprise the 3D image. The grayscale color values represent the attenuation of the x-rays, and thus correlate to different densities in the sample.

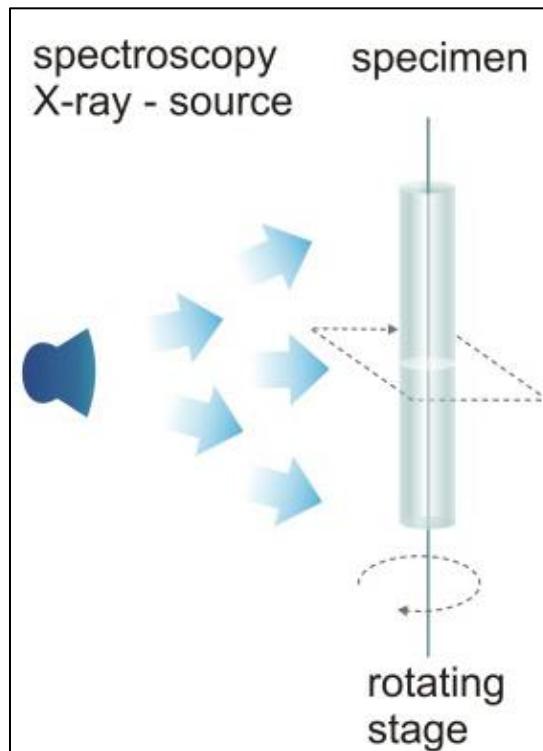


Figure 2.6.1: Schematic of how x-rays pass through a specimen during a micro CT scan. Modified from Jorgensen et al., 1998.

By thresholding the x-ray attenuation for open space and specific minerals of known density, both open porosity and healed porosity can be identified.

To obtain three-dimensional porosity data, Newberry sub-cores (Figure 2.2.1) were scanned using the Temple University School of Medicine's SkyScan 1172 high-resolution micro CT scanner (Figure 2.6.2). Scans were taken with aluminum and copper filters at the machine's highest resolution ($26.7 \mu\text{m}$) and power (100 kV and 100 μA) settings in an attempt to resolve the smallest grains possible allowed by this method and to account for the large attenuation in rock.



Figure 2.6.2: Image of the Temple University School of Medicine's SkyScan 1172 high-resolution micro CT scanner.

2.6.1 Reconstruction and Post-processing

Upon scan completion, which can take up to 4 hours to reach, the 3D images were reconstructed using the program NRecon. Reconstruction is the mathematical process of converting sonograms, or raw scan data, into thousands of two-dimensional slice bitmap images (Ketcham and Carlson, 2001). These images are comprised of $26.7 \mu\text{m}$ -sized pixels, each with grayscale values correlating to a tomographic inversion for x-ray attenuation (which correlates with material density). The program CTan was then used to post-process the data. CTan has numerous tools for post-processing the bitmaps that allow: (1) correlating attenuation to density, and (2) isolating attenuation or density values to produce 3D porosity maps. The *thresholding* tool allows for pixels of desired grayscale values to be selected, made binary, and isolated from the rest of the image. Materials in the samples that have a higher density than most of the host rock and a larger, more easily identifiable grain size (*i.e.* calcite) are represented by pixels with grayscale values that are lighter than the uniform background of the host rock. Materials with lower densities than the host rock (*i.e.* open space) are represented by pixels with grayscale values that are darker than the uniform background of the host rock (Figure 2.6.3). By thresholding the grayscale values that correlate to the x-ray attenuation for open space (pores) and specific minerals of known density, open and healed porosity can be identified, respectively, and mapped in three dimensions.

The density contrast between the healed minerals in the Newberry rock and the host rock is not great enough to be fully resolved by thresholding alone (Figure 2.6.4). In

addition, many of the mineral grains comprising the matrix are smaller than a single pixel, complicating the correlation of attenuation with mineral density (see discussion).

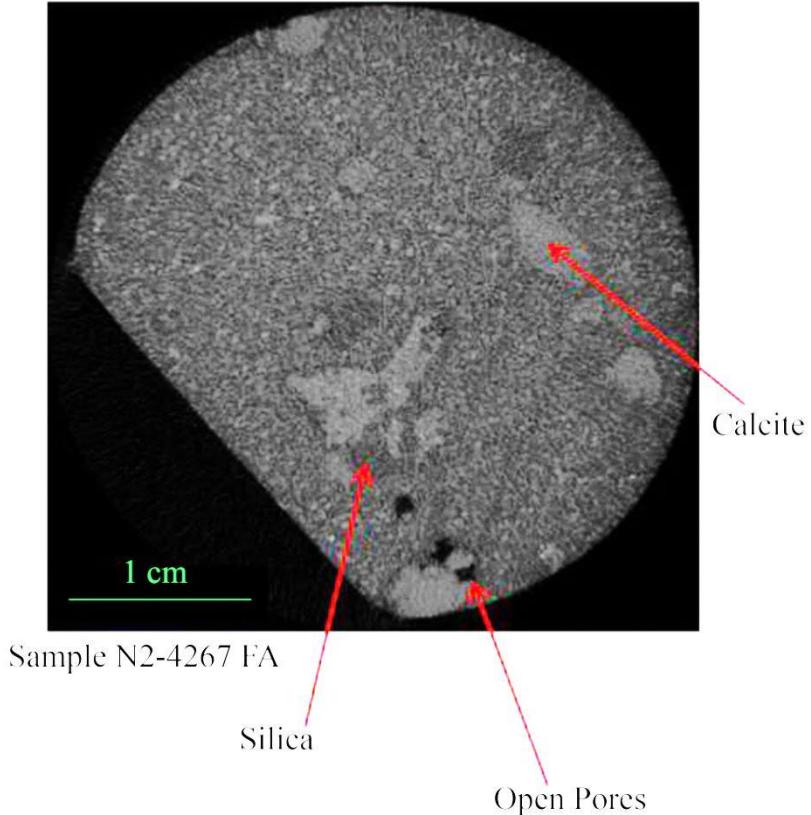


Figure 2.6.3: Cross-sectional slice of micro CT scan. Materials with lower densities than the host rock show up as darker shades of gray (open pores and silica in the photo) and materials with higher densities than the host rock show up as lighter shades of gray (calcite in the photo).

The *despeckle* tool available in CTan was used to filter out unwanted pixels that were selected for, but not representative of healed minerals. This processing results in three sets of binary images: (1) open porosity, (2) pore-filling minerals less dense than the matrix, and (3) pore-filling minerals more dense than the matrix. After the post-processing methods were complete and desired porosity map categories were generated, the bitmaps were uploaded into MATLAB[®] for further analysis.

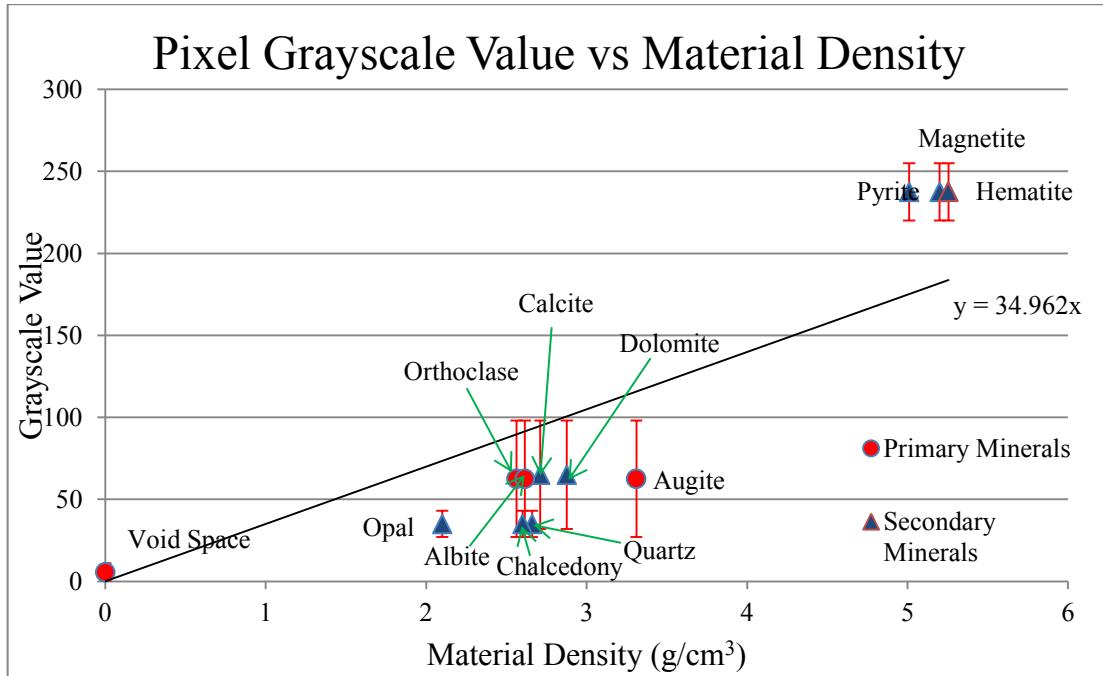


Figure 2.6.4: Pixel grayscale value vs. material density. Plot shows the broad range of grayscale values that represent materials in a micro CT scan. Error bars represent the lightest and darkest shades of grey that are meant to represent the respective material.

2.6.2 Determining Porosity Amounts

Scripts to automatically upload, trim, and process these data were developed in MATLAB® in order to analyze the processed bitmaps. See Appendix A for the scripts associated with this method. The MATLAB® algorithm allowed variation in porosity to be assessed within a moving cube with dimensions of set size that is moved along an established transect parallel to the z-axis of the sample. Similar to the high resolution image analysis, a cube size needed to be determined that (1) captured a sufficient number of pixels to yield a reliable porosity measurement, (2) did not lie outside the bounds of

the image data, and (3) can be directly compared to the size of the 2D analysis window. To investigate the window size appropriate for this study, a window sensitivity analysis was performed. This method of sensitivity analysis as described below determined an appropriate window size that generated results for this method. Figure 2.6.5 details the window sensitivity analysis. A MATLAB[®] algorithm (Appendix A) was written that plotted a summary of the variation in the profile of healed porosity for different sized cubes (Figure 2.6.5a), box plots illustrating the variation in porosity within the family of cubes for each window size tested (Figure 2.6.5b), and the mean and standard deviation of porosity along the sample profile as a function of the window size (Figure 2.6.5c). The box plots in Figure 2.6.5b show that the median of the data stabilizes early on and outliers are constrained more and more as window size increases. With very small cube sizes, the box plots usually hit or miss capturing porosity measurements, resulting in the broad range of outliers to the data. The largest cube sizes are rarely porosity-free. Figure 2.6.5c shows that the mean of the data stabilizes rather quickly, and as window size increases, the standard deviation decreases.

To satisfy the requirements of an appropriate window size, the cube size and pixels captured for 3D analysis was 0.1 x 0.1 x 0.1 cm and 52,526 pixels, respectively. The cube moved in set-increments, calculating porosity at each bitmap (26.7 μm spacing) by dividing the number of colored pixels by the total number of pixels contained in the cube. The fractional porosity at each step was then plotted against position along the transect. The total average porosity was calculated by taking the average of all the points comprising the transect data. The intersection of the cube with each bitmap results in a square window at constant z- position (i.e. orthogonal to the transect) that sub-samples

the porosity within each bitmap traversed and forms the basis for the volume average. Construction of parallel transects of equal window dimensions allows direct comparison of results from 2D and 3D methods. Figure 2.6.6 provides a direct comparison of core, CT image, computerized bitmaps, and the variation of healed, open, and skeletal porosity along an axis-parallel transect.

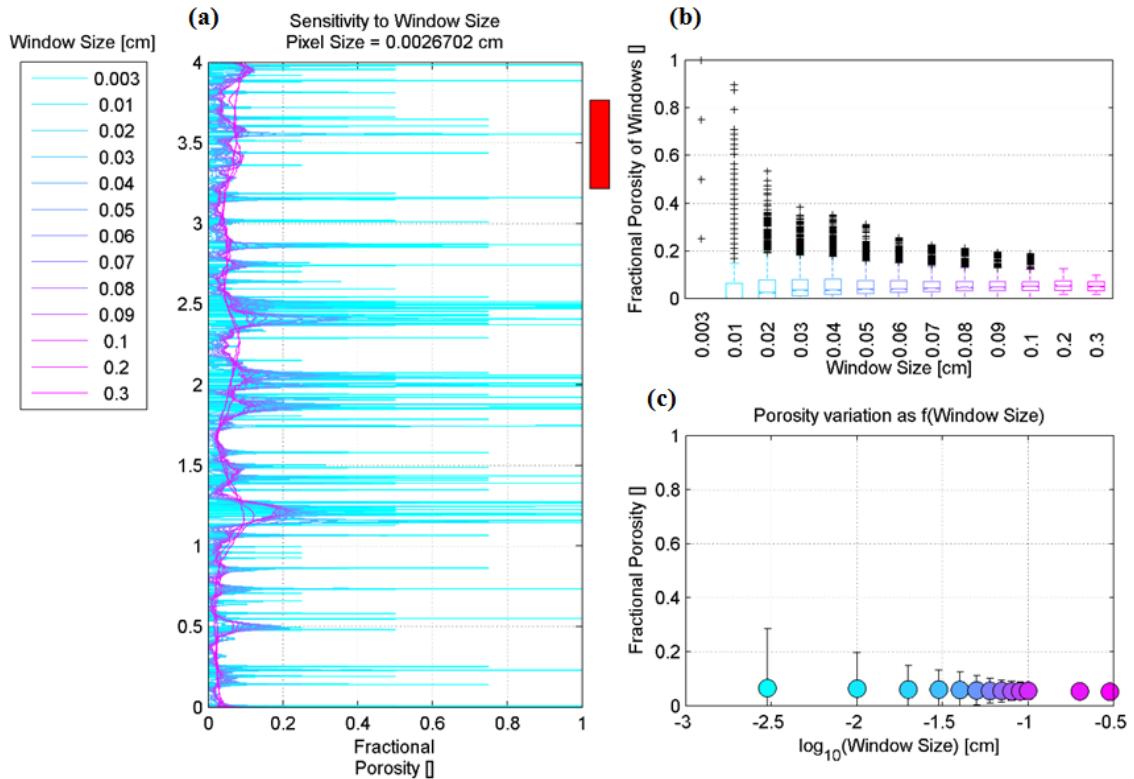


Figure 2.6.5: Window sensitivity analysis for micro CT scans. (a) Summary of the variation in the profile of the healed porosity for different sized cubes (sample 3937F). Darkest purple line represents the cube size used in this study 0.1 cm^3 . (b) Box plots illustrating the variation in porosity within the family of windows for each window size tested. (c) mean and standard deviation of porosity along the sample profile as a function of window size. Red box indicates approximate location of high resolution analysis.

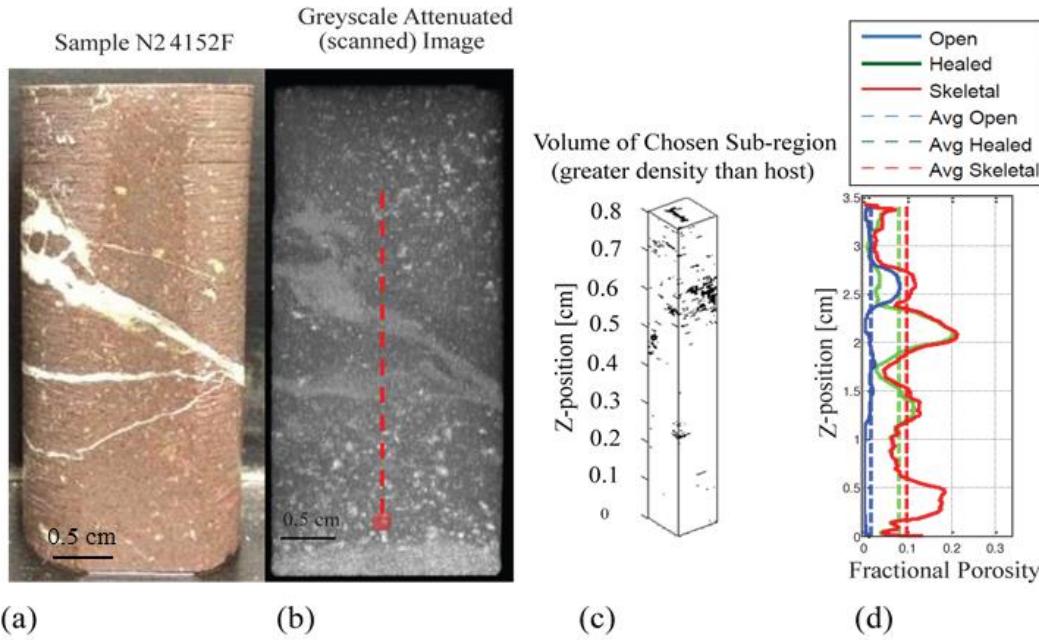


Figure 2.6.6: Micro CT method. (a) Photograph of core sample approximately 2.6 x 6 cm; (b) Vertical cross-section of micro CT scan of the core plug with 26.7 micron resolution. Note that the dashed red transect line for analyzing porosity is visualized and is located in the center of the core ; (c) Trimmed binary bitmaps to be measured for porosity values; (d) Variation in open, healed, and skeletal porosity along scan-line (b) and within volume shown in (c).

2.6.3 A Note About Thresholding Categories

It is important to remember that three bitmap categories were established by thresholding the image data to produce: (1) bitmaps representing open porosity, (2) bitmaps representing materials with a greater density than the host rock, and (3) bitmaps representing materials with a lower density than the host rock. In order to portray healed porosity, the results of the latter-two categories needed to be altered. Since open pores also have a density lower than the host rock, the results of open porosity were subtracted from the results of the materials with a lower density than the host rock. These results were then added to the results of materials with a greater density than the host rock to

correctly represent healed porosity. Figure 2.6.7 is a workflow diagram that visualizes the steps necessary for micro CT analysis, including the scanning the core, reconstructing the bitmaps, thresholding the porosity, despeckling unwanted pixels, and analyzing transect porosity, all of which are detailed above.

2.7 Measurements of Fracture Roughness

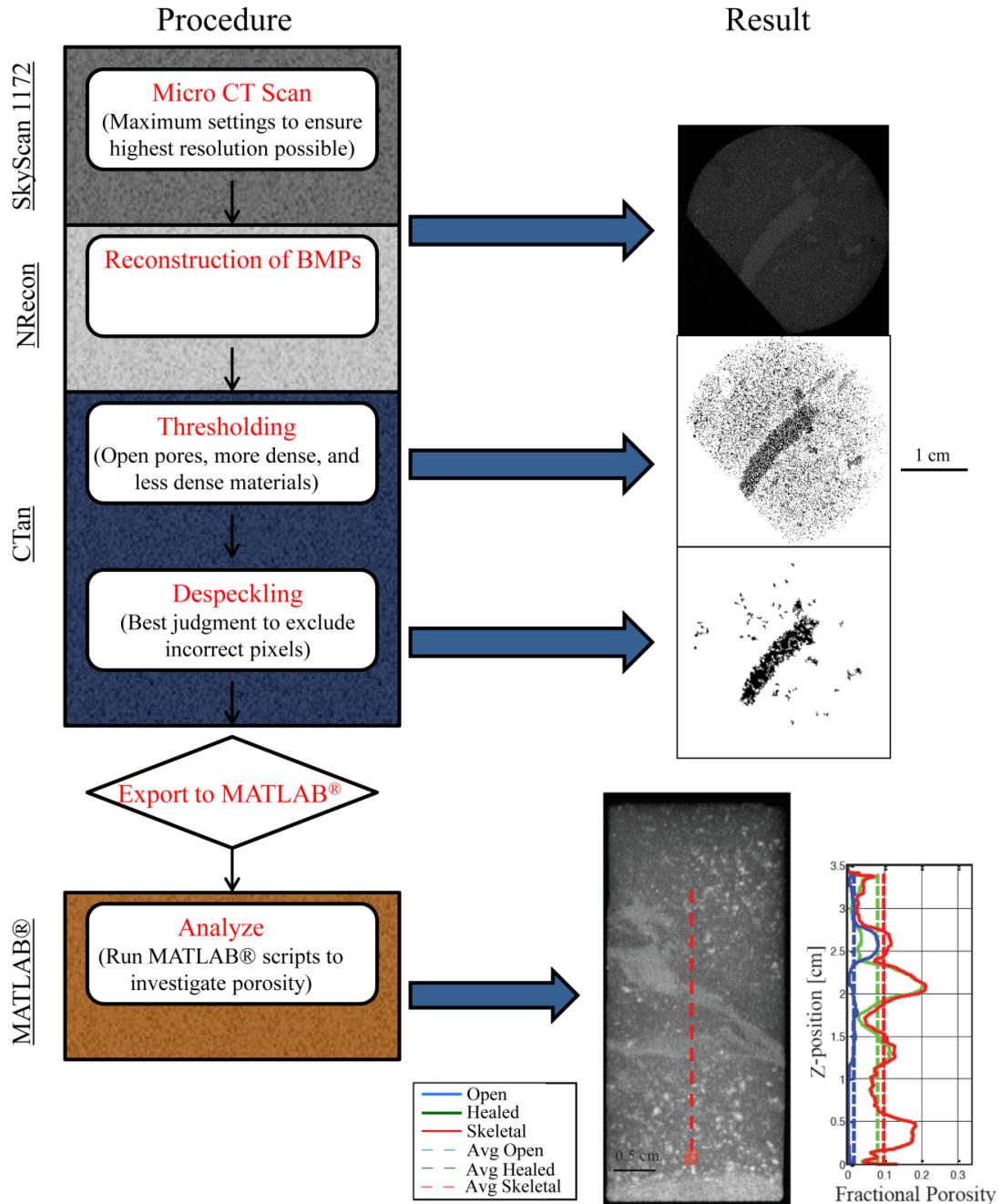
High-resolution mapping of porosity in thin sections, and the textures evident in photomicrographs of the fractured rock, document the minimum accumulated dilation associated with the fracture. In most samples, healed porosity is actually comprised of layers of secondary minerals (Figure 1.2.3) that vary in thickness along the fracture surface and indicate relative timing relationships through superposition and cross-cutting. The base of each of these layers documents a paleo-fracture surface and thus the correlated histories of surface roughness, dilation, and healing in a natural fracture experiencing repeated failure. Thin sections aligned with the slip direction and orthogonal to the fracture surface were used to digitize the fracture surface. Textural relationships between the host rock and layers of cements evident from petrographic analysis and cathodoluminescence (CL) allowed distinct layers to be mapped, which were then interpreted to represent distinct slip/dilation events. Thus the initial fracture surface roughness and its modification through repeated failure were measured as well as the associated porosity generation due to dilation accompanying slip.

2.7.1 Identifying, Digitizing, and Assessing Fracture Surfaces

Precipitation of secondary minerals from fluid flow in a fracture gives rise to healing minerals (thus healed porosity), which can repeatedly fracture and re-seal. The progression of healing and slipping through time provides insight into how a fracture's roughness might change with fracture maturity, provided the fracture surfaces are preserved.

Though the mineralogy of the fracture cement can be uniform (i.e. entirely calcite), as described below, CL can help distinguish the surface boundaries generated by different slip events. Migrating fluids in the subsurface contain impurities (minor species such as iron and manganese) that, when precipitated into cement and analyzed via CL, cause the different layers to be readily distinguishable. As long as the fluid impurity content changes for each healing event, differences in the CL results can be identified by differences in the color and intensity of luminescence.

CL involves focusing a high-intensity electron beam onto a rock sample to energize the electrons in atoms comprising the sample. The beam is focused by a deflection magnet assembly that bends the beam of electrons onto the sample. When the electron beam is applied to the sample long enough, the electrons of the sample become sufficiently excited and emit radiation in the visible spectrum (Marshall, 1988).



A fracture at an early stage of development (sample N2-3937 FA) and containing petrographically distinct layers of cement was selected to analyze the progressive evolution of roughness and dilation. The early stage fracture is characterized by small slip at the scale of the thin section and relatively simple geometry facilitating an accurate mapping of the series of fracture surfaces interpreted as past slip and dilation events. In addition, the primary cement in this fracture is calcite, with minor quartz. Calcite shows strong luminescence under CL providing the best opportunity to reliably distinguish layers.

The thin section billet containing this early-stage calcite-filled fracture was placed into the vacuum specimen chamber of a Luminoscope[®] Model ELM-2A (Figure 2.7.1). Since CL requires thin sections to be uncovered, and since the thin sections in this study were covered during preparation, the billet from which the thin section was cut was used for analysis. After the vacuum chamber reached a pressure between 100 and 150 millitorr, the energy source was initiated. Visible luminescence was possible after the cathode was given roughly 15 minutes to warm up and when the beam voltage was greater than 5 kV. The pattern of luminescence was documented in photomicrographs and subsequently superimposed over a scanned image of the thin section to help identify correlated cement layers.

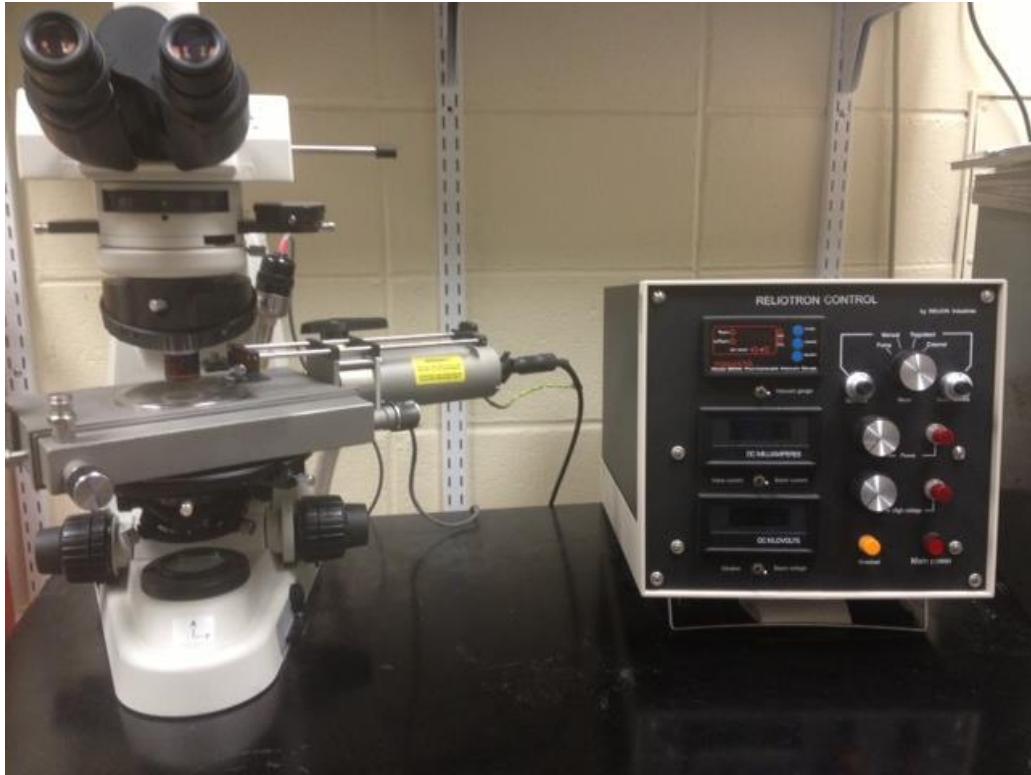


Figure 2.7.1: Photograph of the Luminoscope[®] Model ELM-2A used for cathodoluminescence analysis.

By using the combined petrographic mineral/textural analysis and CL images, entire surfaces were identified and traced with a color pencil on a printed picture of the thin section. Relative ages were inferred from superposition and cross-cutting relationships. After fracture surfaces were identified, a scanned image of the thin section was uploaded into the digitizing software DIGIT where each fracture surface was digitized by hand. The resulting sets of Cartesian coordinates of each point along the digitized surfaces were saved as separate text files to be further analyzed by MATLAB[®]. Algorithms were created that assessed the statistical characteristics of the topography of these surfaces including the surface length, the topography relative to a flat, planar reference surface, and the wavelength and amplitude of asperities comprising the surface

topography.

2.7.2 Determining Grain and Pore Size Distribution

Fracture surfaces are inherently rough (e.g., Power and Tullis, 1991; Dieterich and Kilgore, 1996), but primary characteristics such as grain and pore geometries have the potential to contribute to or control the roughness of fracture surfaces. In an attempt to correlate surface roughness characteristics to attributes of the host rock, the grain and pore size distributions were measured. These measurements are comprised of the orthogonal long and short axes of the grains and pores as visible within a roughly 1 cm by 1 cm photomicrograph taken at 4x magnification. Grains smaller than 0.02 cm were not reliably measured, so this analysis is skewed toward the large sized grains and represents a complete catalog of the grains that exceed this minimum dimension. Although the smaller grain sizes were neglected due to the limitations of the measurement method, it is expected that the larger features contribute most to the amplitude and wavelength of asperities on the fracture surface, and which force the walls apart during slip to accumulate dilation.

CHAPTER 3

RESULTS

3.1 Description of Porosity in Samples

Of the porosity types discussed in section 1.2.5 (Figure 1.2.3), primary and secondary healed porosity dominates at Newberry and is primarily comprised of calcite and silica. Silica is present in the forms of chalcedony, opal, and quartz. The healed porosity occurs either in primary vesicles (P.V.) of the host rock (Figure 3.1.1a, b, d) having an elliptical shape, some measuring as large as 5 mm in diameter along the long axis (as discussed in the later section on surface roughness: Section 3.5), in secondary elongated fractures (S.F.) (Figure 3.1.1a, b) having a linear shape, or within secondary damage zones (D.Z.) surrounding the fractures (Figure 3.1.1 b, c) having broken, irregular shapes.

The layering of these cements begins at the fracture or vesicle walls and continues to heal inward until (1) all space for mineral growth is impeded, leaving behind a fully healed structure, or (2) healing suddenly stops, leaving behind open pore space. Open porosity (O.P.) in the samples is minimal, but where present it occurs within elliptical vesicles (Figure 3.1.1a, b), within fractures (Figure 3.1.1b), and occasionally within damage zones associated with fractures (Figure 3.1.1c). This open space can be attributed to (1) a dissolution of minerals or a lack of mineral precipitation by the healing fluids, as identified by smooth healed surfaces surrounding the opening walls, or (2) an

actual break and dilation within the already healed structure, as identified by rough healed surfaces surrounding the opening walls.

As pores approach a fracture and its corresponding damage zone, their shapes become elongated, with definite short and long axes. Away from the fracture, however, pores are more spherical and uniformly shaped. Figure 3.1.1d shows a section of footwall distal to a fracture that contains healed vesicles that are nearly spherical, as they do not occur close to a fracture. Figure 3.1.1c, on the contrary, is a section of heavily damaged fault breccia containing very elongated, irregular pore shapes often associated with areas of brittle deformation.

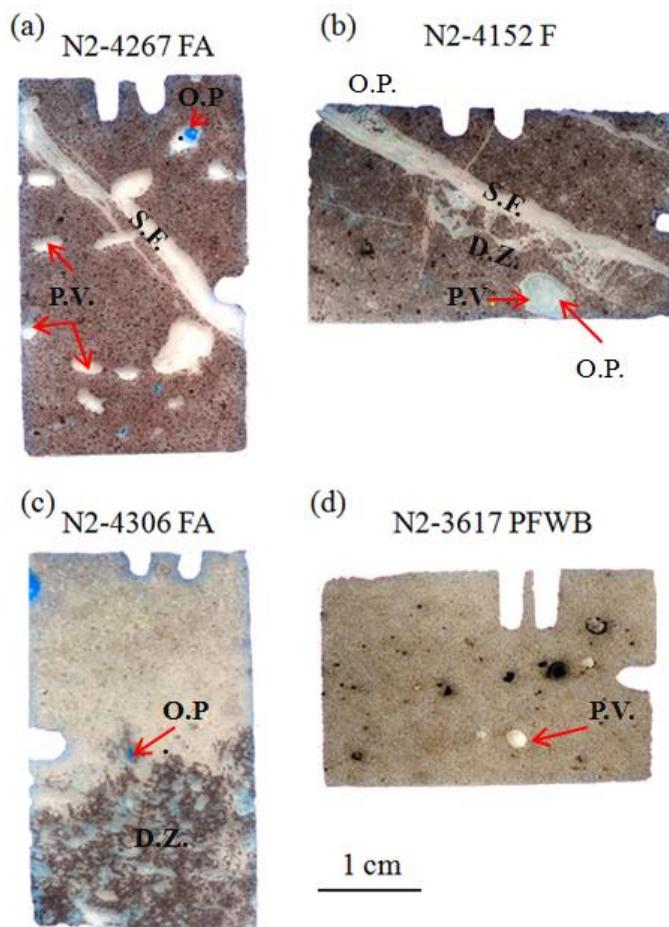


Figure 3.1.1: Thin sections showing types of porosity in Newberry Rock. (a) Thin section shows healed secondary fracture (S.F.), primary vesicles (P.V.), and open

porosity (O.P.); (b) Thin section shows healed secondary fracture (S.F.), primary vesicle (P.V.), open porosity (O.P.), and damage zone (D.Z.); (c) Thin section shows open porosity (O.P.) and damage zone (D.Z.); (d) Thin section showing a spherical primary vesicle (P.V.).

3.2 Point Counts and Automated Image Analyses

Point count analyses and automated image analyses can be directly compared because these methods calculate the porosity across an identical area so that differences in sample size do not lead to false correlations. Both methods yield porosity maps that give insight into the 2D pore size and fracture shape. Point count porosity maps are pixilated and of lower resolution than the automated thin section image maps; however, the categorization of points is highly accurate and flexible because each point is evaluated by a researcher and confirmed petrographically. Automated thin section porosity maps show much more detail in 2D porosity structure, but lose accuracy because of the lack of researcher-decision during the thresholding process. Point counts yield open porosity values within the range of 0.1% to 6% and healed porosity values within the range of 14.2% to 73.3%. Automated image analyses yield open porosity values within the range of 0% to 5.5% and healed porosity values within the range of 24% to 63.4 %. Fractional porosity values of identical samples significantly differ between the two methods. For the Newberry rocks in this study, point counts typically yield skeletal porosity values that are 5-19% lower than the automated image analyses (Figure 3.2.1). The values of open porosity tend to be similar across the techniques for identical samples, but healed porosity values are significantly different, and are directly attributable to the difference seen in skeletal porosity. See Appendix B for a complete table of porosity results. See Appendix C for a full set of point count and automated image analysis porosity maps and plots.

3.3 High-Resolution Images and Micro CT Scans

The transect data for high-resolution images and micro CT scans can be directly compared because the moving windows along each transect are constrained to have the same dimensions and analyze moving regions of comparable sizes and sufficient pixel count. Appendix D and E contain full datasets of high resolution and micro CT scan results, respectively.

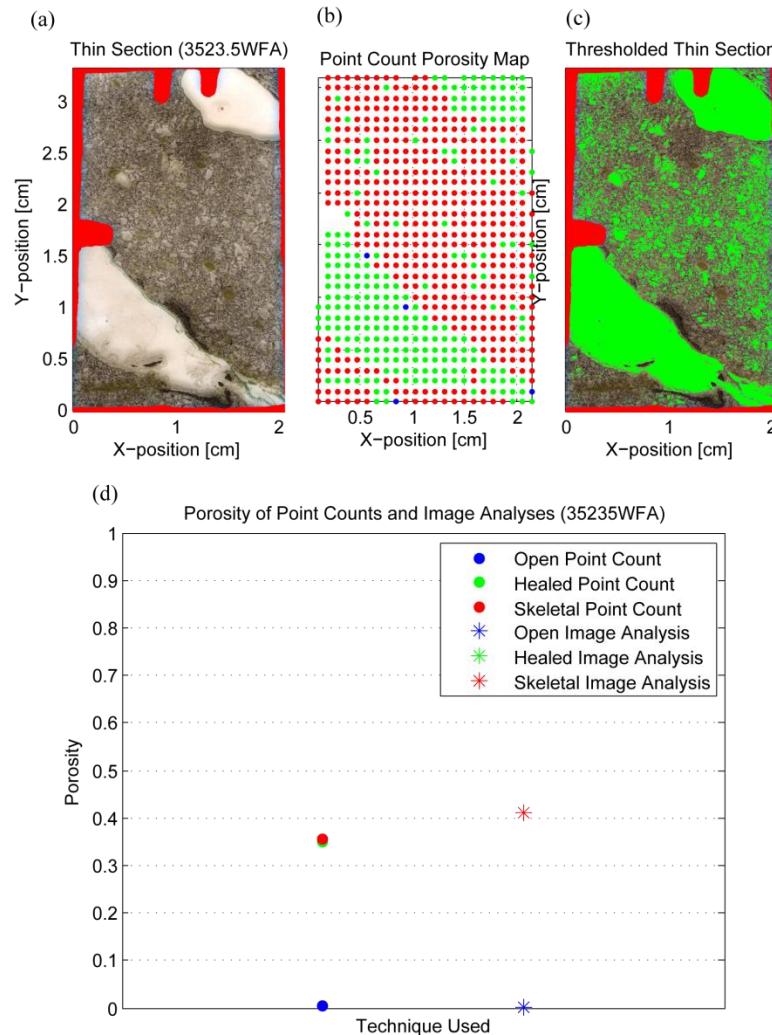


Figure 3.2.1: Thin section analysis results. (a) Scanned thin section image; (b) Porosity map derived from point counts; (c) Porosity map derived from automated analysis; (d)

Plot of fractional porosity vs. technique used.

For the rocks at Newberry, high resolution image analyses yield open porosity values within the range of 0% to 1.7% and healed porosity values within the range of 13.7% to 56.4%. Micro CT scans yield open porosity values within the range of 0% to 1.6% and healed porosity values within the range of 0.7% to 16.3%. Both methods yield skeletal porosity values that increase, peak, and decrease as the moving window approaches, coincides with, and moves away from the fractures of interest (Figure 3.3.1). However, while similar patterns in the transect data were identified between the two methods, the peak values of porosity are significantly different between the two techniques. In particular, micro CT scans produce consistently lower values of open and healed porosity along the transect. In some samples, the peak value across a fracture reaches as little as one-fourth of the value obtained in the high resolution image analysis, despite the uniqueness of open-porosity identification in both techniques. In addition to showing consistently lower values of skeletal porosity, micro CT results yield anomalously high spikes in porosity not found in the high resolution image analysis.

As expected, when viewing the high-resolution image transect data, peaks in the fractional porosity along the transect directly coincide with areas that are visibly filled with porosity in the photo stitches (Figure 2.5.1). With micro CT results, however, this is not the case. Anomalously high spikes in porosity occur throughout the samples in locations where they cannot be independently verified from the 3D image of the core (Figure 3.3.2). These spikes are exaggerated and more prominent towards the tops and bottom of each transect. In particular, open porosity is amplified at the tops of transects whereas healed porosity is amplified at the bottom of the transects (Figure 3.3.3).

Another effect not seen in the high resolution image analysis but evident in the micro CT results involves the appearance of strong shadow zones surrounding minerals of relatively high density (Figure 3.3.4). For example, high density minerals such as pyrite cause a shadow zone to develop around the mineral grain, suggesting the presence of open porosity where it clearly does not exist. So while these two methods investigate porosity in similar regions of the same core sample, significant differences in the results exist that can cause discrepancy when interpreting the data.

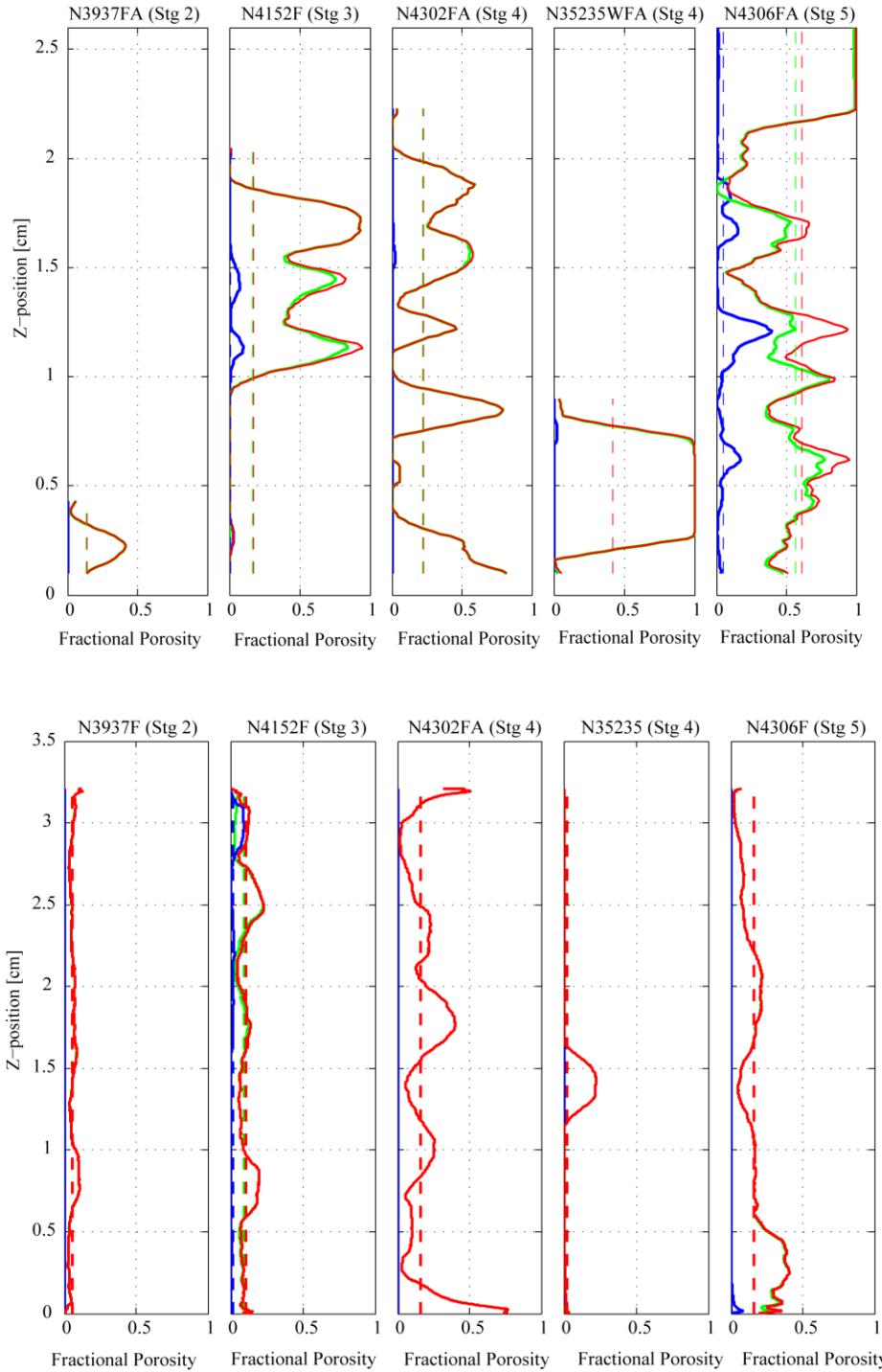


Figure 3.3.1: High resolution image and micro CT results. Top) Porosity transects for five high-resolution image analyses; Bottom) Comparable porosity transects for five micro CT analyses. Blue, green, and red lines refer to open, healed, and skeletal porosity, respectively. Dashed lines indicate an average value of fractional porosity.

Greyscale Attenuated (scanned)
Image

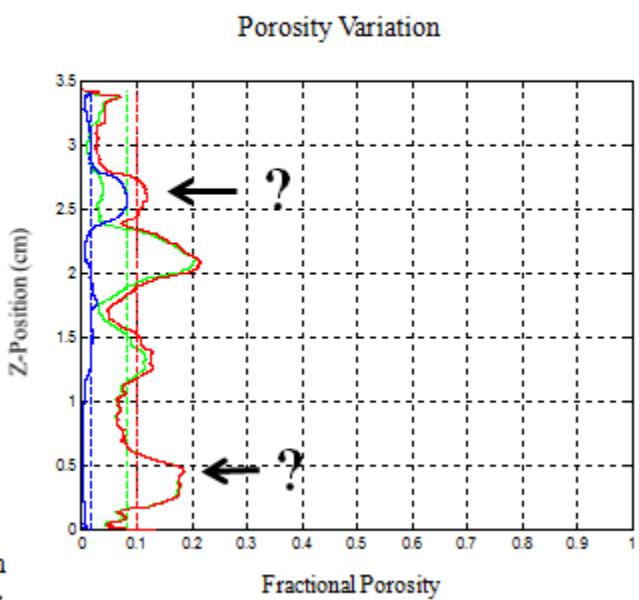
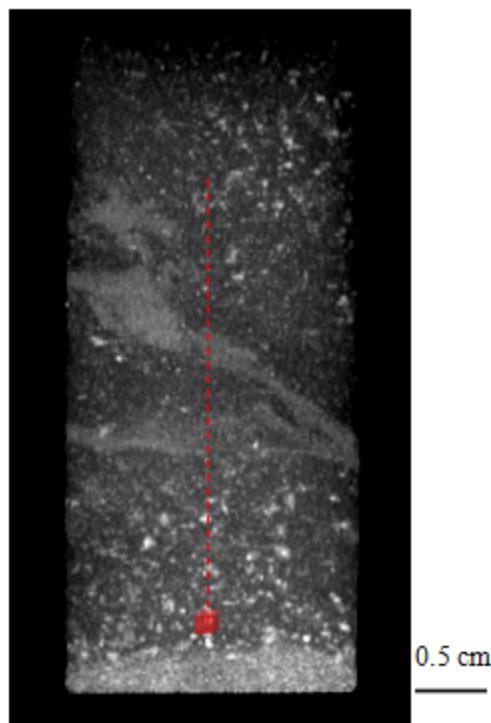


Figure 3.3.2: Anomalous peaks in micro CT data. Arrows point to peaks that can't be visually attributed to porosity presence along the transect.

Greyscale Attenuated (scanned) Image

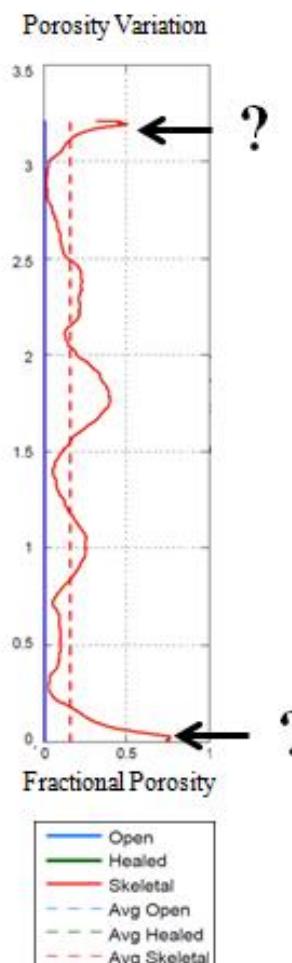
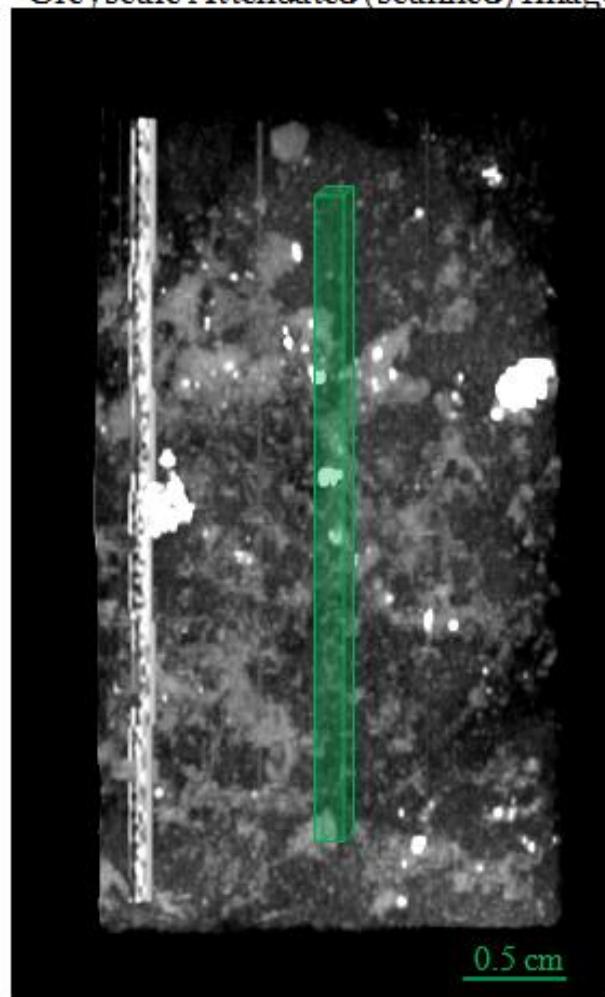


Figure 3.3.3: Edge effects of micro CT scans. Arrows point to areas of anomalously high porosity values.

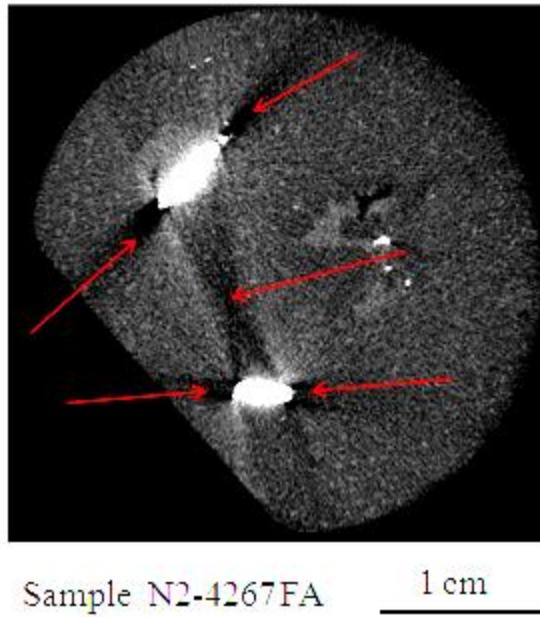


Figure 3.3.4: Areas of low attenuation found near pyrite grains in a 2D slice (Bitmap of an micro CT scan. Red arrows point to areas of low attenuation).

3.4 Multi-Transect Analysis to Asses Statistical Robustness of Measured Porosity Structures

3.4.1: High Resolution Image Transect Variability

To assess the statistical variability of porosity within samples captured during high resolution image analysis, multiple transects were analyzed. Thirteen scan lines sampling open and healed porosity variation in the y-direction were evenly distributed along the x-direction of the width of the sample. These scan lines can be visualized on top of the sample as different colors in Figure 3.4.1 showing the measured magnitude of open (left) and healed (right) fractional porosity along each scan line.

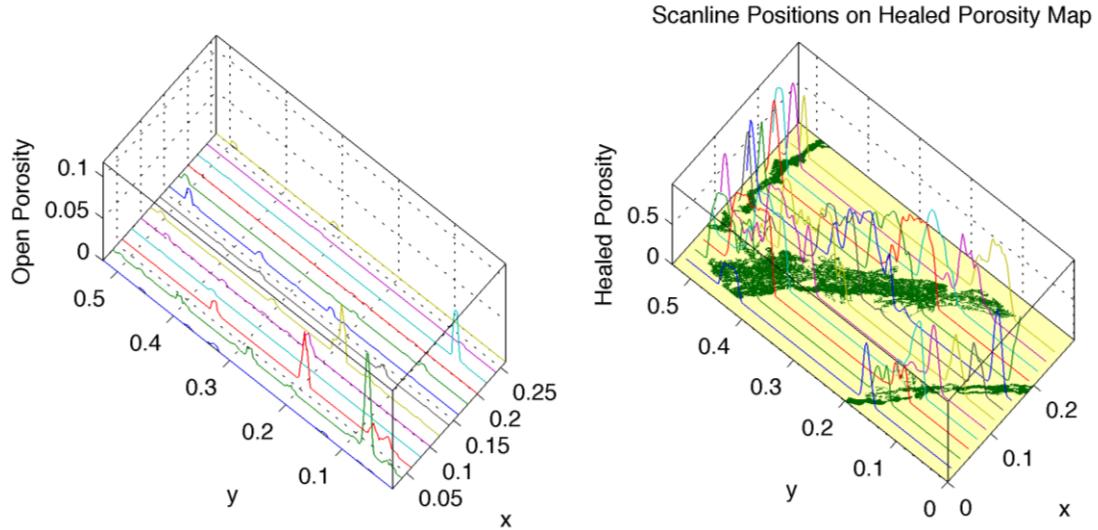


Figure 3.4.1: Perspective view of results of open and healed porosity multi-transect analysis for high resolution image analysis. Left: Open porosity transects. Right: Healed porosity transects.

The results of this analysis can be seen in Figure 3.4.2 where open (Figure 3.4.2a) and healed (Figure 3.4.2b) porosity along the thirteen transects is plotted. As before, the porosity is determined from the number of pixels corresponding to pores compared to the total number of pixels within a sample window. In Figure 3.4.2, lines represent the boxcar moving average of each window as it moves along the sample. The colors represent the measurements of distinct transects. The two box plots on the right (Figure 3.4.2c, d) summarize the range of measurements along the thirteen transects comprising this analysis. The diamonds represent the mean porosity along the scan line. Figure 3.4.2c pertains to healed porosity whereas Figure 3.4.2d pertains to open porosity.

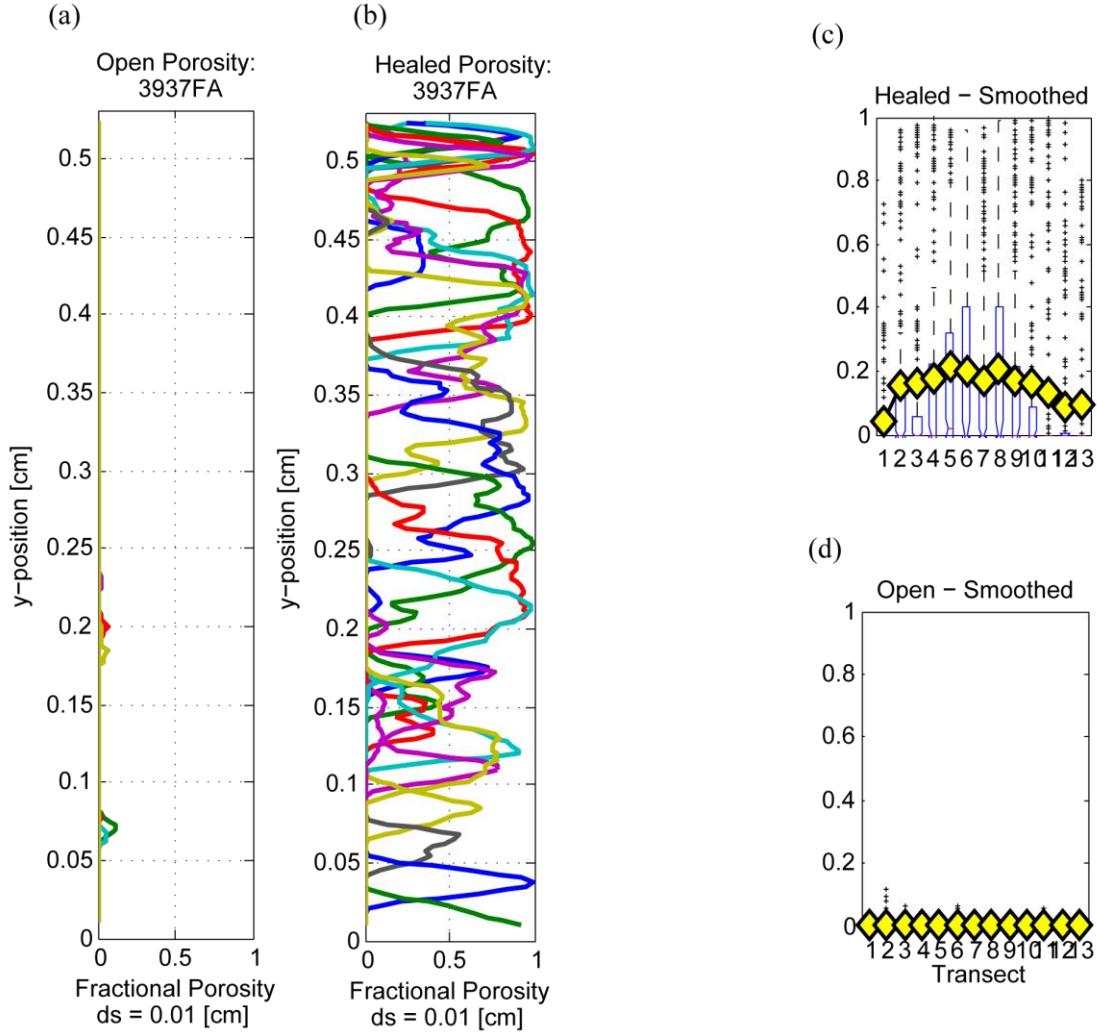


Figure 3.4.2: High resolution image based analysis of open and healed porosity along multiple transects parallel to the y-axis of the sample. (a) Open porosity variation; (b) Healed porosity variation; (c) Box plot for healed porosity; and (d) Box plot for open porosity.

The statistical variability in the (a) open and (b) healed porosity along the family of thirteen transects is seen in Figure 3.4.3 and was summarized by computing the minimum, 25th, 50th, and 75th percentiles of porosity at each y-position from the set of scan lines. In Figure 3.4.3, individual scan lines are superimposed as gray lines for reference.

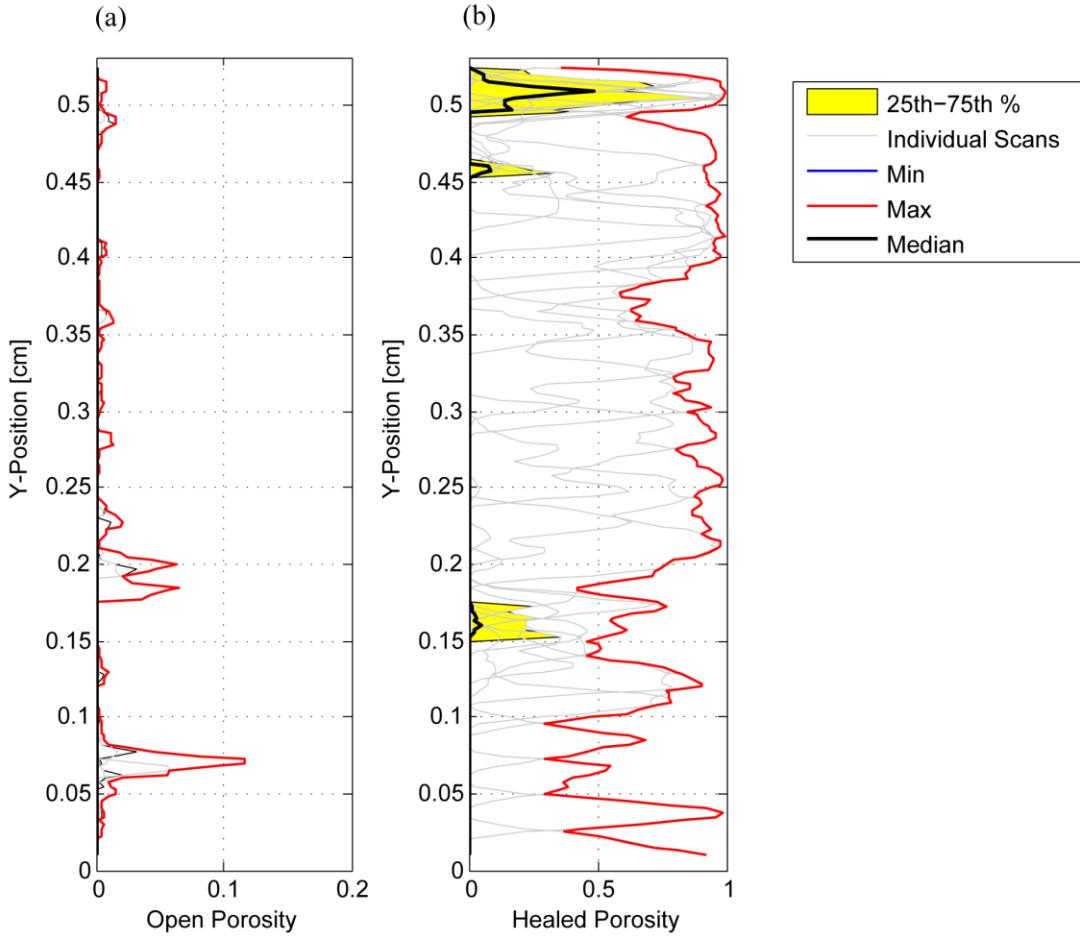


Figure 3.4.3: Statistical variability in the open and healed porosity along the family of 13 high resolution image transects.

3.4.2 Micro CT Transect Variability

To assess the statistical variability of porosity within samples captured by the micro CT, multiple transects were analyzed (Figure 3.4.4). The volume comprised of stacked binary bitmaps of open and healed porosity is represented in Figure 3.4.4 by the dashed magenta lines outlining the rectangular solid shown in perspective. Individual transects are indicated by colored lines recording sample points along transect lines

parallel to the long, z-axis, of the sample volume. The sample window within each bitmap is plotted to scale at the top blue transect line.

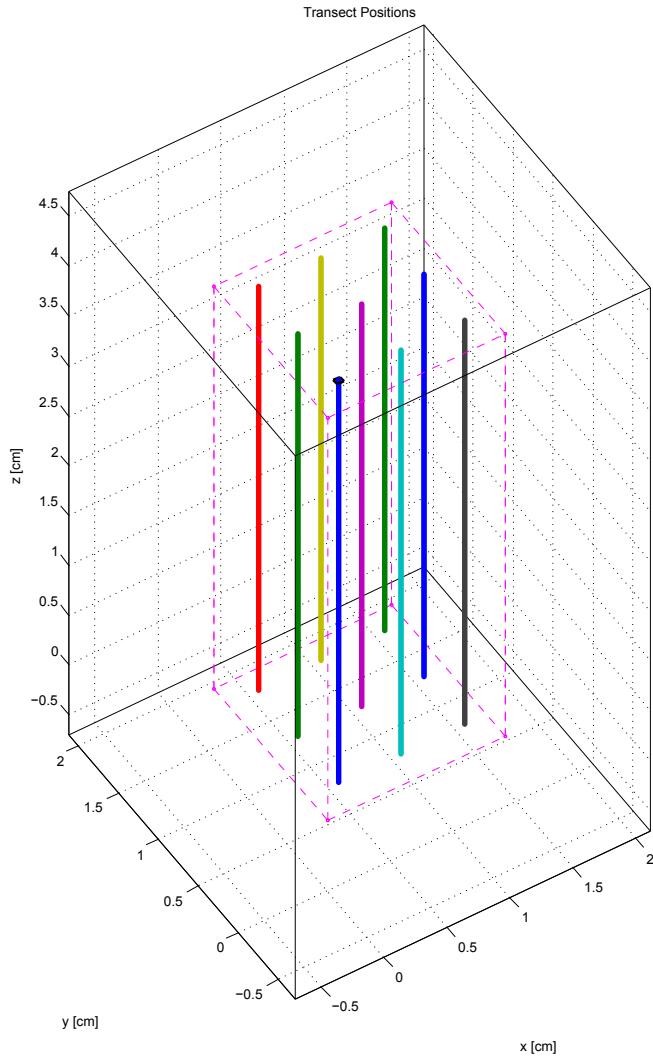


Figure 3.4.4: Visual of multi-transect analysis for micro CT scans. Individual transects are indicated by colored lines running parallel to the z-axis.

The results of this analysis can be seen in Figure 3.4.5 where open (Figure 3.4.5a) and healed (Figure 3.4.5b) porosity along multiple transects parallel to the z-axis is plotted. As before, the porosity is determined from the number of pixels corresponding

to pores compared to the total number of pixels within a sample window. In Figure 3.4.5a and b, dots represent the porosity measurements within single, 2D bitmaps perpendicular to the z-axis whereas lines represent the boxcar moving average of multiple bitmaps representing a cube-shaped sampling window. The colors represent the measurements of distinct transects. The four box plots on the right (Figure 3.4.5 c, d, e, f) summarize the range of measurements along the nine transects comprising this analysis. The diamonds represent the mean porosity along the scan line, and the horizontal dashed lines represent the average across all scan lines.

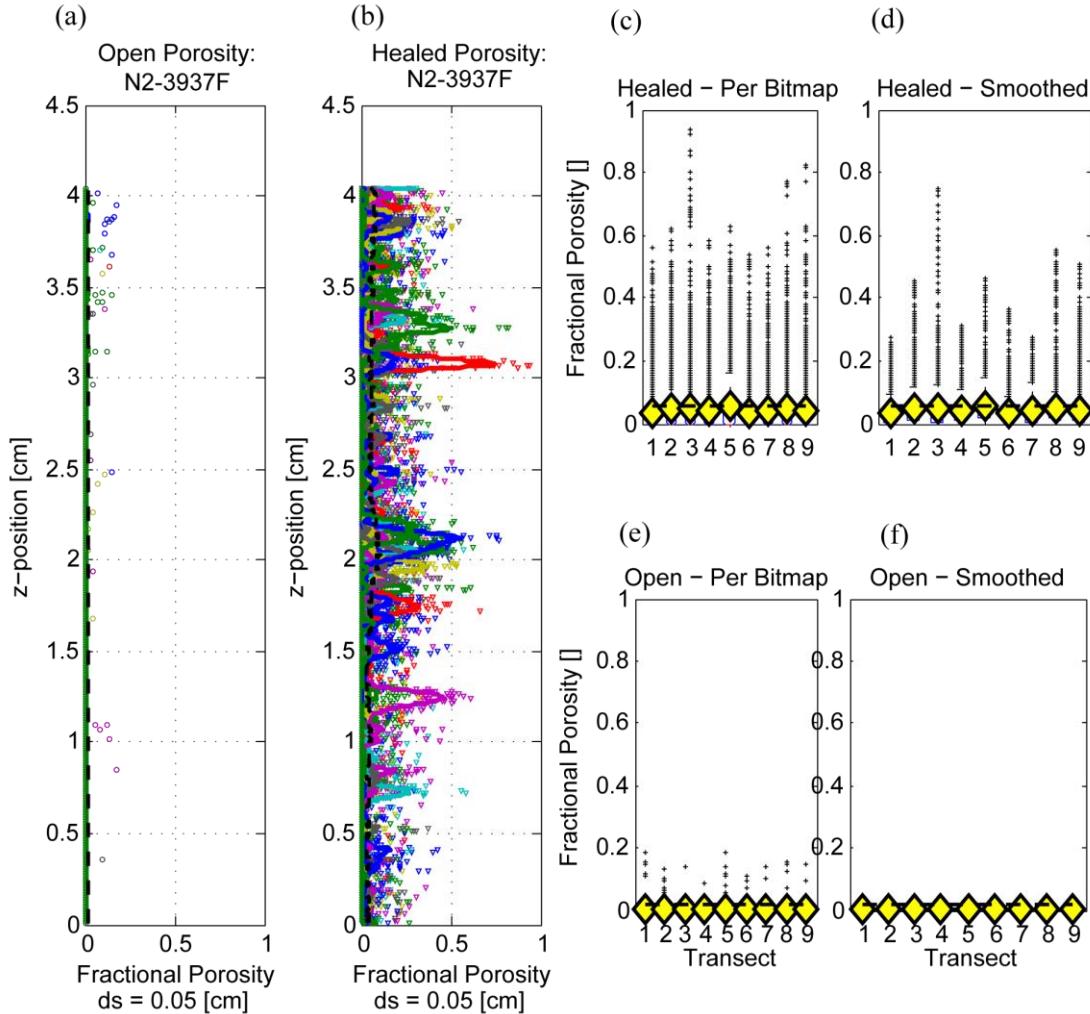


Figure 3.4.5: Micro CT based analysis of open and healed porosity along nine transects

parallel to the z-axis of the sample. (a) Open porosity variation; (b) Healed porosity variation; (c) Box plot for healed porosity per bitmap; (d) Box plot for healed porosity smoothed; (e) Box plot for open porosity per bitmap, and (f) Box plot for open porosity smoothed.

The box plots show the variation in porosity measurement within each sample window comprising scan lines 1 through 9. The upper row is for the healed porosity whereas the lower row is for the open porosity. The statistical variability in the open and healed porosity along the family of nine transects is seen in Figure 3.4.6 and was summarized by computing the minimum, 25th, 50th, and 75th percentiles of porosity at each z-position. In Figure 3.4.6, individual scan lines are superimposed as gray lines for reference.

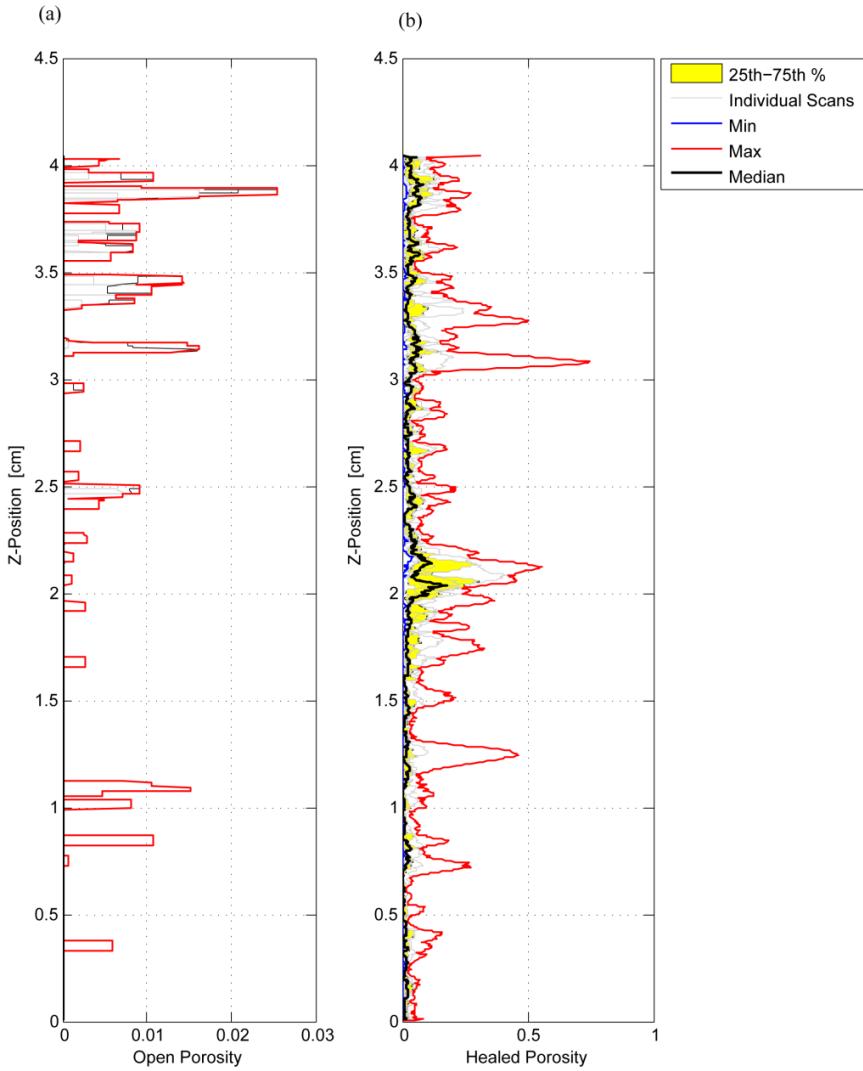


Figure 3.4.6: Statistical variability in the open and healed porosity along the family of nine Micro CT transects.

For each of the high resolution image and micro CT transects analyzed, open porosity measurements remained constant (Figure 3.4.2d and 3.4.5e, f). There is a systematic rise and fall in porosity as the moving windows encounter pores, as expected (Figure 3.4.1 and 3.4.5b). The porosity measurements in the wall rock adjacent to the fracture surfaces shows no systematic variation, whereas the location of peak porosity along some of the transects changes substantially. For the high resolution image analysis, viewing the transects and their magnitudes overlain on the thresholded sample image of healed

porosity (Figure 3.4.1b) shows a relationship between the angle at which the fracture intersects the transects and the position of the maximum porosity. Because the central and lower left fractures are inclined with respect to the y-axis (Figure 3.4.1b) the position of the maximum porosity progressively migrates along scan lines. Transects that are nearly orthogonal to the upper-right fracture in Figure 3.4.1b yield maximum porosity positions that roughly coincide with each other. For the micro CT analysis, the shift in porosity peaks along scan lines is not readily seen because the positions of the scan lines do not parallel a sample x- or y- axis (Figure 3.4.4). Although each of the scan lines for high resolution image analysis and micro CT has porosity peaks at different y- and z-positions, respectively, box plots show that the mean porosity values of each scan line remain relatively constant (Figure 3.4.2c,d and 3.4.5c, d, e, f). In Figure 3.4.2c, the mean healed porosity drops at the margins of the sample because the scan line positions begin to fall outside of the boundaries of the data. Thus, despite differences in the position of maximum porosity, the statistical characteristics of the similar length transects for both methods as summarized in Figure 3.4.3 and Figure 3.4.6 can still be compared.

3.5 Fracture Surface Roughness

Through careful study of relative age relationships, superposition, textural and mineralogic variation, and cement grouping based on cathodoluminescence (CL) color variation, 19 pairs of fracture surfaces were identified and differentiated into three groups (denoted Segments A-C) (Figure 3.5.1a, b). Segment A occurs in the oxidized region of the thin section. This segment of cements is relatively linear, can be traced along the entire length of the fracture, and displays the dullest colors of orange when exposed to

CL. Segment B also occurs in the oxidized region of the thin section but displays significantly brighter colors of orange when exposed to CL. This segment terminates when it intersects the damage zone of the fracture. Lastly, segment C, occurring in the non-oxidized region of the thin section, is linked to, but branches off from Segment B at a curved junction. This segment displays the brightest shades of orange while under the CL. Again, these differing shades of orange represent changes in chemistry of the precipitating fluids that formed the cement. Independently rotating the trace map to visualize the topography along each surface (Figure 3.5.1c) reveals that Segment C has the greatest variation in along-fracture topography. A plot the frequency of asperity amplitudes of each surface (Figure 3.5.1d) shows three distinct groups, one of which coincides with the long axis 2D grain size distribution of the host rock (Figure 3.5.1e). A box plot summarizing the variation in the topography of each surface (Figure 3.5.2a) and a plot of the ratios of piecewise fracture surface length, L_s , to straight-line length, L_{\min} reveals the variation in surface area and roughness associated with each break (Figure 3.5.2b) as a function of relative age.

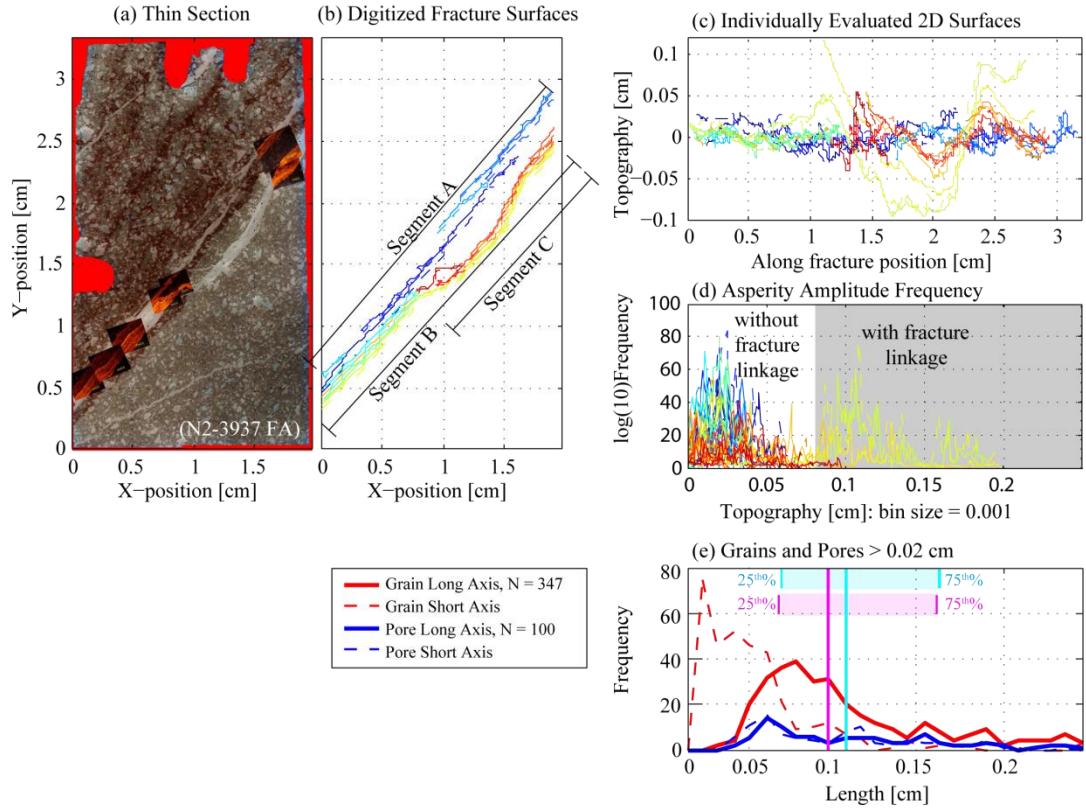


Figure 3.5.1: Fracture roughness results. (a) Cathodoluminescence images superposed onto thin section scan to aid in distinguishing correlated cements; (b) Mapped fracture surfaces. Multiple surfaces are evident from the superposition and crosscutting relationships of distinguishable cements. Solid lines indicate upper surfaces and dashed lines indicate lower surfaces; (c) Topography of rotated surfaces from (b); (d) Corresponding frequency of asperity amplitude along each fracture surface (colors are the same in both plots); (e) 2D size distribution of the greater than 0.2 mm fraction of phenocrysts and pores in the thin section.

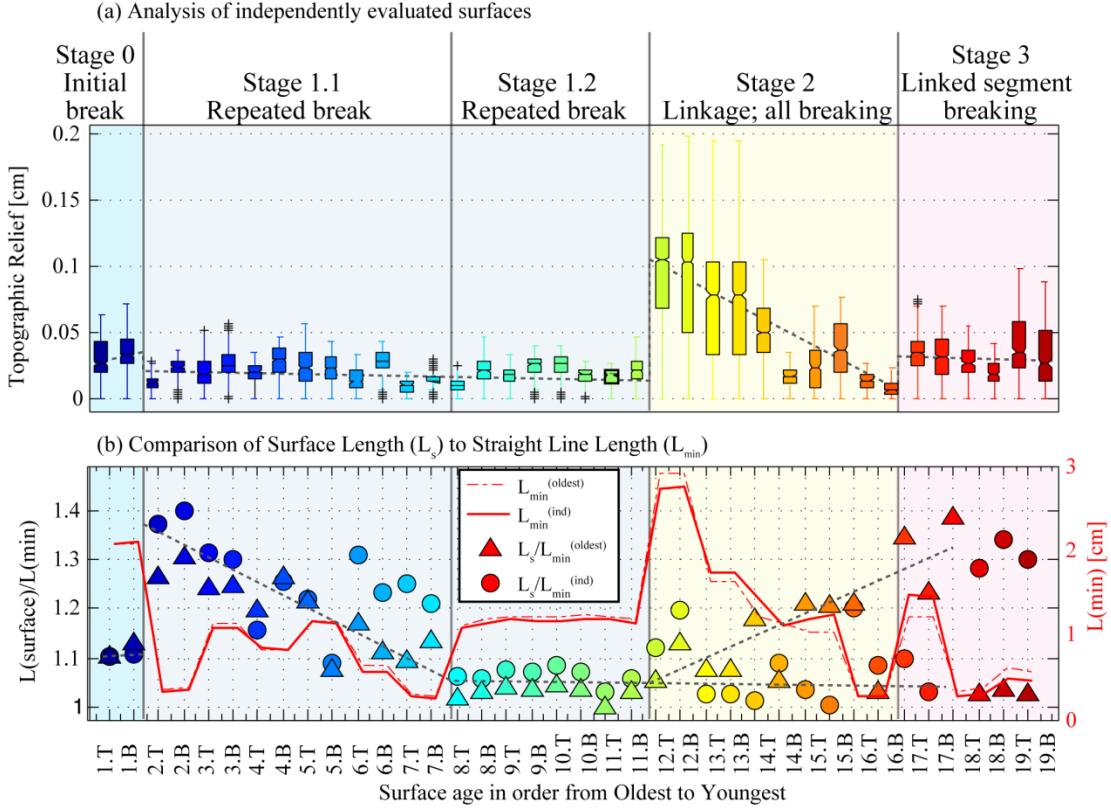


Figure 3.5.2: Plots from fracture surface data. (a) Variation in topographic relief of the individually analyzed fracture surfaces. The boxes span the 25th to 75th percentiles, with the inner, horizontal line corresponding to the median. The notch indicates the 95% confidence interval of the median; other median values outside this notch are statistically different populations to 95% certainty. The whiskers indicate the 10th and 90th percentiles and the plus symbols are outliers; (b) Ratio of piecewise fracture surface length, L_s , to straight-line length, L_{min} , as a measure of surface roughness for individually evaluated surfaces (circles) and surfaces evaluated in references to the oldest surface (triangles). Notice that since not all of the fracture is reactivated during each slip and dilation (see reference in Figure 6b), successive surfaces are not all the same length. The minimum straight line length of these surfaces is recorded by the red and red dashed lines which make reference to the right-side y-axis, also in red.

CHAPTER 4

DISCUSSION

4.1 Method Comparison

4.1.1 Point Counts versus Automated Image Analyses

Point counting thin sections is a relatively straightforward method that can be conducted by nearly any researcher. Aside from time and labor costs, the method is relatively inexpensive and requires only a petrographic microscope to complete. However, while point counts provide point-by-point observations of minerals in thin section, ultimately providing insight into mineral textures and porosity evolution, they do take time, require constant researcher-decision, and result in pixilated porosity maps that lack high-resolution detail. An automated process that can save time and improve resolution is often desired. In this study, the automated thin section analysis constructs detailed porosity maps in a matter of seconds, but generates greater estimates of healed porosity when compared to point counts (Figure 3.2.2; Appendix C). The healed minerals in the Newberry samples are predominately calcite and silica consisting of chalcedony, opal, and quartz. In transmitted light, these minerals share the same shades of white as the ubiquitous plagioclase of the host rock. When point counting, this issue is quickly resolved by researcher-decision, however the automated analysis fails to differentiate the two materials. The automated algorithm appropriately categorizes the healed minerals as healed porosity, but in doing so incorrectly includes the plagioclase grains, resulting in an overestimation of healed porosity. Staining the samples for plagioclase, though not a perfect solution to the problem, could improve the accuracy in

discriminating between these minerals. This study concludes that while the automated process saves time and improves resolution of porosity maps, it neglects to incorporate the conscious researcher decisions that improve the uniqueness of mineral identification. Identification can be misleading and result in overestimations of healed porosity values if the healed minerals share a similar color as the host rock.

4.1.2 High Resolution Images vs. Micro CT Scans

Obtaining porosity measurements at a high resolution is often desired to increase porosity map detail, to resolve the smallest pores possible, to increase the accuracy of porosity measurements, to characterize the surface area exposed to fluids, and to investigate the micromechanics of fracture slip, among many goals. Constructing porosity maps from high-resolution stitched images, as this study has done, provides such detail and is a logical alternative to point counts. This method is very time intensive, laborious, and requires photo editing software. Once prepared, however, it is relatively inexpensive to run. Constructing the high-resolution stitched images and their porosity maps, however, takes up to a full day or longer for each sample. The pixel-selection process can be painstakingly slow, but because researcher decision is used to manually map contiguous pixels and define pore structures on such a high resolution, the porosity maps that result are highly detailed and reliable. These maps, however, lack insight into the 3D size and connectivity of pores since they are only a 2D slice of a 3D material. Micro CT scans can fill in this gap, but this study found this method to have disadvantages of its own.

Micro CT imaging ultimately results in a 3D porosity map that provides insight into the size, shape, and connectedness of pores—attributes that 2D methods fail to provide. While micro CT scans provide a new insight into pore structure, they are time consuming and expensive. Each CT scan takes up to 6 hours including post-processing time, and can be relatively expensive depending on the desired number of samples.

Though previous studies found that differentiating geologic materials is often straightforward and intuitive because material boundaries correlate to density transitions (e.g., Ketcham and Carlson, 2001) this study found that this is not always the case. Because micro CT scans rely on material density, materials that lack sufficient density contrasts are difficult to successfully post-process. In particular, the density contrast between the host rock minerals (mainly plagioclase) and the healed minerals (calcite and silica varieties) at Newberry is relatively low (Figure 2.6.4). Although a density contrast could be seen between these materials and the host rock, differentiation during post-processing was (1) vague and not straightforward and (2) limited by the grain size of the materials. When thresholding for healed porosity, the non-uniqueness of the healed mineral densities causes a significant portion of host rock to be incorrectly included in the healed mineral component. To deal with this over-estimation of healed porosity, the selected host rock grains were despeckled out of the dataset, but so too were some healed grains (Figure 4.1.1). Major problems arise when the healed grains become smaller and more isolated, because they become even harder to differentiate from the host rock grains. For example, it is easy to identify large grains of interconnected calcite or silica (Figure 4.1.1) because of the space that they encompass; however when these materials approach the same size as the surrounding host rock, differentiation is nearly impossible

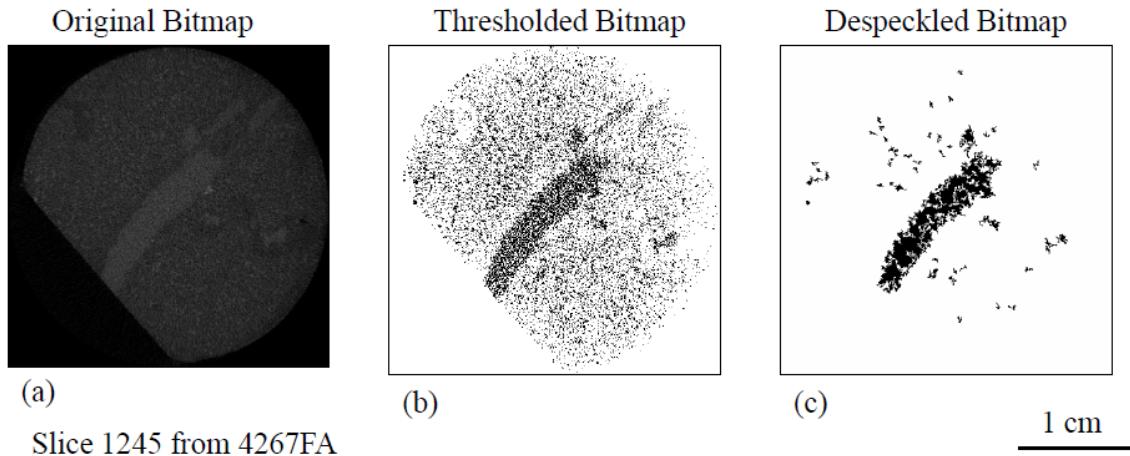


Figure 4.1.1: Thresholding and despeckling images from micro CT post-processing; (a) Unprocessed bitmap; (b) Bitmap thresholded for calcite vein. Notice how a majority of the host rock is incorporated into the threshold; (c) Despeckled bitmap post-thresholding. Notice how pixels within the calcite vein were omitted.

and the grains are often omitted. A best-guess method needs to be employed, and even that, this study found, results in gross underestimations of healed porosity. In contrast, open porosity, because of the high density contrast, is clearly identified.

Some micro CT transects contain anomalously high peaks in porosity data that cannot be accounted for when viewing a scanned image of the rock core (Figure 3.2.2). The 3D image obtained in a micro CT scan is comprised of thousands of horizontal 2D slices. The core image that results is similar to a deck of cards—each card supplying a data point for porosity information. To gain insight into what causes porosity peaks that are seemingly unaccounted for from a 2D view, individual bitmap slices were analyzed. As seen in Figure 4.1.2, the transect being analyzed runs through the center of the core and encounters porosity structures not seen from a front-on 2D view of the core, thus providing insight about the sample that a 2D method would fail to provide. In particular, the upper-most peak in open porosity and the bottom-most peak in healed porosity is

attributed to an open pore and filled vesicle that exist in the middle of the sample, respectively.

On every micro CT transect, the skeletal porosity is amplified at the top and bottom of the sample. In particular, open porosity peaks at the tops of the samples and healed porosity peaks at the bottoms of the samples. While in the micro CT scanner, the top of the core is exposed to low-density open air and the bottom of the core rests on a high-density brass plate. As the x-rays approach and leave the core during the scanning process, the open air and brass holder influence the attenuation values, resulting in falsely-peaked edge effects seen in the data. Though a next step in this research involves finding a correction factor to resolve this problem, cropping off the ends of the transects and focusing on the region not influenced by edge effects is a temporary solution to the problem.

The shadow zones that appear in micro CT scans around pyrite grains in the Newberry samples (Figure 3.2.4) result from the extreme attenuation of x-rays that would be expected from passing through a high density material such as pyrite. The incoming x-rays attenuate so much that they fail to pass through the pyrite grains, resulting in shadow zones that can be mistaken for open porosity. Since large pyrite grains were not common in the samples, choosing a transect location that did not produce a shadow zone was relatively common. Caution should be taken if micro CT is employed on samples containing numerous high-density minerals.

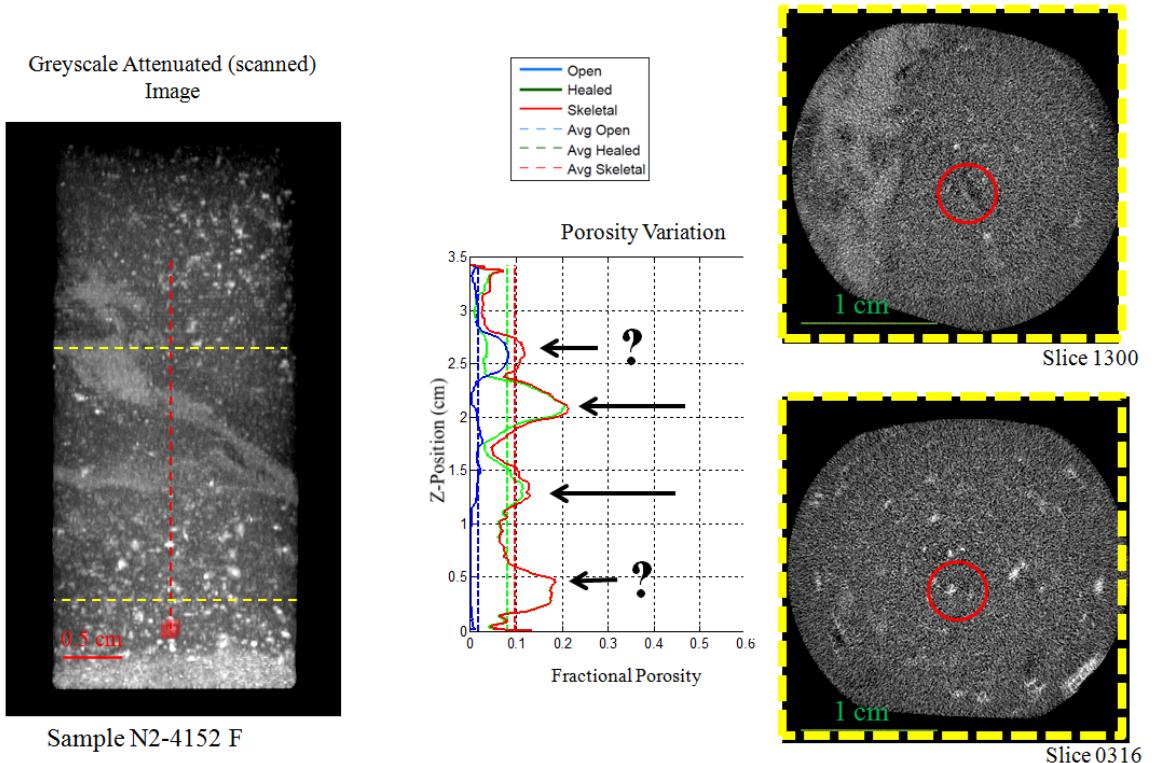


Figure 4.1.2: Further investigation of unexplained porosity peaks. Yellow dashed lines represent locations of 2D slices that were extracted from the 3D data. Yellow Dashed boxes are the 2D slices. Circled in red are features that are not revealed from traditional 2D approaches.

Lastly, the pixel size of the micro CT scans ($26.7 \mu\text{m}$) is much larger than the pixel size of the high-resolution analyses ($\sim 1.04 \mu\text{m}$). Because of this size difference, the high-resolution image stitches can accurately resolve pores on a smaller scale. Pores that are smaller than the pixel size of the micro CT imaging therefore yield pixel greyscale values that are inaccurate. What results is known as the partial volume effect, whereby the volume of a pixel is comprised of more than one distinct material, yielding a greyscale value that represents an average of their properties (Ketcham and Carlson, 2001). Though methods have been developed to interpret this effect (e.g., Johns et al., 1993), it remains difficult to reliably differentiate the fine details when multiple materials

are contained in a single pixel. These inaccuracies may attribute to some of the differences seen in the porosity data between the two techniques. Table 4.1.1 is a brief summary of the advantages, disadvantages, and pixel sizes of each method discussed above.

Table 4.1.1

Brief summary of the advantages, disadvantages, and pixel size of each method evaluated in this study

Method	Advantages	Disadvantages	Pixel Size
Point Count	Inexpensive (time and labor) Straightforward Common Relatively Quick Insight into porosity evolution Insight into mineralogy/textures of grains	Very pixelated/low resolution -No insight into true size/connectivity of pores -Does take a few hours	1 mm
Automated Image	-Inexpensive (time and labor) -Much higher resolution than point count -Takes minutes	-Writing the algorithm can take time -Relies on transmitted light of minerals (may overestimate porosity values) -No insight into true size/connectivity of pores	20.7 µm
High Resolution Image	-Inexpensive (time and labor) -Highest resolution of my methods -Conscious decisions made by researcher -Yields detailed, reliable porosity maps	-Very time intensive -No insight into true size/connectivity of pores	1.04 µm
Micro CT	-Yields a 3D porosity map -Insight into pore size and connectivity -Insight into anisotropy measurements	-Expensive (time and labor + scan cost) -Time -Resolution lower than some grains or pores -Corrections (edge effects, pyrite shadowing, etc.) -Thresholding/Processing Problems	26.7 µm

4.2 Geologic Context

4.2.1 Key Attributes of Developing Fracture Porosity Revealed by Analyses

The results of this study show that as fracture stage increases from 2 through 5, there is a general increase in skeletal porosity (Figure 4.2.1). This trend is consistently documented by each of the methods evaluated in this study, and is consistent with separate analyses conducted by Fetterman and Davatzes (2011) based on both macro- and microscopic porosity mapping. Fetterman and Davatzes (2011) relied solely on hand sample mapping at the millimeter to decimeter length-scales and semi-automated analysis of petrographic thin sections. They note that the former methods underestimate open porosity since most of the open pores are sub-millimeter in scale. This study has shown that the latter method overestimates healed porosity because of the non-uniqueness of the color of pore filling minerals, which if present in other rock types, will cause the same discrepancy. The combined analyses of porosity measurement in this study reveal that the high-resolution image transect data characterize details of porosity occurrence near the individual pore-length scale, yielding new details about fracture reactivation, relative ages of pore-filling cements, surface roughness of reactivated fractures, dilation, and thus healing history. For example, in Figure 4.2.2, the healed porosity transect of a stage 4 fracture in sample N2-3523.5 WFA peaks to and remains at 100% as the window moves across the fracture trace, with the only open porosity occurring at the margins or outside the fracture. This spatial distribution suggests that complete healing of the fracture “core” is consistent with either a lack of reactivation or with the inability of this well-developed fracture to dilate if reactivated. It also suggests that sufficient cement could at one time be transported to this fracture, but porosity analysis alone is unable to

distinguish whether the cements are locally sourced, represent a dissolution and re-precipitation of cataclastic fault rocks, or require transport from distances much greater than the fracture thickness. X-ray fluorescence (XRF) analysis of elemental chemistry, especially relatively immobile elements including titanium and zirconium, suggests that the composition of this material is indistinguishable from the host rock on either side of the fracture, and thus does not require long transport of fluids to support the cementation (Fetterman and Davatzes, 2011), although some intermediate stage fractures do. Fetterman and Davatzes (2011) further suggest that dilation is suppressed in this case as the fracture filling far exceeds the topography of the fractures walls that could facilitate dilation.

In contrast, Figure 4.2.3 shows a stage 3 to 4 fracture in sample N2-4267 FA that contains both open and healed porosity. The open porosity is located in the middle of the fracture (see red circle in the left panel of Figure 4.2.3) and is elongated parallel to the walls of the host rock contact with the healed porosity and is much thinner in the fracture-perpendicular direction suggesting it resulted from relatively recent fracturing. Similar to the previous example, open porosity also occurs on the margin of the fracture as well as in the host rock away from the fracture. The occurrence of open porosity in some of the fractures at Newberry suggests that recent dilatant fracturing has occurred and is confirmed petrographically by the truncation of pore-filling cement layers. This relationship, evident in high-resolution image analysis, suggests that at least some fractures within andesite and basalt at Newberry are prone to dilatant failure. Their occurrence is necessary to support geothermal circulation because they can provide open pathways for fluid flow. However, conductive temperature gradients indicating a lack of

convection (Cladouhos et al., 2012) suggest that there is insufficient naturally connected porosity. This crack-porosity also suggests that EGS style stimulation could promote self-propagating shear failure to increase both the volume and connectedness of pore space to create a reservoir in this hot rock.

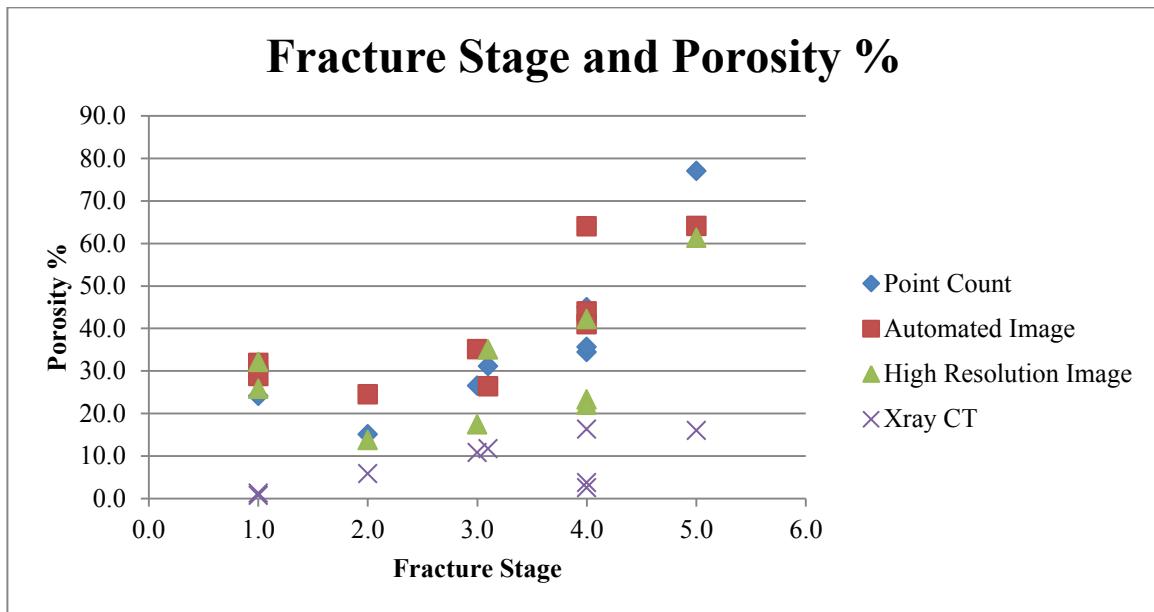


Figure 4.2.1: Fracture stage vs. percent skeletal porosity.

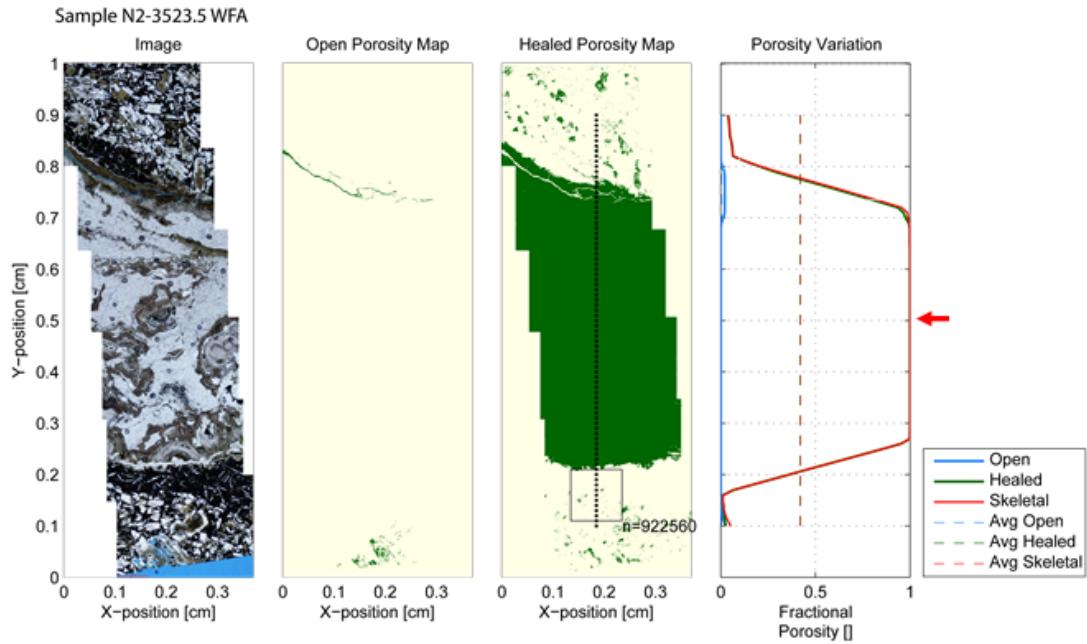


Figure 4.2.2: High resolution image scan showing a completely healed fracture. Red arrow points to peak porosity at 100%.

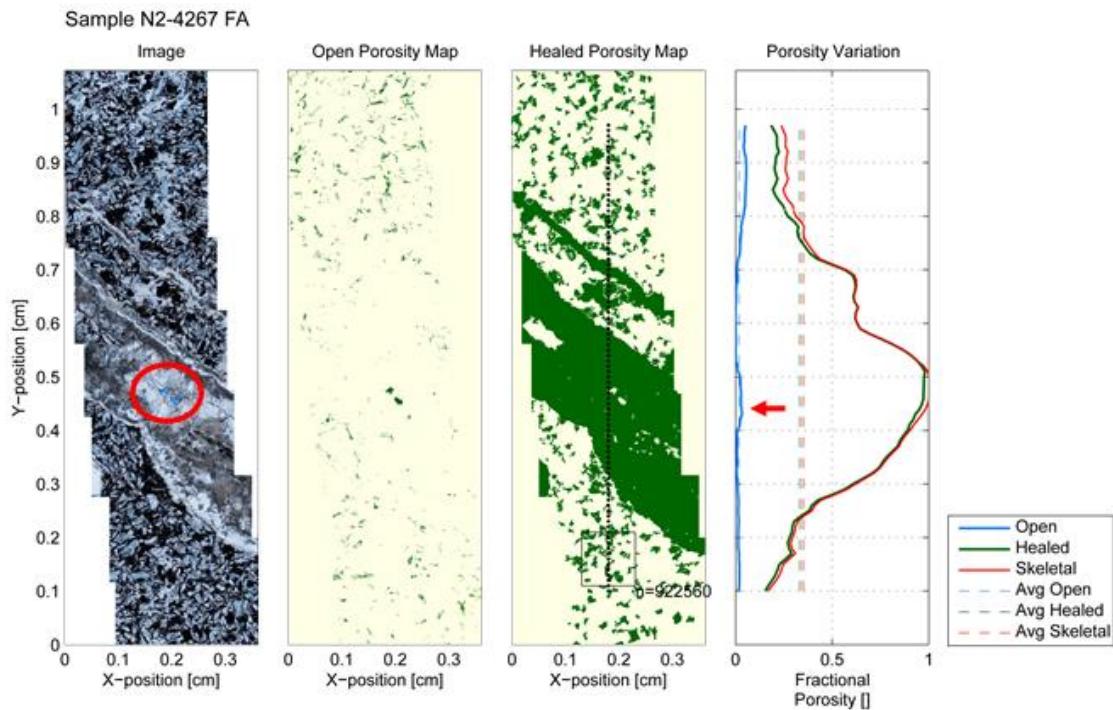


Figure 4.2.3: High resolution image scan of a fracture that contains open porosity. Red circle and arrow point to the location of open porosity and the corresponding peak in the transect data, respectively.

Although encouraging, these two-dimensional analyses provide limited insight to pore connectivity within and surrounding natural fractures. Similarly, the size distribution of these pores is not well constrained. Micro CT scans provide additional insight. Pixel-size limitations of the micro CT scans imply that larger, connected pores can be imaged, but small, isolated pores less than 25-75% of the 26.7 micron pixel size are likely to go undetected. Even for larger pores, since porosity is defined within a sampling window designed to contain hundreds of pixels, isolated pores just a few pixels in width will make only small changes to the measured porosity along a sample transect. Thus, there is a discrepancy between the 2D and 3D porosity measurements, and the pores visible in the high-resolution thin section porosity maps (compare Figures 4.2.2, 4.2.3 and 4.2.4, as well as the summary in Figures 4.2.1). Despite these limitations, the transects show that the lower stage fractures (stage 2) are characterized by open porosity distributions relative to fracture position that have very small peaks and even smaller average open porosity (Figure 4.2.4). As the fracture stages increase, porosity peaks and the fraction of open porosity also increases (Figure 4.2.4).

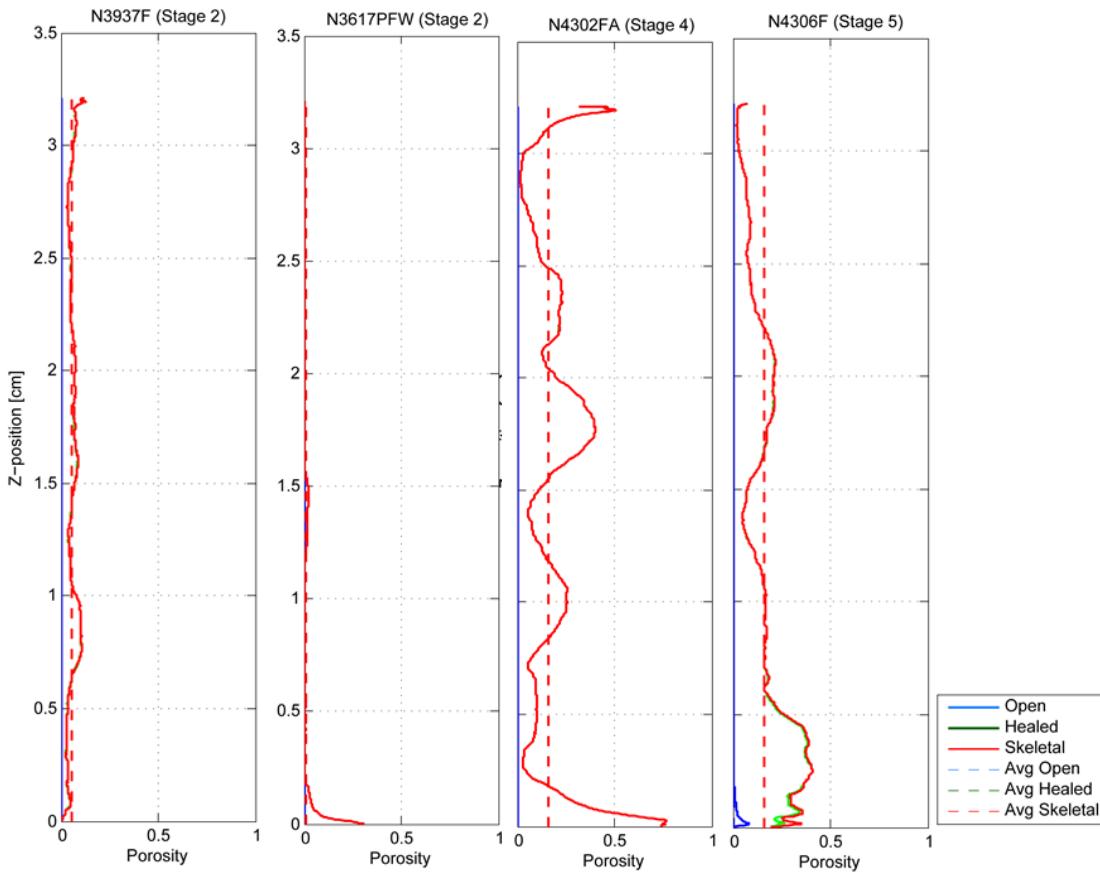


Figure 4.2.4: Micro CT scans showing low open porosity. Cube size = 1x1x1mm; 1 mm³

4.2.2 Fracture Surface Roughness

The geometry of the fracture surfaces and its correlation with dilation accompanying repeated slip reveals 19 dilatant slip events resulting in 38 surfaces. These surfaces fall into three distinct groups, A-C, that suggest 3 distinct episodes (ages) of fracture growth (Figure 3.4.1), each of which sustained multiple slip events. Segment A is interpreted to represent the initial group of fracturing that continued to dilate and eventually give rise to segment B, as suggested by layering and cross-cutting relationships visible petrographically and by distinct color variation under the cathodoluminescence. Segment B continues to slip and dilate until Segment C forms and

links to Segment B. Taking into account this relative timeline can determine how surface roughness evolved as a function of fracture development. Figure 3.4.2a suggests that surface topography decreases after the initial break and remains relatively constant as slip recurs. The linkage of fractures, however, introduces greater topography, which gradually decreases as the linked fractures continue to slip.

The along-surface length of the fracture surfaces (L_s) was compared to a straight-line length (L_{min}) (Figure 3.4.2b). Surface length relates to the tortuosity of the void space within the fracture—that is, the physical length that fluid would take when traveling through the fracture—and the surface area available to sustain dissolution/precipitation reactions. After the initial break, the L_s/L_{min} ratio increases and gradually decrease as slip continues. Once fracture linkage occurs, the L_s/L_{min} ratio increases again, and continues to increase as the linked surface continues to break. Note here that variation in L_{min} results from the fact that during most fracturing events only a portion of the fracture breaks open. Lastly, because asperity frequency and grain size share a common length scale, it may be possible to predict the dilating effects of induced fractures on an otherwise unfractured rock.

4.2.3 History of Dilatancy and Implications for EGS

The results of this study shed light onto the geologic and mechanical history of the rocks at Newberry—a history that, if dominated by dilatant failure and repeated fracture slip, can be useful to Enhanced Geothermal Systems (EGS) stimulation. EGS stimulation through the injection of fluids stresses a system by lowering the effective normal stress and thus reduces frictional resistance to slip. If lowered sufficiently, the *in*

situ stresses will cause some fractures to slip. Once flow through a fracture is then started, thermal contraction due to cooling from the colder injection water is likely to reduce the frictional contact strength of asperities (e.g., Dempsey et al., 2013). The key question is whether this approach will replicate the conditions that naturally produced dilatant failure, assuming that dilatant failure has occurred naturally in the past. This study shows that the rocks at Newberry are in fact dominated by dilatant failure and repeated fracture slip, thereby being applicable for EGS stimulation.

The behavior of the initial fracture in the rock at Newberry is largely dictated by the initial porosity present prior to the fracture event. Initial porosity refers to the initial pore space associated with the formation of a rock, and can include intergranular pores, vugs, and vesicles. This initial porosity influences the failure behavior of rock, including the initiation of failure and its subsequent evolution during deformation. Ultimately, the initial porosity helps determine if the applied stress and resulting deformation will result in the dilatancy or compaction of pore space. In EGS experiments, dilatant behavior during deformation is desired to increase overall storativity, pore connectivity, and surface area of contact to enhance heat transfer from the rock to a percolating fluid. Thus understanding the initial porosity, in conjunction with porosity developed due to the natural fracturing process of Newberry rock, will document the past potential for dilatancy and provide critical context for assessing the ongoing EGS demonstration project currently being conducted (e.g., Petty et al., 2013; Cladouhos et al., 2013), (see Section 4.2.4 below).

Rock characterized by high initial, open porosity, typically compact as it approaches brittle failure (Edmond and Paterson, 1972; Brace, 1978; Jamison and Teufel,

1979; Wong et al., 1997) whereas lower initial porosity promotes dilatation approaching brittle failure (e.g., Fredrich et al., 1989). As rock is loaded to failure, it initially experiences minor, elastic pore closure. As differential stress increases, however, permanent deformation via cracking develops. For relatively high porosity rocks, this latter stage can result in pore destruction as grains are crushed. In relatively low porosity rocks, crack growth and formation of a through-going fracture continues to be dilatant. (Zhu et al., 2007; Fortin et al., 2009; Schofield and Wroth, 1968) (Figure 4.2.5).

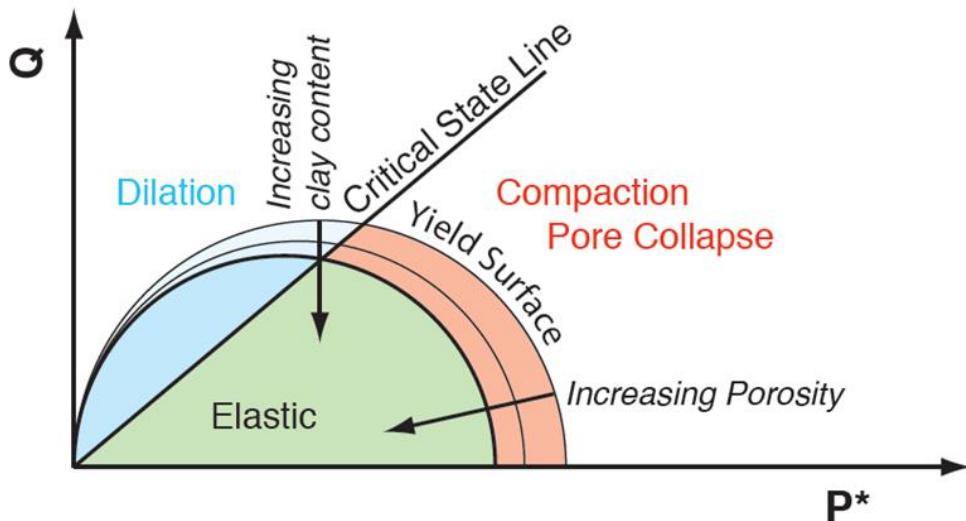


Figure 4.2.5: Model of tendency toward dilation, constant volume shearing along the critical state line, or compaction due to stress boundary conditions. (Summarized as the effective mean normal stress (P^*) and differential or deviatoric stress (Q) and initial rock porosity structure). (See Paterson and Wong (2005) for additional details.)

Different estimations of the initial pore space needed for dilatancy during brittle failure have been investigated in the past. The boundary between compact and porous rock, however, is still debated. Brace (1978) suggested that a minimum of 5% void space

is necessary to consider a rock porous, whereas Baud et al. (2000) suggests this cutoff is as low as 3%. Others suggest 15-20% in sandstone as related to the formation of deformations bands (e.g., Eichhubl et al., 2010). Although the porosity at which rock transitions from dilatant to compactant varies due to rock properties and stress state, in general the behavior is defined by clay abundance, overall mineralogy, grain size, and initial primary porosity. For the rocks in this study, since clay abundance is highly temperature dependent, it has minimal impact at temperatures of 200-250 °C, which occurs at relative shallow depth (Figure 1.3.2). The overall mineralogy of the rocks is dominated by plagioclase, magnetite, clinopyroxene, quartz, potassium feldspar, and chalcopyrite (Bargar and Keith, 1999; Fetterman and Davatzes, 2011). Limited grain size analysis (Figure 3.5.1 e) shows that the groundmass of minerals less than 0.02 cm in size represents roughly 80% of the total. Lastly, this study found that the current open porosity at Newberry is usually in the range of 0% to 5% (Appendix B). Taking these considerations into account, and consistent with observations in the core, it is reasonable to postulate that failure at Newberry tends to be dilatant. Modifications to pore structure at more developed stages of fractures in which fault rocks develop will tend to be characterized by compactions, however since raising pore fluid pressure has little or no impact on the differential stress of the rock and merely reduces the effective normal stress, it is likely that dilatant failure will be promoted during an EGS stimulation.

The onset of fracture formation in the samples from Newberry is associated with dilation preserved by mineral precipitation within the fracture. Subsequent slip on the established fracture produces dilatancy as a result of the separation of the rough fracture surfaces as the asperities slide over each other, as well as the separation of microcracks

adjacent to the fracture. The relative development of slip and dilation is inferred from the layering and cross-cutting relationship of fractures. In the case of microcracks, relative timing is inferred from the corresponding mineral phase associated with cement layers in the main fracture that match the filling in the microcracks revealed in petrographic and cathodoluminescence analyses (e.g. Figures 3.5.1a and 4.2.3). Thus, the process of fracture formation and slip on simple, early stage fractures creates a locally maximized porosity. This behavior is consistent with dilatant behavior typical of low porosity rocks found in previous studies (e.g., Zoback and Byerlee, 1975; Stormont and Daemen, 1992; Zhang et al., 1994; Kiyama et al., 1996; Peach and Spiers, 1996; Zhu and Wong, 1999).

Within the dilated fracture after the initial break, porosity is reduced and pores become more isolated by mineral precipitation, predominantly of calcite and silica, including chalcedony, opal, and quartz. Layers of cement, as well as fracturing of these layers, reveal that these healed regions were subjected to a repeated cycle of shearing, dilating, and healing. The repeated formation of new fracture surfaces can be mapped from the distinct petrographic relationships and the character of the cathodoluminescence which aids in the correlation of cementation events associated with the same fluid source. As fracture stage increases, there is an increase in the amount of healed porosity and the number of generations of cements. The formation of breccia within the cement results in a fracture zone rather than a distinct fracture surface. Throughout this cycle, cements are primarily comprised of either calcite or silica and result in very low remnant open porosity. Both of these conditions promote future dilatancy (Davatzes and Hickman, 2010). Because the present open porosity values remain, in most samples, less than 5%, dilatant behavior during failure that increases permeability should still be expected.

Thus, if fluid pressure can be delivered to these fractures to enable reactivation in shearing (i.e., causing slip), the naturally fractured rock at Newberry should generate self-propagated fractures causing a persistent increase in permeability, as necessary for successful EGS.

Well-developed natural fractures analyzed in core (Figure 2.5.1) are associated with surrounding zones of high crack density, constituting a damage zone. Chester and Logan (1986) argue that these cracks result from both the fracture nucleation and growth processes, as well as continued slip on a rough fracture. This is also generally brought out by acoustic emission monitoring of triaxially deformed samples (e.g., Lockner, 1993). Figure 2.5.1 shows such a damage zone from Newberry. This damage zone has the potential to accumulate and link together into further discontinuities, thereby decreasing the strength of the rock and making the fracture more susceptible to reactivation (Moore et al., 2009). At Newberry, damage zones are evident at relatively early stages of fracture formation and persist at later stages at the periphery of the fault rock (Fetterman and Davatzes, 2011). The presence of a damage zone surrounding the main fracture surface (or of fault rock) has two key implications: (1) it provides the potential to maintain open, connected porosity, promoting high permeability parallel to the fault, although cross-fault permeability might or might not be preserved, and (2) the additional fractures help to maximize the surface area of contact between percolating fluids and the rock, thereby increasing the heat exchange between the fluid and the rock—a situation that could significantly increase the proportion of extractable heat within an EGS.

The detection of the damage zone is sensitive to the method of porosity measurement (e.g., point count, image analysis, micro CT) and the evaluation methods

(e.g., transect vs. image mapping). This study found that different techniques tend to characterize damage zones differently, sometimes even omitting their presence altogether. Direct thin section analysis successfully allows for the damage zone to be analyzed, though the point count maps generated in this study are too pixelated to definitively distinguish such zones, as seen in Figure 3.2.1 b and Appendix C. The automated image analysis method in this study allows for porosity maps to be generated that can capture aspects of the damage zone, but again, the non-uniqueness of mineral colors (discussed above) may produce misleading porosity maps. In this study, micro CT scans do not successfully map the presence of damage zones, however, Nasseri et al. (2009) succeeded in mapping fracture damage zones using micro CT scans. This difference appears to largely reside with the density contrast between open pore space, healing minerals, primarily rock forming minerals, and the pixel size. For transect-based analysis, window size is an additional contributing factor. In this study, the majority of the open porosity is healed, minimizing density contrast, whereas Nasseri et al. (2009) studied the open cracks in granitic rock. Since the density contrast between the healed minerals and the host rock at Newberry is relatively small (discussed above), and because the healed minerals in the damage zones have grain sizes that approach the pixel size of the micro CT scans, this study finds that when mapping healed damage zones of fine-grained volcanic rock, micro CT scans fail to produce accurate, resolvable results.

High resolution image analyses therefore provide the best means of mapping fracture damage zones. In the transect analysis of the mapped porosity structure, changes in damage zone porosity can be seen as the sampling window moves across the damage zones and fractures. The critical issues here are whether the chosen window size is small

enough to resolve high detail (Figure 2.5.2) and the orientation of the transect relative to the fracture. For the purposes of comparing image and micro CT approaches in this study, the window size was most often relatively large in edge-length compared to the fracture thickness to capture a large number of pixels for calculating porosity. In this case the fracture can appear to be abnormally large and a function of the window edge-length (the peak porosity is also suppressed to lower maximum values), and no distinct damage zone can be recognized. However, smaller window sizes obtainable in the high resolution images are capable of resolving the damage zone (e.g., Figure 2.5.2).

High resolution image analyses reveal that the majority of the damage zones at Newberry remain healed, whereas secondary open porosity resides in the central part of the main fractures in the early stages. This suggests that damage zones, though created in the rock at Newberry and initially a conduit for fluid flow, do not reactivate and dilate upon earthquake stimulation, suggesting the openings between fracture surfaces to be the main conduit of EGS fluid stimulation fluid. However, the most developed fractures which contain a core of fine-grained fault rock only retain the potential for dilation in the relatively brittle, low porosity damage zone.

4.2.4 Correlations to EGS Induced Seismicity

Recently, injection into NWG well 55-29, just 0.5 km from the GEO N-2 Corehole, induced seismic events in the vicinity of the cores examined in this study (Figure 4.2.6). In Figure 4.2.6, the stress model derived for the NWG55-29 well shows the attitude of the principal horizontal stresses as thick black lines, where one standard deviation of uncertainty is indicated by the yellow wedges (Davatzes and Hickman,

2011). The length of these lines is scaled to a reference vertical stress indicated by the red circle. Since both of the black lines indicating the relative magnitude of the principal horizontal stresses are shorter than the diameter of the circle, a normal faulting stress regime is indicated. The inner blue circle indicates the relative magnitude of the static, equilibrated pore fluid pressure. Note that the earthquakes are roughly aligned with the maximum horizontal stress direction, also consistent with normal faulting and locally mapped fissure alignments and fault scarps (see Cladouhos et al., 2011). The cross sections indicate that micro-earthquake activity occurred in the vicinity of GEO-N2 both in terms of map position and depth relative to the core samples. Seismic slip is associated with slip weakening behavior typical of brittle, dilatational coulomb failure (Brown, 1987; Lockner and Beeler, 2002). In most cases, seismic events associated with injection wells are presumed to result from the reduction of normal tractions on natural fractures accompanying the invasion of the injection fluid. This process lowers the normal traction that contributes to both the frictional resistance to slip and the maintaining of crack closure. The occurrence of induced earthquakes at the position of the natural fractures analyzed from core samples supports the potential correlation between the dilatant failure modes characterized in the core samples and induced slip. This provides the opportunity to correlate the combined geologic natural fracture characterization and rock mechanics measurements extrapolated along GEO N-2 using the geophysical logs with earthquake behavior.

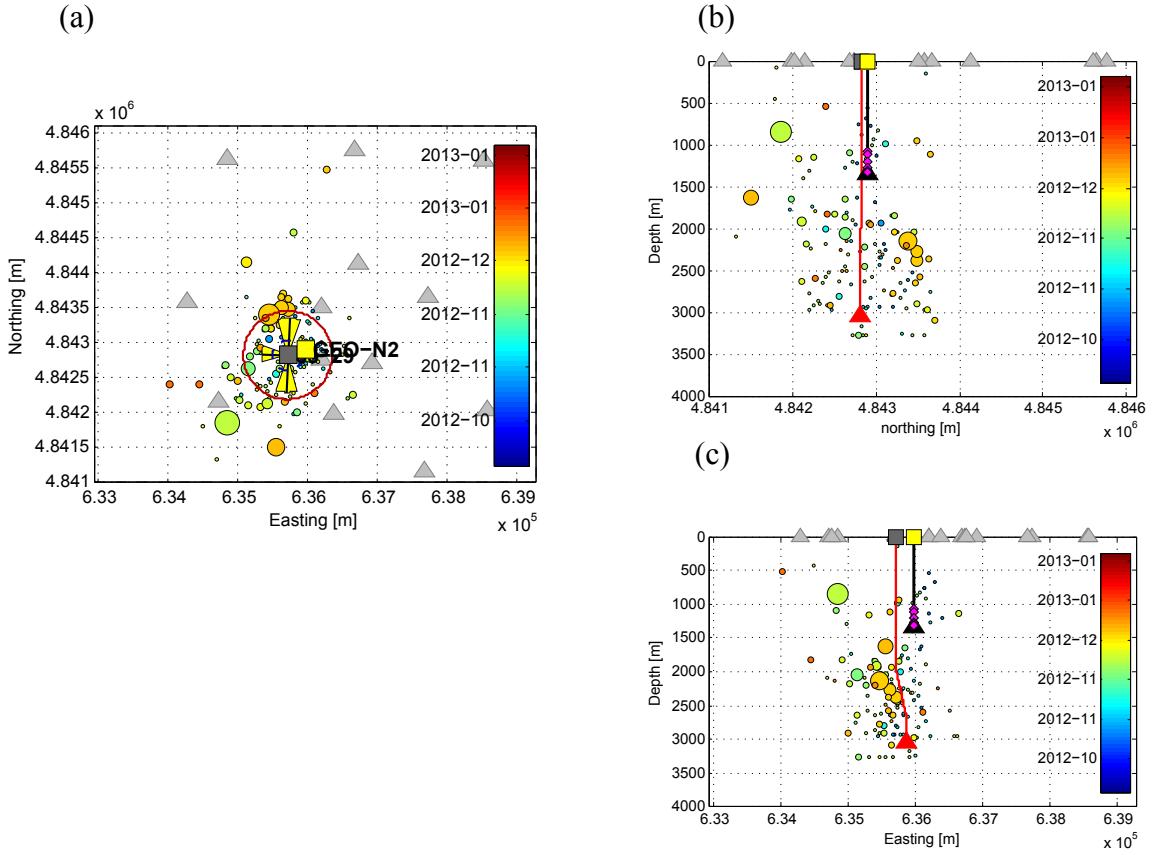


Figure 4.2.6: Induced seismic events in the vicinity of Newberry Core; (a) Map of initial catalog earthquake locations (circles color coded by date) maintained by the Lawrence Berkeley National Laboratory, University of California, Berkeley, with seismic stations indicated by gray triangles; (b) S-N and (c) W-E cross-sections of seismicity show in (b) and (c). Core locations in GEO-N2 (yellow square) are shown as magenta diamonds. Color code is as in Figure (a).

CHAPTER 5

CONCLUSION

Values of pore volume and structure of naturally fractured rock from the Newberry Volcano, Oregon differ significantly when measured by different techniques. This discrepancy is greatest for values of healed porosity and particularly for more highly fractured rock. At Newberry, the amount of healed porosity significantly exceeds open porosity, both in primary and secondary porosity.

Point counts provide insight into porosity evolution and the mineralogy and texture of grains, while at the same time yielding highly accurate and reliable porosity maps. Automated thin section thresholding, though faster and better resolved than point counts, tends to over-estimate healed porosity when compared to thin section point counts conducted on the same sample because of transmitted light color similarity. This discrepancy can be avoided in different rock types if healed and host minerals do not share similar colors. Because these two methods deal with 2D thin sections of rock cores, they fail to provide insight into the true size and connectivity of pores.

Detailed analysis of high-resolution images of petrographic thin-sections provide the most accurate means of porosity characterization. Though very time consuming, they provide the best resolved, most accurate porosity maps. They do not, however, provide detailed information about the true size and connectivity of the pores and should be augmented with data from micro CT scans. Though micro CT scans provide insight into the 3D size, connectivity, and distribution of pores, the scans were found to consistently yield different estimates of porosity amounts when compared to the high resolution image

stitches. They are also more expensive. The differences in porosity results can be attributed to the non-uniqueness of material density, scanning edge effects, and the large pixel size found in the micro CT method.

In summary, in order to recommend a single method over others for porosity analysis, the context in which it is used needs to be considered. If research requires solely an average of porosity within a sample, point counting will generate appropriate results. The process can be reliably automated only if the mineralogy between the host rock and porosity types is distinctly different color in transmitted light. If research seeks to establish details regarding the history of porosity evolution within a sample, point counts and high resolution image analyses will provide that context. Studies that investigate the true connectivity of the pores should use micro CT scans, however fine-grained and extensively healed samples should be augmented with petrographic analyses because of complications of pixel size and mineral density non-uniqueness, mentioned above.

The results from the combined analyses reveal that there is a general increase in skeletal porosity as a function of fracture maturity, coinciding with results from Fetterman and Davatzes (2011). New 3D transect data provide additional insight into pore size and connectivity, though pores smaller than 26.7 um tend to go undetected during the analysis. Despite these limitations, the transects show that the lower stage fractures are characterized by open porosity distributions relative to fracture position that have very small peaks and even smaller average open porosity values. As the fracture stages increase, porosity peaks and the fraction of open porosity also increases. This study shows that most of the porosity in Newberry rock is healed, suggesting that (1) not

much recent fracturing has occurred and (2) the pore space was once interconnected such that precipitating fluids could flow through it. Of the porosity that remains open, it occurs in the central parts of later stage fractures, primary vugs, and vesicles. The amount of open porosity in the Newberry samples is mostly under 5%, suggesting that dilatant behavior and increased permeability can be expected upon induced fracturing.

Fracture roughness measurements from Newberry show that even simple fractures sustain multiple opening events accompanying shearing. The size distribution of the >0.2mm size fraction of primary grains and pores strongly correlates with the roughness of fracture surfaces. Thus fundamental characteristic length scales associated with host rock (grain size and pore size, vesicles in this case) influence the potential for dilation through their control on fracture surface roughness. Preliminary findings suggest that this roughness directly influences dilation and progressively evolves both as a function of repeated slip and the linkage of cracks that occurs as a function of fracture development.

Lastly, the measurements of porosity in this study suggest that the mechanical history of basaltic and andesitic rocks at Newberry is comprised of multiple slip and dilation events, as evidenced by the presence of healed, different aged layers of calcite and silica, as well as healed damage zones surrounding main fractures. Textural and mineralogical analysis allowed by the evaluation of thin sections helped to identify calcite and silica cements, which tends to promote dilatant behavior upon brittle failure. High-resolution image analyses reveal that the damage zones in the vicinity of fractures that contain open porosity are mainly healed, suggesting that they do no reactivate upon fracture stimulation, implying stimulation fluids would likely flow though the reactivation of the main fracture instead.

REFERENCES CITED

- AltaRock (2011). Newberry EGS Demonstration Phase 1 Report. 290 p. plus 27 Appendices.
- Antonellini, M.A., Aydin, A. (1994). Effect of Faulting on Fluid Flow in Porous Sandstones: Petrophysical Properties. American Association of Petroleum Geologists Bulletin, p. 355-377.
- Antonelli, M.A., Aydin, A., and Pollard, D.D. (1994). Microstructure of Deformation Bands in Porous Sandstones at Arches National Park, Utah. Journal of Structural Geology, p. 941-959.
- Anderson, S.H., Peyton, R.L., Wigger, J.W., Gantzer, C.J. (1992). Influence of Aggregate Size on Solute Transport as Measured Using Computed Tomography. Geoderma, v. 53, p. 387-398.
- Arnold, J.R., Testa, J.P.J., Friedman, P.J., Kambic, G.X. (1982). Computed Tomographic Analysis of Meteorite Inclusions. Science, v. 219, 383-384.
- Bargar, K., Keith, T. (1999). Hydrothermal Mineralogy of Core from Geothermal Drill Holes at Newberry Volcano, Oregon. U.S. Geological Survey Professional Paper 1578, p. 2.
- Barton, N. (1973). Review of a New Shear Strength Criterion for Rock Joints. Engineering Geology, v. 7, p. 287-332.
- Barton, N. (1986). Deformation Phenomena in Jointed Rock. Géotechnique, v. 36.2, p. 147-67.
- Barton, N. (2007). Rock Quality, Seismic Velocity, Attenuation and Anisotropy. Taylor & Francis Group, London, UK, 729 p.
- Baud, P., Zhu, W., Wong, T.F. (2000). Failure Mode and Weakening Effect of Water on Sandstone. Journal of Geophysical Research, v. 105 n. B7, p. 16371-16389.
- Brace, W. F. (1978). Volume Changes During Fracture and Frictional Sliding: A Review. Pure Applied Geophysics, v. 116 p. 603-614.
- Brown, S., Scholz, C.H. (1985). Closure of Random Elastic Surfaces in Contact. Journal of Geophysical Research, v. 90, n. B7, p. 5531.
- Brown, S., Scholz, C.H. (1986). Closure of Rock Joints. Journal of Geophysical Research, v. 91, n. 5, p. 4939-4948.

- Brown, S. (1987). Fluid Flow Through Rock Joints: The Effect of Surface Roughness. *Journal of Geophysical Research*, v. 92, n. B2, p.1337-1347.
- Chester, F. M., Logan, J.M. (1986). Implications for Mechanical Properties of Brittle Faults from Observations of the Punchbowl Fault Zone, California. *Pure Applied Geophysics*, v. 124, p. 79-106.
- Caine, J.S., Evans, J.P., Forster, C.B. (1996). Fault Zone Architecture and Permeability Structure. *Geology*, v. 24 n. 11, p. 1025–1028.
- Cladouhos, T. T., Petty, S., Callahan, O., Osborn, W., Hickman, S.H., Davatzes, N. C. (2011). The Rold of Stress Modelling in Stimulation Planning at the Newberry Volcano EGS Demonstration Project. *PROCEEDINGS: Thirty-Sixth Workshop on Geothermal Reservoir Engineering*, Stanford University, Stanford, California, January 31 – February 2, 2011.
- Cladouhos, T. T., Osborn, W.L., Petty, S., Bour, D., Iovenitti, J., Callahan, O., Nordin, Y., Perry, D., Stern, P. L. (2012). Newberry Volcano EGS Demonstration--Phase 1 Results. *PROCEEDINGS: Thirty-Seventh Workshop on Geothermal Reservoir Engineering*, Stanford University, Stanford, California, January 30 – February 1, 2012.
- Cladouhos, T.T., Petty, S., Nordin, Y., Moore, M., Grasso, K., Uddenberg, M., Swyer, M., Julian, B., Foulger, G. (2013). Microseismic Monitoring of Newberry Volcano EGS Demonstration. *PROCEEDINGS: Thirty-Eighth Workshop on Geothermal Reservoir Engineering*, Stanford University, Stanford, California, February 11-13, 2013.
- The Committee on Fracture Characterisation and Fluid Flow. (1996). U.S. National Research Council, *Rock Fractures and Fluid Flow: Contemporary Understanding and Applications*, National Academy Press, Washington D.C., 551 p.
- Conroy, G.C., Vannier, M.W. (1984). Noninvasive Three-Dimensional Computer Imaging of Matrix-Filled Fossil Skulls by High Resolution Computed Tomography. *Science*, v. 226, p. 456–458.
- Constantz, B., Ison, I., Fulmer, M., Poser, R., Smith, S., VanWagoner, M., Ross, J., Goldstein, S., Jupiter, J., Rosenthal, D. (1995). Skeletal Repair by in Situ Formation of the Mineral Phase of Bone. *Science*, v. 267, n. 5205, p. 1796-1799.
- Cook, R., Hover, K. (1999). Mercury Porosimetry of Hardened Cement Pastes. *Cement and Concrete Research*, v. 29 p. 933-943.
- Crawford, B.R., Myers, R.D., Woronow, A., Faulkner, D.R., Rutter, E.H. (2002). Porosity-Permeability Relationships in Clay-bearing Fault Gouge. *Society of Petroleum Engineers*, p.1-13.

- Curewitz, D., Karson, J.A. (1997). Structural Settings of Hydrothermal Outflow: Fracture Permeability Maintained by Fault Propagation and Interaction. *Journal of Volcanology and Geothermal Research*, v. 79 n. 3-4 p. 149-168.
- Davatzes, N.C., Hickman, S.H. (2010). Stress, Fracture, and Fluid-Flow Analysis Using Acoustic and Electrical Image Logs in Hot Fractured Granites of the Coso Geothermal Field, California, U.S.A.. AAPG Memoir v. 92, Ch 24, p. 1 – 35.
- Davatzes, N.C., Hickman, S.H. (2011). A Preliminary Analysis of Stress in the Newberry EGS Well NWG 55-29. *GRC Transactions*, v. 35, p. 323-332.
- Dempsey, D., Kelkar, S., Lewis, K., Hickman, S., Davatzes, N.C., Moos, D., Zemach, E. (2013). Modeling Shear Stimulation of the EGS Well 27-15 Using a Coupled Thermal-Hydrological-Mechanical Simulator. ARMA 13-608, San Francisco, CA June 23-26, 13 p.
- Dieterich, J., Kilgore, B.D. (1996). Imaging Surface Contacts: Power Law Contact Distributions and Contact Stresses in Quartz, Calcite, Glass and Acrylic Plastic. *Tectonophysics*, v. 256 n. 1-4, p. 219-239.
- Edmond, J.M., Paterson, M.S. (1972). Volume Changes During the Deformation of Rocks at High Pressures. *International Journal of Rock Mechanics and Mining Sciences*, v. 9, p. 161-182.
- Eichhubl, P., Hooker, J.N., Laubach, S.E. (2010). Pure and Shear-Enhanced Compaction Bands in Aztec Sandstone. *Journal of Structural Geology*, v. 32, n. 12, p. 1873–1886.
- Facca, G., Tonani, F. (1967). The Self-Sealing Geothermal Field. *Bulletin Volcanologique*, v. 30, n. 1, p. 271-73.
- Fetterman, J.A., Davatzes, N. C. (2011). Evolution of Porosity in Fractures in the Newberry Volcano Geothermal System, Oregon, USA: Feedback Between Deformation and Alteration. *GRC Transactions*, v. 35, n. 1, p. 339 – 346.
- Fitterman, D., (1988). Overview of the Structure and Geothermal Potential of Newberry Volcano, Oregon. *Journal of Geophysical Research*, v. 93, n. B9, p. 10059-10066.
- Fortin, J., Stanchits, S., Dresen, G., Gueguen, Y. (2009). Acoustic Emissions Monitoring During Inelastic Deformation of Porous Sandstone: Comparison of Three Modes of Deformation. *Pure Applied Geophysics*, v. 116, p. 823-841.
- Fourie, S., (1974). The Cranial Morphology of *Thrinaxodonliorhinus* Seeley. *Annals of the South African Museum*, v. 65, p. 337–400.

Fredrich, J., Evans, B., Wong, and T.F. (1989). Micromechanics of the Brittle to Plastic Transition in Carrara Marble. *Journal of Geophysical Research*, v. 94, n. B4, p. 4129–4145.

Fredrich, J., Greaves, K., Martin, J. (1993). Pore Geometry and Transport Properties of Fontainebleau Sandstone. *International Journal of Rock Mechanics and Mining Sciences & Geomechanics Abstracts*, v. 30, n. 7, p. 691-697.

French, D., Ward, C.R., Butcher, A. (2008). Qemscan For Characterisation of Coal AND Coal Utilisation By-Products. Cooperative Research Centre for Coal in Sustainable Development. Research Report 93.

Grunewald, E., Knight, R. (2011). A Laboratory Study of NMR Relaxation Times in Unconsolidated Heterogeneous Sediments. *Geophysics*, v. 76, n. 4, p. G73-83.

Hearst, J.R., Nelson, P.H., Paillet, F.L. (2000). *Well Logging for Physical Properties: A Handbook for Geophysicists, Geologists, and Engineers*. John Wiley and Sons, Ltd, Chichester, West Sussex, England, 483 p.

Heijs, A.W.J., De Lange, J., Schouten, J.F.T., Bouma, J. (1995). Computed Tomography as a Tool for Non-Destructive Analysis of Flow Patterns in Macroporous Clay Soils. *Geoderma*, v. 64, p. 183–196.

Ingebritsen, S.E., Sanford, W.E. (1998). *Groundwater in Geologic Processes*. Cambridge University Press, Cambridge, UK, 345 p.

Jaeger, J.G., Cook, N.G.W., Zimmerman, R.W. (2007). *Fundamentals of Rock Mechanics*. Blackwell Publishing, Malden, MA, USA, 475 p.

Jamison, W.R., Teufel, L.W. (1979). Pore Volume Changes Associated with Failure and Frictional Sliding of a Porous Sandstone. *Proceedings of the 20th U.S. Symposium on Rock Mechanics*, p. 163–170.

Jensen, R. A., Donnelly-Nolen, J.M., McKay, D. (2009). A Field Guide to Newberry Volcano, Oregon. *The Geological Society of America: Field Guide* 15, p. 53-79.

Johns, R.A., Steude, J.S., Castanier, L.M., Roberts, P.V. (1993). Nondestructive Measurements of Fracture Aperture in Crystalline Rock Cores Using X-ray Computed Tomography. *Journal of Geophysical Research*, v. 98, n. B2, p. 1889–1900.

Jorgensen, S.M., Demirkava, O., Ritman, E.L. (1998). Three-Dimensional Imaging of Vasculature and Parenchyma in Intact Rodent Organs with X-ray Micro-CT. *American Journal of Physiology*, p. H1103-H1114.

- Ketcham, J., Carlosn, W.D. (2001). Acquisition, Optimization and Interpretation of X-Ray Computed Tomographic Imagery: Applications to the Geosciences. *Computers & Geosciences*, v. 27, p. 381-400.
- Kiyama, T., Kita, H., Ishijima, Y., Yanagidani, T., Aoki, K., Sato, T. (1996). Permeability in Anisotropic Granite Under Hydrostatic Compression and Triaxial Compression Including Post-Failure Region. *North America Rock Mechanics Symposium*, p. 1161-1168.
- Krauskopf, K. B., Bird, D.K. (1995). *Introduction to Geochemistry*. McGraw-Hill, New York, USA, 640 p.
- Kretz, R. (1993). A Garnet Population in Yellowknife Schist, Canada. *Journal of Metamorphic Geology*, v. 11, p. 101–120.
- Krumbein, W.C. (1935). Thin-Section Mechanical Analysis of Indurated Sediments. *The Journal of Geology*, v. 43, n. 5, p. 482-496.
- Lee, Y.J., Wiltschko, D.V. (2000). Fault Controlled Sequential Vein Dilation: Competition Between Slip and Precipitation Rates in the Austin Chalk, Texas. *Journal of Structural Geology*, v. 22, p. 1247-1260.
- Lockner, D.A. (1993). The Role of Acoustic Emission in the Study of Rock Fracture. *International Journal of Rock Mechanics and Mining Sciences and Geomechanics Abstracts*, v. 30, n. 7, p. 883-899.
- Lockner, D.A., Beeler, N.M. (2002). Rock Failure and Earthquakes. *International Handbook of Earthquake and Engineering Seismology*, Chapter 32, p. 505-537.
- Long, J. C. S., Gilmour, P., Witherspoon, P.A. (1985). A Model for Steady Fliud Flow in Random Three-Dimensional Networks of Disc-Shaped Fractures. *Water Resources Research*, v. 21, p. 1105-1115.
- MacLeod, N.S., Sammel, E.A. (1982). Newberry Volcano, Oregon: A Cascade Range Geothermal Prospect. *Oregon Geology*, v. 44, p. 123-131.
- Mavko, G., Mukerji, T., Dvorkin, J. (2009). *The Rock Physics Handbook*: Cambridge University Press, Cambridge, U.K., 339 p.
- Moore, J.R., Sanders, J.W., Dietrich, W.E., Glaser, S.D. (2009). Influence of Rock Mass Strength on the Erosion Rate of Alpine Cliffs. *Earth Surface Processes and Landforms*, v. 34, p. 1339–1352.

- Nasseri, M.H.B., Young, R.P., Rezanezhad, F., Cho, S.H. (2009) Application of 3D X-ray CT Scanning Techniques to Evaluate Fracture Damage Zone in Anisotropic Granitic Rock. Proceedings of the 3rd CANUS Rock Mechanics Symposium, Toronto, Paper 4050. 12 p.
- Olmstead, D.L., Wermiel, D.E., (1988). Newberry Crater Geothermal Resource Area, Deschutes County, Oregon: Oregon Department of Geology and Mineral Industries Open-File Report, n. O-88-3, scale 1: 24,000.
- Paterson, M.S., Wong, T.F. (2005). Experimental Rock Deformation: The Brittle Field. Springer, Berlin, Germany, 348 p.
- Peach, C.J., Spiers, C.J. (1996). Influence of Crystal Plastic Deformation on Dilatancy and Permeability Development in Synthetic Salt Rock. *Tectonophysics*, v. 256, p. 101-128.
- Petty, S., Nordin, Y., Cladouhos, T.T., Swyer, M. (2013). Improving Geothermal Project Economics with Multi-Zone Stimulation: Results from the Newberry Volcano EGS Demonstration. *PROCEEDINGS: Thirty-Eighth Workshop on Geothermal Reservoir Engineering*, Stanford University, Stanford, California, Feb. 11-13, 2013
- Peyton, R.L., Haeffner, B.A., Anderson, S.H., Gantzer, C.J. (1992). Applying X-ray CT to Measure Macropore Diameters in Undisturbed Soil Cores. *Geoderma*, v. 53, p. 329–340.
- Plas, L.v.D., Tobi, A. C. (1965). A Chart for Judging the Reliability of Point Counting Results. *American Journal of Science*, v. 263 n. 1, p. 87-90.
- Power, W.L., Tullis, T.E. (1992). The Contact Between Opposing Fault Surfaces at Dixie Valley, Nevada, and Implications for Fault Mechanics. *Journal of Geophysical Research*, v. 97, n. B11, p. 15425-5435.
- Power, W.L., Tullis, T.E. (1991). Euclidean Fractal Models for the Description of Rock Surface Roughness. *Journal of Geophysical Research*, v. 96, n. B1, p. 415-424.
- Rowland, S., Duebendorfer, E., Schiefelbein, I. (2008). Structural Analysis and Synthesis. Blackwell Publishing, Malden, MA, 301 p.
- Schofield, A. N., Wroth, C. P. (1968). Critical state soil mechanics. McGraw Hill, London, 310 p.
- Schreurs, G., Hanni, R. (1998). 4-D Analysis of Analogue Model Experiments. *American Association of Petroleum Geologists Bulletin*, v. 82, n. 10.

- Sonnenthal, E., Spycher, N., Callahan, O., Cladouhos, T., Petty, S. (2012). A Thermal-Hydrological-Chemical Model for the Enhanced Geothermal System Demonstration Project at Newberry Volcano, Oregon. PROCEEDINGS: 37th Workshop on Geothermal Reservoir Engineering, Stanford University, Stanford, California, January 30 -February 1, 2012.
- Stormont, J.C., Daemen, J.J.K. (1992). Laboratory Study of Gas Permeability Changes in Rock Salt During Deformation. International Journal of Rock Mechanics, Mining Science and Geomechanics Abstracts, v. 29, p. 325 -342.
- Wong, T.F., David, C., Zhu, W. (1997). The Transition from Brittle Faulting to Cataclastic Flow in Porous Sandstones: Mechanical Deformation. Journal of Geophysical Research, v. 102, n. B2, p. 3009–3025.
- Zeng, Y., Gantzer, C.J., Peyton, R.L., Anderson, S.H. (1996). Fractal Dimension and Lacunarity of Bulk Density Determined with X-ray Computed Tomography. Soil Science Society of America, Journal 60, p. 1718–1724.
- Zhang, R.Y., Liou, J.G., and Cong, B.L. (1994). Petrogenesis of Garnet-Bearing Ultramafic Rocks and Associated Eclogites in the Su-Lu Ultrahigh-Pressure Metamorphic Terrane, China. Journal of Metamorphic Geology, v. 12, p. 169–186.
- Zhu, W., Wong, T.F. (1999). Network Modeling of the Evolution of Permeability and Dilatancy in Compact Rock. Journal of Geophysical Research, v. 104, p. 2963-2971.
- Zhu, W., Wong, T.F. (1997). Shear-Enhanced Compaction in Sandstones Under Nominally Dry and Water Saturated Conditions, the Proceedings of 36th U. S. Rock Mechanics Symposium, New York, paper 364.
- Zhu, W., Laurent, M., Wong, T.F. (2007). A Probabilistic Damage Model of Stress-Induced Permeability Anisotropy During Cataclastic Flow. Journal of Geophysical Research, v. 112, n. B10207, 29 p.
- Zoback, Mark D., Byerlee, J.D. The Effect of Cyclic Differential Stress on Dilatancy in Westerly Granite Under Uniaxial and Triaxial Conditions. Journal of Geophysical Research, v. 80, n. 11, p. 1526-530.

APPENDIX A

MATLAB SCRIPTS AND FUNCTIONS

Automated Image Analysis Script and Point Count Plot

```
% TITLE: POINT_COUNT_MASTER_ANALYSIS.m
%
% AUTHOR: Justin Roth
%
% PURPOSE:
%   - To Automatically threshold scanned images of thin sections and
%     calculate the open, healed, and skeletal fractional porosity values
%
%
%
% clear all, close all

%%%%%%%%%%%%%%%
%% 1.0 INPUTS
%%%%%%%%%%%%%%%
%% File Path Information
dir_path_PC = 'DATA_PC/';
dir_path_IM = 'DATA_Image/';
coda = '.tif'; % specify file extension type

%% Summary of Image and Point Count Files for Analysis
%FILE: 3937FA
% TSname_image = '3937FAtsimage'; % load thin section image file
% PC_open = '3937FAPCopen.txt';
% PC_healed = '3937FAPChealed.txt';
% PC_host = '3937FAPChost.txt';
% DP = 484; % in [pixels per cm]
%
% %FILE: 3617PFWA
% TSname_image = '3617PFWAtsimage'; % load thin section image file
% PC_open = '3617PFWAPCopen.txt';
% PC_healed = '3617PFWAPChealed.txt';
% PC_host = '3617PFWAPChost.txt';
% DP = 484; % in [pixels per cm]
%
% %FILE: 3617PFWB
% TSname_image = '3617PFWBtsimage'; % load thin section image file
% PC_open = '3617PFWBPCopen.txt';
% PC_healed = '3617PFWBPChealed.txt';
% PC_host = '3617PFWBPChost.txt';
% DP = 484; % in [pixels per cm]
%
% %FILE: 4152F
% TSname_image = '4152Ftsimage'; % load thin section image file
% PC_open = '4152FPCopen.txt';
% PC_healed = '4152FPChealed.txt';
% PC_host = '4152FPChost.txt';
% DP = 484; % in [pixels per cm]
%
% %FILE: 4267FA %% Very good Example to Show
TSname_image = '4267FAtsimage'; % load thin section image file
PC_open = '4267FAPCopen.txt';
PC_healed = '4267FAPChealed.txt';
PC_host = '4267FAPChost.txt';
DP = 484; % in [pixels per cm]
```

```

%
% %FILE: 4302FA
% TSname_image = '4302FAtsimage'; % load thin section image file
% PC_open = '4302FAPCopen.txt';
% PC_healed = '4302FAPChealed.txt';
% PC_host = '4302FAPChost.txt';
% DP = 484; % in [pixels per cm]
%
% %FILE: 4303PFWA
% TSname_image = '4303PFWAtsimage'; % load thin section image file
% PC_open = '4303PFWAPCopen.txt';
% PC_healed = '4303PFWAPChealed.txt';
% PC_host = '4303PFWAPChost.txt';
% DP = 484; % in [pixels per cm]
%
% %FILE: 4303PFWB
% TSname_image = '4303PFWBtsimage'; % load thin section image file
% PC_open = '4303PFWBPCopen.txt';
% PC_healed = '4303PFWBPChealed.txt';
% PC_host = '4303PFWBPChost.txt';
% DP = 484; % in [pixels per cm]
%
% %FILE: 4305PHW
% TSname_image = '4305PHWtsimage'; % load thin section image file
% PC_open = '4305PHWPCopen.txt';
% PC_healed = '4305PHWPChealed.txt';
% PC_host = '4305PHWPChost.txt';
% DP = 484; % in [pixels per cm]
%
% %FILE: 4306FA
% TSname_image = '4306FAtsimage'; % load thin section image file
% PC_open = '4306FAPCopen.txt';
% PC_healed = '4306FAPChealed.txt';
% PC_host = '4306FAPChost.txt';
% DP = 484; % in [pixels per cm]
%
% %FILE: 4308PFW
% TSname_image = '4308PFWtsimage'; % load thin section image file
% PC_open = '4308PFWPCopen.txt';
% PC_healed = '4308PFWPChealed.txt';
% PC_host = '4308PFWPChost.txt';
% DP = 484; % in [pixels per cm]
%
% %FILE: 35235WFA *****DOES NOT WORK*****
% TSname_image = '35235WFAtsimage'; % load thin section image file
% PC_open = '35235WFAPCopen.txt';
% PC_healed = '35235WFAPChealed.txt';
% PC_host = '35235WFAPChost.txt';
% DP = 484; % in [pixels per cm]

%%%%%%%%%%%%%%%
%% 2.0 Load Data
%%%%%%%%%%%%%%%
% Load Image Data
C_o = imread([dir_path_IM,TSname_image,coda]);
    for i = 1:3 % flips the rows of the image while preserving columns
        C(:,:,i) = flipud(C(:,:,i));
    end
% Check on Image in Matrix Row Column coordinate system
    % figure
    % subplot(1,2,1)
    % image(C_o), axis equal, axis tight

```

```

% title(':Matrix Coordanates: Original Image')
% subplot(1,2,2)
% image(C), axis equal, axis tight
% title('Matrix Coordinates: Flipped Image')

% Load Point Count Data
PCo = load([dir_path_PC,PC_open]);
PCh = load([dir_path_PC,PC_healed]);
PCr = load([dir_path_PC,PC_host]);

%%%%%%%%%%%%%%%
%% 3.0 PLOT DATA
%%%%%%%%%%%%%%%

%% 3.1 Common Plotting
hf2 = figure(2);
% [h,ax] = GenerateTransectAxes(hf2);
nc = 4; % number of columns in plot
[h,ax] = GenerateTransectAxesColumns(hf2,nc);

MyAxisLim = [0 2 0 3.5];

%% 3.2 Plot Thin Section Image
axes(ax(1));
% SCALE IMAGE DATA to physical dimensions
[XC,YC] = ScaleImage(C,DP);
% TS image
image(XC(:,1),YC(:,1),C);
% Format plot
xlabel('X-position [cm]')
ylabel('Y-position [cm]')
title('Thin Section Image')
hold on
% set(gca,'YDir','normal') % set properties of the axes within figures
axis equal, box on, grid on
axis(MyAxisLim) % axis tight
title('Thin Section Image')

%% 3.3 Plot Point Count Data
% Convert the x-axes and Y-axes from cm to mm
% define variables
Xo = PCo(:,1); % open porosity
Yo = PCo(:,2);
Xh = PCh(:,1); % healed porosity
Yh = PCh(:,2);
Xr = PCr(:,1); % host rock
Yr = PCr(:,2);

% scale
Xos=Xo./10;
Yos=Yo./10;
Xhs=Xh./10;
Yhs=Yh./10;
Xrs=Xr./10;
Yrs=Yr./10;

% Plot
axes(ax(2));
plot(Xos,Yos,'.b') % open
hold on
plot(Xhs,Yhs,'.g')
plot(Xrs,Yrs,'.r')

```

```

xlabel('X-position [cm]'); % ylabel('Y-position [cm]')
title('Point Count Porosity Map')
colormap(bone); % also consider 'bone'
% set(gca,'YDir','normal') % set properties of the axes within figures
axis equal, box on, grid on
axis(MyAxisLim) % axis tight
hold on

%% 3.3 Thresholding the TS image file
axes(ax(3));
% 3.3.1 Set Limits for RGB color threshold on identifying Open and
% EXPLICIT SETTINGS - UNIVERSAL FOR ALL SAMPLES
% Healed
MinRheal = 133;
MaxRheal = 255; %% 255 will capture the region corresponding to no TS
MinGheal = 150; %175; % 200; % 0;
MaxGheal = 255;
MinBheal = 1; % 0; % a 1 avoids picking the red the region
% corresponding to no TS
MaxBheal = 255;

% Open
MinRopen = 0;
MaxRopen = 10; % 0;
MinGopen = 0;
MaxGopen = 150; % 210;
MinBopen = 0;
MaxBopen = 255;

% ALTERNATIVE: Devine a two files of six columns each where each
% COLUMNS correponds to a different "best" set of thresholding limits
% for a given sample.
% Then the ROWS correspond to:
% MinRopen,MaxRopen,MinGopen,MaxGopen,MinBopen,MaxBopen
% The column headers should be the correponding file names as per the
% point count data table at the beginning of the script so that the
% definition of "PC_open" can be used to determine which row to use in
% plotting
%
% nheaderlines = 1;
% delim = \t;
% hlim = importdata('PC_Healed_Phi_Limits.txt',delim,nheaderlines);
% Ic = find(PC_healed == hlim.colheaders{1,:});
% MinRheal = hlim(1,Ic);
% MaxRheal = hlim(2,Ic);
% MinGheal = hlim(3,Ic);
% MaxGheal = hlim(4,Ic);
% MinBheal = hlim(5,Ic);
% MaxBheal = hlim(6,Ic);
%
% olim = importdata('PC_Open_Phi_Limits.txt',delim,nheaderlines);
% Ic = find(PC_open == olim.colheaders{1,:});
% MinRopen = olim(1,Ic);
% MaxRopen = olim(2,Ic);
% MinGopen = olim(3,Ic);
% MaxGopen = olim(4,Ic);
% MinBopen = olim(5,Ic);
% MaxBopen = olim(6,Ic);

% 3.3.2 Located Pixels within the Threshold Limits
% Break 3D array into series of 3, 2D arrays, each 2D array corresponds
% to a page of the 3D array; 1 page for each R, G, B

```

```

C1=C(:,:,:,1); C2 = C(:,:,:2); C3 = C(:,:,:3);
% Search Array for Pixels outside the thin section
I0 = find(C1 == 255);

% Search arrays for threshold for healed porosity
I1 = find(C1>=MinRheal & C1<=MaxRheal & ...
           C2>=MinGheal & C2<=MaxGheal & ...
           C3>=MinBheal & C3<=MaxBheal);
% Replace identified values with zero
C1(I1) = 0; C2(I1) = 255; C3(I1) = 0;

% Threshold for open porosity
% lower lim. & upper lim.
I2 = find(C1>=MinRopen & C1<=MaxRopen & ...
           C2>=MinGopen & C2<=MaxGopen & ...
           C3>=MinBopen & C3<=MaxBopen); % Search arrays for threshold
% Replace identified values with zero
C1(I2) = 0; C2(I2) = 0; C3(I2) = 255;
% Rebuild 3D array
D = C1; D(:,:,:2) = C2; D(:,:,:3) = C3;
% Plot image with threshholde color matrix D
image(XC(1,:),YC(:,1),D);

% Plot formatting
xlabel('X-position [cm]')
hold on
set(gca,'YDir','normal') % sets properties of the axes within figure
axis equal, box on, grid on
axis(MyAxisLim) % axis tight
title('Thresholded Thin Section Image')

%% 4.0 CALCULATE STATISTICS FOR COMPARISON OF APPRACHES
axes(ax(4))
% point count
phi.pc.t = length(Xos)+length(Xhs)+length(Xr);
phi.pc.o = length(Xos)/phi.pc.t;
phi.pc.h = length(Xhs)/phi.pc.t;
phi.pc.s = phi.pc.o + phi.pc.h;
% automated image analysis
phi.im.t = numel(C1)-length(I0);
phi.im.o = length(I2)/phi.im.t;
phi.im.h = length(I1)/phi.im.t;
phi.im.s = phi.im.o + phi.im.h;
% Formatting
MyMarker = {'MarkerSize',10,'LineWidth',1.25,'MarkerFaceColor','y'};
h4 = plot([0 1],[0 1],'k--');
hold on
h1 = plot(phi.pc.o, phi.im.o,'ob',MyMarker{:});
h2 = plot(phi.pc.h, phi.im.h,'og',MyMarker{:});
h3 = plot(phi.pc.s, phi.im.s,'or',MyMarker{:});
xlabel('Fractional Point Count Porosity []')
ylabel('Fractional Automated Image Analysis []')
title('Method Comparison')
axis square, box on, grid on
legend([h1 h2 h3 h4], 'Healed', 'Open', 'Skeletal', '1:1 Corr. Line')
phimax = 1.0;
axis([0 phimax 0 phimax])

%% PLOT THIN SECTION IMAGE FOR REFERENCE
% C2 = TSi;
% Thresh = find(C2(:,:,:,1)>=133&C2(:,:,:,2)>=116&C(:,:,:,3)>=101);
% C(Thresh)=0;

```

```
% [Xc,Yc] = ScaleImage(C,DP);
% image(Xc(1,:),Yc(:,1),C); % TS image
% xlabel('X-position [cm]')
% ylabel('Y-position [cm]')
% title('Thin Section Image')
% hold on
% set(gca,'YDir','normal') % sets properties of the axes within figures
% axis equal, box on, axis tight, grid on
% title('Thin Section Image')
```

High Resolution Image Analysis Script

```
%% MASTER_Image_Analysis.m
%
% AUTHORS: Nick Davatzes and Justin Roth
% DATE: 2013
%
% PURPOSE: Analysis Image Data for Porosity Variation along a transect
%
% NOTES: can convert to function by:
%   1. removing the clear all, close all
%   2. comment out lines beginning with FName_open, FName_healed, DP
%   3. comment out x, dy and ds parameters
%   4. at end of the script capture all important parameters into a
%      structure array "OUT"
%
% Insert at top:
% function [OUT] = PorosityImage(FName_open,FName_healed,DP,x,dy,ds)
%

clear all, close all
%%%%%%%%%%%%%
%% STEP 1: SET BASIC PARAMETERS %%%
%%%%%%%%%%%%%
% STEP 1.1: Define Optional Tests
% - Test Transect Sensitivity
T_sensitivity = 'N'; % 'Y' = yes, 'N' = no
% - Write WORKSPACE Variables to File
Data_export = 'N';
% - Export Transect Figure as .fig file
Figure_export = 'N';

% SCANLINE PARAMETERS
% Optimal window size should consider the following factors
% 1. Property attempting to characterize
% 2. Grain size: probably not meaningful unless window size >= 10 X grain
%    size
% 3. Pore size and distribution
% 4. Pixel resolution: want at least 10 X pixel resolution sized box
% 5. Run Time --> practical
% 6. If comparing to other data: e.g., x-ray CT, it should be conducted
%    a consistence sample volume across both analyses
ds = 0.01; % 0.05; % window edge length in cm; ds = 0.1 default
dy = 0.002; % window step increment in cm, 0,.01 preferred?
% Should be less than ds --> dy = ds/5;

% FORMATTING
%MyColor = flag(3);
MyColor = [0 .4 0 ;.1 .5 1;1 0 0];
MyColor2 = [0 0 0 ;.1 .5 1;1 1 .9];
MyColor3 = [0 .4 0 ;1 0 1;1 1 1];

cmap1 = MyColor3;
cmap2 = MyColor2;
cmap = [cmap1;cmap1;cmap2];

A = 0.6; % scale for colormap for healed porosity

%%%%%%%%%%%%%
%% STEP 2: IMPORT FILES %%%
%%%%%%%%%%%%%
% NOTE:
```

```

% Script Requires access to two sets of files:
% (1) Detailed, high resolution section mosaics
%     (a) open porosity binary image
%     (b) healed porosity binary image
% (2) lower resolution, whole thin section images

%% STEP 2.1: DEFINE Data set to use in analysis: FILE NAMES:
% Choose the file to analyze by uncomming the corresponding filename
% definitions and commenting the previously used set.
%
% NOTR: Do not include extensions in file names, all must be ".tif"

% FILE: 3937FAbin
 FName_image = '3937FAbinimage'; % load thin section image file
 FName_open = '3937FAbinopen'; % load binary image file
 FName_healed = '3937FAbinhealed'; % load binary image file
 DP = 9605; % in [pixels per cm]
%%
% % FILE: 4302FAbin
% FName_image = '4302FAbinimage'; % load thin section image file
% FName_open = '4302FAbinopen'; % load binary image file
% FName_healed = '4302FAbinhealed'; % load binary image file
% DP = 9605; % in [pixels per cm]
%%
% FILE: 35235WFAbin
% FName_image = '35235WFAbinimage'; % load thin section image file
% FName_open = '35235WFAbinopen'; % load binary image file
% FName_healed = '35235WFAbinhealed'; % load binary image file
% DP = 9605; % in [pixels per cm]
%%
%FILE: 3617PFWAbin
% FName_image = '3617PFWAbinimage'; % load thin section image file
% FName_open = '3617PFWAbinopen'; % load binary image file
% FName_healed = '3617PFWAbinhealed'; % load binary image file
% DP = 9605; % in [pixels per cm]
%%
% FILE: 3617PFWBbin
% FName_image = '3617PFWBbinimage'; % load thin section image file
% FName_open = '3617PFWBbinopen'; % load binary image file
% FName_healed = '3617PFWBbinhealed'; % load binary image file
% DP = 9605; % in [pixels per cm]
%%
% % FILE: 4152Fbin
% FName_image = '4152Fbinimage'; % load thin section image file
% FName_open = '4152Fbinopen'; % load binary image file
% FName_healed = '4152Fbinhealed'; % load binary image file
% DP = 9605; % in [pixels per cm]
%%
% % FILE: 4267FAbin
% FName_image = '4267FAbinimage'; % load thin section image file
% FName_open = '4267FAbinopen'; % load binary image file
% FName_healed = '4267FAbinhealed'; % load binary image file
% DP = 9605; % in [pixels per cm]
%%
% --> Good Example with different porosities to image
%FILE: 4303PFWAbin (TRANSECT DOES NOT RUN ALONG UP-WELL; RATHER PERP.)
% FName_image = '4303PFWAbinimage'; % load thin section image file
% FName_open = '4303PFWAbinopen'; % load binary image file
% FName_healed = '4303PFWAbinhealed'; % load binary image file
% DP = 9605; % in [pixels per cm]
%%
% % FILE: 4303PFWBbin(**OPEN & IMAGE FILES NEED TO BE FIXED)
% FName_image = '4303PFWBbinimage'; % load thin section image file

```

```

% FName_open = '4303PFWBbinopen'; % load binary image file
% FName_healed = '4303PFWBbinhealed'; % load binary image file
% DP = 9605; % in [pixels per cm]
%%
% % FILE: 4305PHWbin
% FName_image = '4305PHWbinimage'; % load thin section image file
% FName_open = '4305PHWbinopen'; % load binary image file
% FName_healed = '4305PHWbinhealed'; % load binary image file
% DP = 9605; % in [pixels per cm]
%%
%FILE: 4306FAbin
% FName_image = '4306FAbinimage'; % load thin section image file
% FName_open = '4306FAbinopen'; % load binary image file
% FName_healed = '4306FAbinhealed'; % load binary image file
% DP = 9605; % in [pixels per cm]
%%
% FILE: 4308PFWbin (**OPEN AND IMAGE FILES NEED TO BE FIXED)
% FName_image = '4308PFWbinimage'; % load thin section image file
% FName_open = '4308PFWbinopen'; % load binary image file
% FName_healed = '4308PFWbinhealed'; % load binary image file
% DP = 9605; % in [pixels per cm]

%% STEP 2.2: IMPPORT DATA
folder = [pwd,'/DATA_Image/'];
coda = '.tif'; % specify file extension type
Mi = imread([folder,FName_image,coda]);
Mo = imread([folder,FName_open,coda]);
Mh = imread([folder,FName_healed,coda]);

%% STEP 2.3: SCALE DATA (
% TRANSFORM FROM TIFF/BITMAP COORDINATES TO SPATIAL COORDINATES)
% STEP 2.3.1: Scale by Dots Per Centimeter
[Xi,Yi] = ScaleImage(Mi,DP);
[Xo,Yo] = ScaleImage(Mo,DP);
[Xh,Yh] = ScaleImage(Mh,DP);

% OPTIONAL VISUALIZATION FOR QA: Visualize image for reference in matrix
% coordinates as per default for TIFF images
% hf1 = figure(1);
% imagesc creates a colored image plot in which the x,y coordinates
% correspond to the column, row positions. Thus there are no zero
% positions and the upper left-hand corner has coordinates 1,1 and the
% lower left-hand corner has coordinates m,n
% subplot(1,3,1); % Visualize Thin Section Image
% imagesc(Mi);
% axis equal, colormap(jet), colorbar, axis tight, title('Open')
% subplot(1,3,2); % Open Porosity Map
% imagesc(Mo), axis equal, colormap(MyColor(2:3,:)),...
% colorbar, axis tight, title('Open')
% subplot(1,3,3); % Healed Porosity Map
% imagesc(A*Mh), axis equal, colormap(MyColor([1,3],:)),...
% colorbar, axis tight, title('Healed')

%% STEP 2.3.2: VISUALIZE THIN SECTION IMAGE in spatial coordinates
hf2 = figure(2);
[h,ax] = GenerateTransectAxes(hf2); % Format without subplot

axes(ax(1));
% subplot(1,5,1)
image(Xi(:,1),Yi(:,1),Mi); % open

```

```

xlabel('X-position [cm]')
ylabel('Y-position [cm]')
title('Healed Porosity')
colormap(cmap); freezeColors; % also consider 'bone'
hold on
set(gca,'YDir','normal') % affects axes properties within figures
axis equal, box on, axis tight, grid on
title('Image Stitch 4306 FA')

% Visualize Porosity Maps
axes(ax(2));
h = image(Xo(1,:),Yo(:,1),Mo); % open
hold on
xlabel('X-position [cm]'); % ylabel('Y-position [cm]')
title('Open Porosity Map')
colormap(cmap); freezeColors; % also consider 'bone'
set(gca,'YDir','normal') % affects axes properties within figures
axis equal, box on, axis tight, grid on
hold on

axes(ax(3));
% image(Xo(1,:),Yo(:,1),Mo); % open
% colormap(cmap)
% hold on
image(Xh(1,:),Yh(:,1),A*Mh); % healed
colormap(cmap); freezeColors
title('Healed Porosity Map')
set(gca,'YDir','normal') % affects axes properties within figures
axis equal, box on, axis tight, grid on
xlabel('X-position [cm]'), % ylabel('Y-position [cm]')

% NOTE: Consider Visualize matrix, open porosity, and healed porosity
% in single tri-colored image. Since both data sets are binary 0 or 1,
% then adding a fraction of one image to the other produces a tri-
% indexed image of 0,A,1, where A is the fraction. This will only work
% if the two image matrices are exactly the same size.
% image(Xo(1,:),Yo(:,1),Mo + A*Mh); % open
% --> currently inoperative... pixel dimensions do not match

axes(ax(5));
% image(Xo(1,:),Yo(:,1),Mo); % open
% colormap(cmap)
% hold on
% Only works if two images are the exact same sized matrices;
% assumes binary images
% Ihm = find(Mh == max(max(Mh)));
% Iom = find(Mo == max(max(Mo)));
Ihp = find(Mh == min(min(Mh)));
Iop = find(Mo == min(min(Mo)));
% Mhp = Mh; Mhp(Ihm) = 0; Mhp(Ihp) = 165;
% Mop = Mo; Mop(Iom) = 0; Mop(Iop) = 0;
% Mho = Mop + Mhp;
% Mho(Ihm) = 0; Mho(Iom) = 0;
Mho = ones(size(Mh));
Mho(Ihp) = 2;
Mho(Iop) = 3;

image(Xh(1,:),Yh(:,1),Mho); % healed
colormap([1 1 1;0.1 0.5 0.1;0 0 0.9])
title('Open + Healed Porosity Map')
set(gca,'YDir','normal') % affects axes properties within figures
axis equal, box on, axis tight, grid on
xlabel('X-position [cm]'), % ylabel('Y-position [cm]')

```

```

box on

%% PLACEHOLDER - ROTATE IMAGE FOR TRANSECT SAMPLING
% --> used to build transects that are normal to the fracture trace
%   across the thin section
%   Two approaches: (1) rotate image using rotation matrix
%                   (2) rotate transect data after calcualted

% (OPTIONAL CHECK: Visualize rotated image)

%%%%%%%%%%%%%
% STEP 3: CALCULATE TRANSECT DATA %
%%%%%%%%%%%%%

% STEP 3.1: Clculation
% Trasect parallel to y-axis, x=A, where A is taken to be the center
% --> could be repeated for multiple x positions in order to
% characterize robustness of distribution laterally as a function of
% position along the fracture...
x = max(max(Xo))/2; %find center x, scalar
% For faster testing use large dy, set dy to 0.1

% NOTE: dy and ds defined at top of file structure for ease of
% manipulations
% (STEP 1)
% dy = 0.01; % window step increment in cm, 0,.01 preferred?
% ds = 0.1; % window edge length in cm

% Open Porosity ("P" for porosity, "o" for open)
Po = ScanlineY(Xo,Yo,Mo,x,dy,ds); % gathers structured array generated
% by transect analysis carred out in
% ScanlineY
% Healed Porosity ("P" for porosity, "h" for healed)
Ph = ScanlineY(Xh,Yh,Mh,x,Po.y,ds);

% STEP 3.2: Visualization of Transect line
axes(ax(3));
plot(Po.x,Po.y,'.k','MarkerSize',5) % dots representing window centers
% from transect
% Visualize example sample box
dummyx = Po.x(7); dummyy = Po.y(7);
dummyX = [dummyx-0.5*ds;dummyx-0.5*ds;dummyx+0.5*ds;...
          dummyx+0.5*ds;dummyx-0.5*ds];
dummyY = [dummyy-0.5*ds;dummyy+0.5*ds;dummyy+0.5*ds;...
          dummyy-0.5*ds;dummyy-0.5*ds];
plot(dummyX,dummyY,'k-')
% Insert text recording number of pixels within a typical window
text(Po.x(1)+0.5*ds,Po.y(1),['n=',num2str(Po.bn(5))])

axes(ax(4));
% Porosity along scanline
% open
hp1 = plot(Po.p0,Po.y,'-', 'LineWidth',1.5, 'Color', MyColor(2,:));
hold on
% healed
hp2 = plot(Ph.p0,Ph.y,'-', 'LineWidth',1.5, 'Color', MyColor(1,:));
hp3 = plot(Po.p0 + Ph.p0,Ph.y,'r-','LineWidth',1); % skeletal

% Reference total porosity of whole slide:
% - arbitrarily depends on the slide dimensions relative to the
%   fracture size
% - if image does not spane whole domain, the value can be contaminated

```

```

% by meaningless pixels along the edges of the slide
Po.p_all = 1-sum(sum(Mo/max(max(Mo)) ))/numel(Mo);
Ph.p_all = 1-sum(sum(Mh/max(max(Mh)) ))/numel(Mh);
Pskeletal= Po.p_all + Ph.p_all;
hp4 = plot(ones(size(Po.y))*Po.p_all,Po.y,'--','Color',MyColor(2,:));
hp5 = plot(ones(size(Po.y))*Ph.p_all,Po.y,'--','Color',MyColor(1,:));
hp6 = plot(ones(size(Po.y))*Pskeletal,Po.y,'r--');
box on, grid on
 xlabel({'Fractional';'Porosity []'})
 title({'Porosity Variation';['w = ',num2str(ds), ' [cm]']})
 xlim([0 1])
 ylim([0 max([max(max(Yo));max(max(Yh))])])

legend([hp1 hp2 hp3 hp4 hp5 hp6], 'Open', 'Healed', 'Skeletal',...
    'Avg Open', 'Avg Healed', 'Avg Skeletal')

% Set Common formatting
axes(ax(1)), ylim([0 max([max(max(Yo));max(max(Yh))])])
axes(ax(2)), ylim([0 max([max(max(Yo));max(max(Yh))])])
axes(ax(3)), ylim([0 max([max(max(Yo));max(max(Yh))])])
axes(ax(5)), ylim([0 max([max(max(Yo));max(max(Yh))])])
% Ensure plot aspect ratios match
sc = get(ax(2),'PlotBoxAspectRatio');
set(ax(4),'PlotBoxAspectRatio',sc);
set(ax(5),'PlotBoxAspectRatio',sc);
% linkaxes(ax);

%%%%%%%%%%%%%%%
%% STEP 4: Place hoolder: Fracture/Pore Edges in BINARY IMAGES %%%
%%%%%%%%%%%%%%%
% OR Look for edge-finding algorithms to impliment.

% CALCULATE GRADIENT TO FIND EDGES and ROUGHNESS
% [Fx,Fy] = gradient(M,(X(1,:),Y(:,1)));

%%%%%%%%%%%%%%%
%% STEP 5: PORE SIZE and SHAPE ANALYSIS STATISTICS %%%
%%%%%%%%%%%%%%%
% - For whole slide
% - within sliding window; do pore characteristics change as a funciton
%   of distance from the fracture surface?
% PORE SHAPE ANALYSIS STATISTICS

%%%%%%%%%%%%%%%
%% STEP 6: Sensitivity analysys to Window Dimensions %%%
%%%%%%%%%%%%%%%

if T sensitivity == 'Y';
% STEP 6.1: TRANSECT SENSITIVITY TO BOX SIZE and STATISTICS (SQUARE):
% --> How does the apparent porosity distribution depend on the window
% size used to measure porosity along the transect?
% -----
% Open/Healed Porosity/Skeletal

% Define window size range
ds_test = [0.001,0.01:0.01:0.1,0.2,0.3]; % in cm
% ds_test = [0.01,0.05,0.1];

% Define variables to store data
Ptest.p = zeros(length(Po.y),length(ds_test));

```

```

Ptest.y = Ptest.p;
Ptest.n = zeros(size(ds_test));
Ptest.q = cell(size(ds_test));
Ptest.labels = Ptest.q;
% Ptest.q = Ptest.n;

%% STEP 6.2: GATHER STATISTICS
% Investigates how variable porosity is within a transect for a given
% window size.
for i = 1:length(ds_test)
    % use same x and Po.y as above for consistency
    Pdummy = ScanlineY(Xh,Yh,Mh,x,Po.y,ds_test(i));
    Ptest.p0(:,i) = Pdummy.p0; % matrix with a column of porosity
                                % measurements for each window edge length
                                % tested
    Ptest.y(:,i) = Po.y; % matrix with a column of y positions for each
                        % window edge length tested
    Ptest.n(i) = numel(Pdummy.in{1}); % number of pixels used to calculate
                                    % porosity
    Ptest.q{i} = quartile(Pdummy.p0); % variability of porosity among
                                    % measurements for each window size
    Ptest.labels{i} = num2str(ds_test(i));
    % Ptest.q(i) = quartile(Pdummy.p0); % variability of porosity among
                                    % measurements for each window size
end
clear Pdummy

%% STEP 6.3: VISUALIZE RESULTS
hf3 = figure(3);
subplot(1,2,1)
% Simple plot option:
% htest = plot(Ptest.p0,Ptest.y,'-'); % plot whole matrix as line
%                                         % series, default line color
MyLineColor = cool(length(ds_test)); % colormap(jet(length(ds_test)))
htest = zeros(size(ds_test))*nan;
for i = 1:length(ds_test)
    hdummy = plot(Ptest.p0(:,i),Ptest.y(:,i),'-', 'Color',...
                  MyLineColor(i,:));
    htest(i) = hdummy;
    hold on
end
% Formatting
box on, grid on
xlabel({'Fractional';'Porosity []'})
xlim([0 1])
title({'Sensitivity to Window Size';['Pixel Size = ',num2str(1/DP),...
    ' cm']})
% Legend
hl = legend(htest,{num2str(ds_test)});
v = get(hl,'title');
set(v,'string','Window Size [cm]');

% Plot stats for each window
subplot(2,2,2)
plot(log10(ds_test),Ptest.n,'r-')
boxplot(Ptest.p0,'notch','on','colors',MyLineColor, ...
    'labels',Ptest.labels,'labelorientation','inline',...
    'outliersize',4,'symbol','k+')
ylabel('Fractional Porosity of Windows []')
xlabel('Window Size [cm]')
ylim([0 1]);
box on, grid on

```

```

    subplot(2,2,4)
    % Plot mean and standard deviation of porosity values from the
    % collection of window calculations for each window edge length
    e_value = std(Ptest.p0);
    errorbar(log10(ds_test),mean(Ptest.p0),e_value,e_value,'ko');
    hold on
    scatter(log10(ds_test),mean(Ptest.p0),10^2,MyLineColor,'filled',...
        'MarkerEdgeColor','k')
    % Plot 25th, 50th, and 75th quartiles as lines Ptest.q

    xlabel('log_{10}(Window Size) [cm]')
    ylabel('Fractional Porosity []')
    title('Porosity variation as f(Window Size)')
    ylim([0 1])
    box on, grid on

    % --> Implement modified boxplot developed for well log data for Batir

    % Statistics

end % end of optional test

%%%%% STEP 7: Comparison of porosity characteristics among samples %%%
%% (for another, separate script)
%
% Some Characteristics to consider
%
% A. METHOD COMPARISON
% 1. correspondence among methods
% 2. does the compatibility of measurements of porosity by different
%    methods depend on the stage of the fracture?
% --> ntoes
%
% B. GEOLOGIC COMPARISONS
% 1. Dependence of porosity change on fracture stage
% 2. Width of "changed" porosity in rock adjacent to the fracture
%    - amount of open and healed
%    - pore size distribution
%    - pore shape
%    - pore alignment
%    (anisotropy; all elongated and elongated in same direction)

%%%%% STEP 8: OUTPUTS: SAVE VARIABLES AND FIGURES %%%
%% STEP 8.1:
if Data_export == 'Y' || Figure_export == 'Y' || T_sensitivity == 'Y'
    % Define file name and name of structured array for export to .mat file
    fname = strrep(FName_image,'binimage','');
    %string replace function
    if Data_export == 'Y';
        % Collect variables to save based on Filename
        eval(['GEON2_',fname,'.Ph = Ph;']);
        eval(['GEON2_',fname,'.Po = Po;']);
        eval(['GEON2_',fname,'.Pskeletal = Pskeletal;']);
        eval(['GEON2_',fname,'.ds = ds;']);
        eval(['GEON2_',fname,'.dy = dy;']);

```

```

if T_sensitivity == 'Y'
    eval(['GEON2_', fname, '.Ptest = Ptest;']);
end
% Save Variables from workspace to current working direction
working_dir = cd;
% NOTE: can force data to be stored to a desired sub-directory via:
save(['GEON2_', fname, '.mat'], ['GEON2_', fname], '-mat');
    % modifiable to save specific variables from workspace
    % via save(FILENAME,VARIABLES)
    % directory_for_storage = [working_dir, '\data_archive\'];
    % save([directory_for_storage,'GEON2_', fname], data);

% To recover data in another script simply use:
% load(FileNameString)
% e.g.,
% load('GEON2_3617PFWA.mat'); % loads saved variables in .mat file
%                               % to the workspace
end
folder = [pwd, '/FIGURES/'];
if Figure_export == 'Y'
    % Save Figure as .fig file - manipulatable in Matlab
    saveas(hf2,[folder,'FIG - GEON2_', fname, '_Trans.fig']);
    saveas(hf2,[folder,'FIG - GEON2_', fname, '_Trans.pdf']);
end
if T_sensitivity == 'Y';
    saveas(hf3,[folder,'FIG - GEON2_', fname, 'W_sensitivity.fig']);
    saveas(hf3,[folder,'FIG - GEON2_', fname, 'W_sensitivity.pdf']);
end
end

```

High Resolution Image Analysis Function (1 of 2):

```
function [out,hf,ht] = PorosityImageAnalysis(fname,DP,dy,ds,...  
    Plotting,T_sensitivity,Data_export,Figure_export)  
  
%% MASTER_Image_Analysis.m  
%  
% AUTHOR: Justin Roth/Nick Davatzes  
% DATE: last modified 05/05/13  
%  
% PURPOSE: Analysis Image Data for Porosity Variation along a transect  
%  
% INPUTS  
% fname = base name without coda that references three files in .tif  
%         format (with matching pixel dimensions):  
%             (1) _image  
%             (2) _open  
%             (3) _healed  
% --> the resulting naming convention is derived from:  
%     [fname,'_type','.tif'];  
% --> which gives  
%     (1)'FNAME_image.tif'  
%     (2)'FNAME_open.tif'  
%     (3)'FNAME_healed.tif'  
%  
% DP      = dots per cm  
% ds      = window size in cm  
% dy      = step distance along transect  
% Plotting = 'Y' or 'N': create plots of the analysis  
% T_sensitivity = 'Y' or 'N': run sensitivity to window size analysis  
% Data_export = 'Y' or 'N': write a structure array of the variables to  
%                     a d.mat file in which the file is named as:  
%                     ['GEON2_',FNAME,'.mat']  
%                     and in which the variables have the name:  
%                     'GEON2_',FNAME  
% Figure_export = 'Y' or 'N': print PDF files of figures if generated  
%  
%  
% OUTPUTS:  
% Figures  
% Sensitivity  
% Variables:  
%  
%  
% insert at top:  
% function [OUT] = PorosityImage(FName_open,FName_healed,DP,x,dy,ds)  
%  
% EXAMPLE SYNTAX:  
%  
% % Example File MetaData  
% % FILE: 3937FAbin (exclude extensions from file names-all must be ".tif")  
% % FName_image = '3937FAbinimage'; % load thin section image file  
% % FName_open = '3937FAbinopen'; % load binary image file  
% % FName_healed = '3937FAbinhealed'; % load binary image file  
% % DP = 9605; % in [pixels per cm]  
%  
% DP      = 9605; % bitmap resolution in points/cm; assumed same in  
%                 % x,y,z for a 26.7 micron pixel  
% dy      = 0.0125; % [cm] step size  
% ds      = 0.025; % [cm] 0.05; % window edge length  
% dx      = 0.2;   % [cm] % 0.05; % 0.3; % space between transects  
%                 % in the x and y direction
```

```

% diameter = 1.5;      % [cm] sample diameter in cm
% Plotting = 'Y';
% T_sensitivity = 'Y';
% Data_export = 'N';
% Figure_export = 'N';
%
% [out,hf,ht] = Poro] = PorosityImageAnalysis(fname,DP,dy,ds,Plotting, ...
%           T_sensitivity,Data_export,Figure_export)
%

%%%%%%%%%%%%%%%
%% STEP 1: SET BASIC PARAMETERS
%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%
if nargin < 2
    error('Image Resolution in dots per cm (DP) unspecified')
end

% STEP 1.1: Define Optional Tests
% Scanline parameters
% Optimal window size should consider the following factors
% 1. Property attempting to characterize
% 2. Grain size: probably not meaningful unless window size >= 10 X grain
%    size
% 3. Pore size and distribution
% 4. Pixel resolution: want at least 10 X pixel resolution sized box
% 5. Run Time --> practical
% 6. If comparing to other data: e.g., x-ray CT, it should be conducted
%    a consistence sample volume across both analyses
if nargin <3
    dy = 0.002 % window step increment in cm, .01 preferred?
                % Should be less than ds --> dy = ds/5;
end
if nargin <4
    ds = 0.01 % 0.05; % window edge length in cm; ds = 0.1 default
end
% - Plot Results?
if exist('Plotting') == 0
    Plotting = 'N'
end
% - Test Transect Sensitivity
if exist('T_sensitivity') == 0
    T_sensitivity = 'N' % 'Y' = yes, 'N' = no
end
% - Write WORKSPACE Variables to File
if exist('Data_export') == 0
    Data_export = 'N'
end
% - Export Transect Figure as .fig file
if exist('Figure_export') == 0
    Figure_export = 'N'
end
if Plotting == 'N' && Figure_export == 'Y'
    error(['Figures are not generated and cannot be exported: ...
            'Set PLOTTING to ''Y'' or FIGURE_EXPORT to ''N'''']);
end

% FORMATTING
%MyColor = flag(3);
MyColor = [0 .4 0 ;.1 .5 1;1 0 0];
MyColor2 = [0 0 0 ;.1 .5 1;1 1 .9];
MyColor3 = [0 .4 0 ;1 0 1;1 1 1];

```

```

cmap1 = MyColor3;
cmap2 = MyColor2;
cmap = [cmap1;cmap1;cmap2];

A = 0.6; % scale for colormap for healed porosity

%%%%% STEP 2: IMPORT FILES %%%
% NOTE:
% Two sets of files:
% (1) Detailed, high resolution section mosaics
% (2) lower resolution, whole thin section images
% Leave extensions off of file names-all must be ".tif"

%% STEP 2.1: FILE NAMES:
%
% %FILE: 3937FAbin
% FName_image = '3937FAbinimage'; % load thin section image file
% FName_open = '3937FAbinopen'; % load binary image file
% FName_healed = '3937FAbinhealed'; % load binary image file
% DP = 9605; % in [pixels per cm]
%%
% %FILE: 4302FAbin
% FName_image = '4302FAbinimage'; % load thin section image file
% FName_open = '4302FAbinopen'; % load binary image file
% FName_healed = '4302FAbinhealed'; % load binary image file
% DP = 9605; % in [pixels per cm]
%%
% FILE: 35235WFAbin
% FName_image = '35235WFAbinimage'; % load thin section image file
% FName_open = '35235WFAbinopen'; % load binary image file
% FName_healed = '35235WFAbinhealed'; % load binary image file
% DP = 9605; % in [pixels per cm]
%%
% %FILE: 3617PFWAbin
% FName_image = '3617PFWAbinimage'; % load thin section image file
% FName_open = '3617PFWAbinopen'; % load binary image file
% FName_healed = '3617PFWAbinhealed'; % load binary image file
% DP = 9605; % in [pixels per cm]
%%
% %FILE: 3617PFWBbin
% FName_image = '3617PFWBbinimage'; % load thin section image file
% FName_open = '3617PFWBbinopen'; % load binary image file
% FName_healed = '3617PFWBbinhealed'; % load binary image file
% DP = 9605; % in [pixels per cm]
%%
% %FILE: 4152Fbin
% FName_image = '4152Fbinimage'; % load thin section image file
% FName_open = '4152Fbinopen'; % load binary image file
% FName_healed = '4152Fbinhealed'; % load binary image file
% DP = 9605; % in [pixels per cm]
%%
% %FILE: 4267FAbin
% FName_image = '4267FAbinimage'; % load thin section image file
% FName_open = '4267FAbinopen'; % load binary image file
% FName_healed = '4267FAbinhealed'; % load binary image file
% DP = 9605; % in [pixels per cm]
%%
% %--> Good Example with different porosities to image
% %FILE: 4303PFWAbin (TRANSECT DOES NOT RUN ALONG UP-WELL; RATHER PERP.)

```

```

% FName_image = '4303PFWAbinimage'; % load thin section image file
% FName_open = '4303PFWAbinopen'; % load binary image file
% FName_healed = '4303PFWAbinhealed'; % load binary image file
% DP = 9605; % in [pixels per cm]
%%
% %FILE: 4303PFWBbin(**OPEN AND IMAGE FILES NEED TO BE FIXED)
% FName_image = '4303PFWBbinimage'; % load thin section image file
% FName_open = '4303PFWBbinopen'; % load binary image file
% FName_healed = '4303PFWBbinhealed'; % load binary image file
% DP = 9605; % in [pixels per cm]
%%
% %FILE: 4305PHWbin
% FName_image = '4305PHWbinimage'; % load thin section image file
% FName_open = '4305PHWbinopen'; % load binary image file
% FName_healed = '4305PHWbinhealed'; % load binary image file
% DP = 9605; % in [pixels per cm]
%%
% %FILE: 4306FAbin
% FName_image = '4306FAbinimage'; % load thin section image file
% FName_open = '4306FAbinopen'; % load binary image file
% FName_healed = '4306FAbinhealed'; % load binary image file
% DP = 9605; % in [pixels per cm]
%%
% %FILE: 4308PFWbin (**OPEN AND IMAGE FILES NEED TO BE FIXED)
% FName_image = '4308PFWbinimage'; % load thin section image file
% FName_open = '4308PFWbinopen'; % load binary image file
% FName_healed = '4308PFWbinhealed'; % load binary image file
% DP = 9605; % in [pixels per cm]

%% STEP 2.2: IMPORT DATA
% Note: All three images are currently loaded at the same time; this
% corresponds to the most memory intensive approach, sequential loading
% and analysis would be faster if memory is limited/the files are very
% large
% folder = [pwd,'/DATA_Image/'];
coda = '.tif'; % specify file extension type

% Mi = imread([folder,fname,'_image',coda]);
% Mo = imread([folder,fname,'_open',coda]);
% Mh = imread([folder,fname,'_healed',coda]);
Mi = imread([fname,'image',coda]);
Mo = imread([fname,'open',coda]);
Mh = imread([fname,'healed',coda]);

%% STEP 2.3: SCALE DATA (TRANSFORM FROM TIFF/BITMAP COORDINATES TO SPATIAL
% COORDINATES)

% STEP 2.3.1: Scale by Dots Per Centimeter
[Xi,Yi] = ScaleImage(Mi,DP);
[Xo,Yo] = ScaleImage(Mo,DP);
[Xh,Yh] = ScaleImage(Mh,DP);

% OPTIONAL VISUALIZATION FOR QA: Visualize image for reference in matrix
% coordinates as per default for TIFF images
% hf1 = figure(1);
% imagesc creates a colored image plot in which the x,y coordinates
% correspond to the column, row positions. Thus there are no zero
% positions and the upper left-hand corner has coordinates 1,1 and the
% lower left-hand corner has coordinates m,n
% subplot(1,3,1); % Visualize Thin Section Image
% imagesc(Mi); axis equal, colormap(jet),

```

```

    % colorbar, axis tight, title('Open')
    % subplot(1,3,2); % Open Porosity Map
    % imagesc(Mo), axis equal, colormap(MyColor(2:3,:)),
    % colorbar, axis tight, title('Open')
    % subplot(1,3,3); % Healed Porosity Map
    % imagesc(A*Mh), axis equal, colormap(MyColor([1,3],:)),
    % colorbar, axis tight, title('Healed')

%% STEP 2.3.2: VISUALIZE THIN SECTION IMAGE in spatial coordinates
if Plotting == 'Y';
hf2 = figure;
[h,ax] = GenerateTransectAxes(hf2); % Format without subplot

axes(ax(1));
% subplot(1,5,1)
image(Xi(:,1),Yi(:,1),Mi); % open
xlabel('X-position [cm]')
ylabel('Y-position [cm]')
title('Healed Porosity')
colormap(cmap); freezeColors; % also consider 'bone'
hold on
set(gca,'YDir','normal') % affects axes properties within figures
axis equal, box on, axis tight, grid on
title('Image Stitch 4306 FA')
% Visualize Porosity Maps

% subplot(1,5,2)
axes(ax(2));
h = image(Xo(:,1),Yo(:,1),Mo); % open
hold on
xlabel('X-position [cm]'); % ylabel('Y-position [cm]')
title('Open Porosity Map')
colormap(cmap); freezeColors; % also consider 'bone'
set(gca,'YDir','normal') % affects axes properties within figures
axis equal, box on, axis tight, grid on
hold on

% subplot(1,5,3)
axes(ax(3));
% image(Xo(:,1),Yo(:,1),Mo); % open
% colormap(cmap)
% hold on
image(Xh(:,1),Yh(:,1),A*Mh); % healed
colormap(cmap); freezeColors
title('Healed Porosity Map')
set(gca,'YDir','normal') % affects axes properties within figures
axis equal, box on, axis tight, grid on
xlabel('X-position [cm]'), % ylabel('Y-position [cm]')

% NOTE: Consider Visualize matrix, open porosity, and healed porosity
% in single tri-colored image. Since both data sets are binary 0 or 1,
% then adding a fraction of one image to the other produces a tri-
% indexed image of 0,A,1, where A is the fraction. This will only work
% if the two image matrices are exactly the same size.
% image(Xo(:,1),Yo(:,1),Mo + A*Mh); % open
% --> option currently unavailable... pixel dimensions do not match

% subplot(1,5,5)
axes(ax(5));
% image(Xo(:,1),Yo(:,1),Mo); % open
% colormap(cmap)

```

```

%      hold on
% Only works if two images are the exact same sized matrices;
% assumes binary images
% Ihm = find(Mh == max(max(Mh)));
% Iom = find(Mo == max(max(Mo)));
Ihp = find(Mh == min(min(Mh)));
Iop = find(Mo == min(min(Mo)));
% Mhp = Mh; Mhp(Ihm) = 0; Mhp(Ihp) = 165;
% Mop = Mo; Mop(Iom) = 0; Mop(Iop) = 0;
% Mho = Mop + Mhp;
% Mho(Ihm) = 0; Mho(Iom) = 0;
Mho = ones(size(Mh));
Mho(Ihp) = 2;
Mho(Iop) = 3;

image(Xh(1,:),Yh(:,1),Mho); % healed
colormap([1 1 1;0.1 0.5 0.1;0 0 0.9])
title('Open + Healed Porosity Map')
set(gca,'YDir','normal') % affects axes properties within figures
axis equal, box on, axis tight, grid on
xlabel('X-position [cm]'), % ylabel('Y-position [cm]')
box on

% Save memory
% clear Ihp Iop Mho

end

%% ROTATE IMAGE FOR TRANSECT SAMPLING
% --> used to build transects that are normal to the fracture trace
% across the thin section
% Two approaches: (1) rotate image using rotation matrix
%                  (2) rotate transect data after calcualted

% (OPTIONAL CHECK: Visualize rotated image)

%%%%%%%%%%%%%%%
%% STEP 3: CALCULATE TRANSECT DATA
%%%%%%%%%%%%%%%

% STEP 3.1: Calculation
% Trasect parallel to y-axis, x=A, where A is taken to be the center
% --> could be repeated for multiple x positions in order to
% characterize robustness of distribution laterally as a function of
% position along the fracture...
x = max(max(Xo))/2; %find center x, scalar

% NOTE:
% - dy and ds defined at top of file structure for ease of
%   manipulations
% - For faster testing use large dy, set dy to 0.1
%
% (STEP 1)
% dy = 0.01; % window step increment in cm, 0,.01 preferred?
% ds = 0.1; % window edge length in cm

% Open Porosity ("P" for porosity, "o" for open)
Po = ScanlineY(Xo,Yo,Mo,x,dy,ds); % gathers structured array generated
% by transect analysis carred out in
% ScanlineY
% Healed Porosity ("P" for porosity, "h" for healed)
Ph = ScanlineY(Xh,Yh,Mh,x,Po.y,ds);

```

```

% Reference total porosity of whole slide
% - arbitrarily depends on the slide dimensions relative to the
%   fracture size
% - if image does not span whole domain, the value can be contaminated
%   by meaningless pixels along the edges of the slide
Po.p_all = 1-sum(sum(Mo/max(max(Mo))))/numel(Mo);
Ph.p_all = 1-sum(sum(Mh/max(max(Mh))))/numel(Mh);
Pskeletal= Po.p_all + Ph.p_all;

% STEP 3.2: Visualization
% Transect line
% subplot(1,5,3)
if Plotting == 'Y';
    axes(ax(3));
    % Plot dots representing window centers from transect
    plot(Po.x,Po.y,'.k','MarkerSize',5)
    % Visualize example sample box
    dummyx = Po.x(7); dummyy = Po.y(7);
    dummyX = [dummyx-0.5*ds;dummyx-0.5*ds;...
               dummyx+0.5*ds;dummyx+0.5*ds;dummyx-0.5*ds];
    dummyY = [dummyy-0.5*ds;dummyy+0.5*ds;...
               dummyy+0.5*ds;dummyy-0.5*ds;dummyy-0.5*ds];
    plot(dummyX,dummyY,'k-')
    % Insert text recording number of pixels within a typical window
    text(Po.x(1)+0.5*ds,Po.y(1),['n=',num2str(Po.bn(5))])

    % subplot(1,5,4)
    axes(ax(4));
    % Porosity along scanline
    % open
    hp1 = plot(Po.p0,Po.y,'-', 'LineWidth',1.5, 'Color', MyColor(2,:));
    hold on
    % healed
    hp2 = plot(Ph.p0,Ph.y,'-', 'LineWidth',1.5, 'Color', MyColor(1,:));
    % skeletal
    hp3 = plot(Po.p0 + Ph.p0,Ph.y,'r-','LineWidth',1);
    % Reference
    hp4 = plot(ones(size(Po.y))*Po.p_all,Po.y,'--','Color',MyColor(2,:));
    hp5 = plot(ones(size(Po.y))*Ph.p_all,Po.y,'--','Color',MyColor(1,:));
    hp6 = plot(ones(size(Po.y))*Pskeletal,Po.y,'r--');

    % Common formatting
    box on, grid on
    xlabel({'Fractional';'Porosity []'})
    title({'Porosity Variation';['w = ',num2str(ds), ' [cm]']})
    xlim([0 1])
    ylim([0 max([max(max(Yo));max(max(Yh))])])
    legend([hp1 hp2 hp3 hp4 hp5 hp6], 'Open', 'Healed', 'Skeletal',...
           'Avg Open', 'Avg Healed', 'Avg Skeletal')

    axes(ax(1)), ylim([0 max([max(max(Yo));max(max(Yh))])])
    axes(ax(2)), ylim([0 max([max(max(Yo));max(max(Yh))])])
    axes(ax(3)), ylim([0 max([max(max(Yo));max(max(Yh))])])
    axes(ax(5)), ylim([0 max([max(max(Yo));max(max(Yh))])])
    % Ensure plot aspect ratios match
    sc = get(ax(2), 'PlotBoxAspectRatio');
    set(ax(4), 'PlotBoxAspectRatio',sc);
    set(ax(5), 'PlotBoxAspectRatio',sc);

    % linkaxes(ax) % option to link x and y axes so if zoom or pan in one
    % subplot, all subplots zoom/pan in same way - maintain same axes
end

%%%%%%%%%%%%%

```

```

%% STEP 4: Fracture/Pore Edges in BINARY IMAGES %
% OR Look for edge-finding algorithms to implement.

% CALCULATE GRADIENT TO FIND EDGES and ROUGHNESS
% [Fx,Fy] = gradient(M,(X(1,:)),Y(:,1));

%%%%%%%%%%%%%
%% STEP 5: PORE SIZE and SHAPE ANALYSIS STATISTICS %
%%%%%%%%%%%%%
% - For whole slide
% - within sliding window; do pore characteristics change as a function
%   of distance from the fracture surface?
% PORE SHAPE ANALYSIS STATISTICS

%%%%%%%%%%%%%
%% STEP 6: Sensitivity analysis to Window Dimensions %
%%%%%%%%%%%%%

if T_sensitivity == 'Y';
% STEP 6.1: TRANSECT SENSITIVITY TO BOX SIZE and STATISTICS (SQUARE):
% --> How does the apparent porosity distribution depend on the window
% size used to measure porosity along the transect?
% -----
% Open/Healed Porosity/Skeletal

% Define window size range
ds_test = [0.001,0.01:0.01:0.1,0.2,0.3]; % in cm
% ds_test = [0.01,0.05,0.1];

% Define variables to store data
Ptest.p = zeros(length(Po.y),length(ds_test));
Ptest.y = Ptest.p;
Ptest.n = zeros(size(ds_test));
Ptest.q = cell(size(ds_test));
Ptest.labels = Ptest.q;
% Ptest.q = Ptest.n;

%% STEP 6.2: GATHER STATISTICS
for i = 1:length(ds_test)
    % use same x and Po.y as above for consistency
    Pdummy = ScanlineY(Xh,Yh,Mh,x,Po.y,ds_test(i));
    Ptest.p0(:,i) = Pdummy.p0; % matrix with a column of porosity
                                % measurements for each window edge length
                                % tested
    Ptest.y(:,i) = Po.y; % matrix with a column of y positions for each
                        % window edge length tested
    Ptest.n(i) = numel(Pdummy.in{1}); % number of pixels used to calculate
                                    % porosity
    % Investigates how variable porosity is within a transect for a given
    % window size.
    Ptest.q{i} = quartile(Pdummy.p0); % variability of porosity among
                                    % measurements for each window size
    Ptest.labels{i} = num2str(ds_test(i));
end
clear Pdummy

%% STEP 6.3: VISUALIZE RESULTS
if Plotting == 'Y';

```

```

ht1 = figure;
subplot(1,2,1)
% Simple version of plot whole matrix as line series, but gives no
% control over line color
% htest = plot(Ptest.p0,Ptest.y,'-');
MyLineColor = cool(length(ds_test));
htest = zeros(size(ds_test))*nan;
for i = 1:length(ds_test)
    hdummy = plot(Ptest.p0(:,i),Ptest.y(:,i),...
        '-','Color',MyLineColor(i,:));
    htest(i) = hdummy;
    hold on
end
box on, grid on
xlabel({'Fractional';'Porosity []'})
xlim([0 1])
ylabel('Distance along Transect [cm]')
title({'Sensitivity to Window Size';...
    ['Pixel Size = ',num2str(1/DP),' cm']})

% Legend
hl = legend(htest,{num2str(ds_test)} );
v = get(hl,'title');
set(v,'string','Window Size [cm]');

% Plot stats for each window
subplot(2,2,2)
plot(log10(ds_test),Ptest.n,'r-')
boxplot(Ptest.p0,'notch','on','colors',MyLineColor,...
    'labels',Ptest.labels,'labelorientation','inline',...
    'outliersize',4,'symbol','k+')
ylabel('Fractional Porosity of Windows []')
xlabel('Window Size [cm]')
ylim([0 1]);
box on, grid on

% Whole Image Porosity

% Average of Transects
subplot(2,2,4)
% Plot mean and standard deviation of porosity values from the
% collection of window calculations for each window edge length
e_value = std(Ptest.p0);
errorbar(log10(ds_test),mean(Ptest.p0),e_value,e_value,'ko');
hold on
scatter(log10(ds_test),mean(Ptest.p0),10^2, ...
    MyLineColor,'filled','MarkerEdgeColor','k')

xlabel('log_{10}(Window Size) [cm]')
ylabel('Fractional Porosity []')
title('Porosity variation as f(Window Size)')
ylim([0 1])
box on, grid on
% Statistics [PLACEHOLDER]
% Plot 25th, 50th, and 75th quartiles as lines Ptest.q
% --> Implement modified boxplot developed for well log data for Batir

end
end % end of optional test

```

%%%%%%%%%%%%%%

```

%% STEP 7: Comparison of porosity characteristics among samples      %%
%% (for another, separate script)
%
% Some Characteristics to consider
%
% A. METHOD COMPARISON
% 1. correspondence among methods
% 2. does the compatibility of measurements of porosity by different
%    methods depend on the stage of the fracture?
% --> ntoes
%
% B. GEOLOGIC COMPARISONS
% 1. Dependence of porosity change on fracture stage
% 2. Width of "changed" porosity in rock adjacent to the fracture
%   - amount of open and healed
%   - pore size distribution
%   - pore shape
%   - pore alignment (anisotropy; all elongated and elongated in same
%     direction)

% -----[PLACEHOLDER]-----

```

```

%%%%% STEP 8: OUTPUTS: SAVE VARIABLES AND FIGURES %
%%%%% STEP 8.1:
% Define file name and name of structured array for export to .mat file
% fname = strrep(FName_image,'binimage',''); %string replace function
[pathstr,fname,ext] = fileparts(fname);
% Collect variables to save based on Filename
eval(['GEON2_',fname,'.Ph = Ph;']);
eval(['GEON2_',fname,'.Po = Po;']);
eval(['GEON2_',fname,'.Pskeletal = Pskeletal;']);
eval(['GEON2_',fname,'.ds = ds;']);
eval(['GEON2_',fname,'.dy = dy;']);
if T_sensitivity == 'Y'
    eval(['GEON2_',fname,'.Ptest = Ptest;']);
end
eval(['out = GEON2_',fname,';']);

% Save Variables from workspace to current working direction
if Data_export == 'Y' || Figure_export == 'Y' || T_sensitivity == 'Y'
    if Data_export == 'Y';
        % NOTE: can force data to be stored to a desired sub-directory via:
        % modifiable to save specific variables from workspace via:
        % save(FILENAME,VARIABLES)
        % OR
        % working_dir = cd;
        % directory_for_storage = [working_dir,'\data_archive\'];
        % save([directory_for_storage,'GEON2_',fname],data);
        save([pathstr,'GEON2_',fname,'.mat'],['GEON2_',fname],'-mat');

        % NOTE: To recover data in another script simply use:
        % load(FileNamesString)
        % e.g.,
        % To load saved variables in .mat file to the workspace
        % load('GEON2_3617PFWA.mat');
    end
    % Save Figures as .fig and PDF to current working directory
    folder = [pwd,'/FIGURES/'];

```

```

if Plotting == 'Y'
    if Figure_export == 'Y'
        % Save Figure as .fig file - manipulatable in Matlab
        saveas(hf2,[folder,'FIG - GEON2_',fname,'_Trans.fig']);
        saveas(hf2,[folder,'FIG - GEON2_',fname,'_Trans.pdf']);
    end
    if T_sensitivity == 'Y';
        saveas(ht1,[folder,'FIG - GEON2_',fname,'W_sensitivity.fig']);
        saveas(ht1,[folder,'FIG - GEON2_',fname,'W_sensitivity.pdf']);
    end
end
%% Define hf and ht handle for figures generated by function if they exist
hf = [];
if exist('hf1') == 1;
    hf = [hf hf1];
end
if exist('hf2') == 1;
    hf = [hf hf2];
end
if exist('hf3') == 1;
    hf = [hf hf3];
end
ht = [];
if exist('ht1') == 1;
    ht = [ht ht1];
end

```

High Resolution Image Analysis Multiple Transect Function (2 of 2):

```
function [out] = PorosityImageMultiScanline(fname,DP,ds,dx,diameter,...  
Plotting)  
  
% function [out] =  
% PorosityImageMultiScanline(fname,DP,ds,dx,diameter,range,Plotting)  
%  
% AUTHOR: NCD  
% CREATED: 2013-06-11  
% VERSION: v1  
%  
% PURPOSE:  
% To generate multiple porosity scanlines in order to compile statistics  
% on the variability of porosity within the thin section  
% Could also provide the basis for defining variability in damage zone  
% with based on a porosity generation criterion  
%  
% INPUTS  
% fname = base name of bitmap files (can include path to subdirectory  
% containing the files  
% DP = resolution of bitmaps in dots/cm  
% ds = window edge length  
% dx = 0.5 % spaceing between transects in cm  
% if dx is:  
% scalar - specifies the spacing between transects  
% 'Single' - requests a single transect at the center of the  
% bitmap  
% [x,y] - specifies a location for a single transect  
% [x1,y1;x2,y2;...] - is a column vector of x and y  
% coordinates explicitly specifying the positions of  
% transects  
%  
% diameter of sample for use in constructing transects  
% range = [imin imax]; min and max indices of bitmaps comprising the CT  
% scan  
% Plotting= 'Y' creates summary plots including  
% (1) plots the open and healed transect data including  
% boxplots summarizing the variation in porosities by  
% bitmap and by smoothed transect corresponding to the  
% "moving" average represented by a cube with sides ds  
% moving through the sample at a spacing of 1/2*ds  
% (2) location of the transects within a wireframe of the  
% outline of the bitmaps  
%  
% OUTPUTS  
% out is a structured array containing the following variables:  
% out.xy = trans_xy;  
% out.Z = Z; clear Z  
% out.Po = Po; clear Po  
% out.Ph = Pg + (Pl - out.Po); clear Pg Pl  
% out.Poss= Pos; clear Pos % smoothed data  
% out.Phss= Pgs + (Pls - out.Pos);  
% out.ds = ds;  
% out.DP = DP;  
% out.nf = nf;  
% out.N = N; % number of pixels within each window; sqrt(N) is the  
% % number of pixles in the row or column directions, which  
% % also correspond to the x and y directions in the local  
% % coordinate system  
% out.edge=edge; clear edge  
%
```

```

% out.Poall = Poall; clear Poall
% out.Phall = Pgall + (Plall - out.Poall); clear Pgall Plall
% out.Nall = Nall; clear Nall
%
% SYNTAX
% fname = ['DATA_Image/3937FAbin'];
% DP = 9605; % bitmap resolution in points/cm; assumed same in x,y,z
%             % for a 26.7 micron pixel
% ds = 0.05; % window edge length
% dx = 0.8; %0.5; % space between transects in the x and y direction
% diameter= 1.5; % sample diameter in cm
% Plotting= 'Y';
%
% [out] = PorosityImageMultiScanline(fname,DP,ds,dx,diameter,Plotting)
%
%
% NOTE: Resolution in vertical and horizontal:
% DP and DP assumed to be the same
%
% Based on modification of FUNCTION: PorosityImageAnalysis
%

%% STEP 1.1: I/O
if exist('DP','var') == 0
    DP = 9605 % max resolution of thin section image at 4x
end
if exist('dx','var') == 0
    dx = 0.2*ds %
end
if exist('Plotting') == 0
    Plot = 'N'
end
% File base name
[pathstr, fnamestr, ext] = fileparts(fname);

%% STEP 1.2: Import Data
% Note: All three images are currently loaded at the same time; this
% corresponds to the most memory intensive approach, sequential loading
% and analysis would be faster if memory is limited/the files are very
% large
% folder = [pwd,'/DATA_Image/'];
coda = '.tif'; % specify file extension type

% Mi = imread([folder,fname,'_image',coda]);
% Mo = imread([folder,fname,'_open',coda]);
% Mh = imread([folder,fname,'_healed',coda]);
Mi = imread([fname,'_image',coda]);
Mo = imread([fname,'_open',coda]);
Mh = imread([fname,'_healed',coda]);

%% STEP 1.3: Scale Image Data (Transform from TIFF/BITMAP Coordinates to
%             Spatial Coordinates)
% Scale by Dots Per Centimeter
[Xi,Yi] = ScaleImage(Mi,DP);
[Xo,Yo] = ScaleImage(Mo,DP);
[Xh,Yh] = ScaleImage(Mh,DP);

% automatically determine the "diameter" from the scaled extent of the
% image if diameter is not manually entered
if isempty(diameter) == 1
    diameter = abs(max(max(Xi)) - min(min(Xi)));
end

```

```

%% STEP 2.0: Define Series of x,y coordinates for axis parallel transects
[ndx] = numel(dx);
if ndx > 1
    if dx == 'Single'
        trans_x = [diameter/2, diameter/2];
    elseif isstr(dx) == 0 && ischar == 0
        % Specify either a [x,y] position for a transect or a set of x,y
        % pairs for transects
        trans_x = dx;
    else
        error(['dx is incorrectly specified please choose: ...
            'dx = scalar, dx = ''Single'', dx = [x,y], ...
            'dx = [x1,y1;x2,y2;...]')
    end
end

% Find min and max x positions so that sample window is completely within
% the sample
% Have transects overlap by 1/2 the window length
if isempty(dx) == 1
    dss = 0.5*ds;
else
    dss = dx;
end
xl = dss;
xr = diameter - dss;

trans_x = [xl:dss:xr];

if isempty(trans_x)
    error(['No transect x-coordinates generated for dx = ',num2str(dx)])
end

%% STEP 3.1: Calculate Porosity
out.Pos = []; out.Phs = out.Pos; out.x = out.Pos; out.yy = out.Pos;
dy = 0.25*ds;
for i = 1:length(trans_x)
    tstart = tic;
    % Open Porosity ("P" for porosity, "o" for open)
    Pos = ScanlineY(Xo,Yo,Mo,trans_x(i),dy,ds); % gathers structured array
                                                    % generated by transect analysis
                                                    % carried out in the function:
                                                    % "ScanlineY"
    % Healed Porosity ("P" for porosity, "h" for healed)
    Phs = ScanlineY(Xh,Yh,Mh,trans_x(i),Pos.y,ds);

    % Collected data in Structured Array
    out.x = [out.x,Pos.x'];
    if i == 1;
        out.xrange = [min(min(Xi)) max(max(Xi))];
        out.yrange = [min(min(Yi)) max(max(Yi))];
        out.trans_x = trans_x;
        out.y = Pos.y';
        out.ds = ds;
        out.DP = DP;
        % out.nf = Pos.nf;
        % out.N = Pos.N;
    end
    out.yy = [out.yy,Pos.y'];
    out.Pos = [out.Pos,Pos.p0'];
    out.Phs = [out.Phs,Phs.p0'];
end

```

```

tpast = toc(tstart);
disp(['--> Transect ',num2str(i), ' of ', num2str(length(trans_x)),...
' complete after ', num2str(tpast), ' seconds']);
end
out.Z = out.y;
out.fnamestr = fnamestr;

% Save Iamge Data
out.Xi = Xi; out.Yi = Yi; out.Mi = Mi; out.Mo = Mo; out.Mh = Mh;

%% STEP 4.0: Export
dummystr = [pathstr,'/','IMAGEmulti_',fnamestr,'.mat'];
file_test = exist(dummystr, 'file');
N = 0;
while file_test == 2
    dummystr = [pathstr,'/','IMAGEmulti_',fnamestr,num2str(N),'.mat'];
    N = N+1;
    file_test = exist(dummystr, 'file');
end
% save([pathstr,'/','IMAGEmulti_',fnamestr,'.mat'],['out'],'-mat')
save(dummystr,['out'],'-mat')

%%%%%%%%%%%%%
%% STEP 5.0: PLOT of RESULTS
%% %%%%%%
if Plotting == 'Y'
% 5.1 PLOT OF ANALYSIS
figure
[m,n] = size(out.x);
xx = [1:1:n];
MyFormatBox = ...
    {'notch','on','colorgroup',[],'outliersize',3,'symbol','k+'};
MyFormatLine = ...
    {'d-k','MarkerSize',10,'MarkerFaceColor','y','LineWidth',1.5};

yy = out.yy; % repmat(out.y,1,length(trans_x));
subplot(1,4,1)
% plot(out.Pos,out.y,'-', 'LineWidth',2);
plot(out.Pos,yy,'-', 'LineWidth',2);
hold on
xlabel({'Fractional Porosity';['ds = ',num2str(out.ds), ' [cm]']})
ylabel('z-position [cm]')
title({'Open PorosityL:';fnamestr})
colormap(jet)
% plot(out.Poall,out.y,'k--','LineWidth',1.5)
% xlim([0 1]); box on; grid on;
subplot(1,4,2)
% plot(out.Phs,out.y,'-', 'LineWidth',2);
plot(out.Phs,yy,'-', 'LineWidth',2);
hold on
xlabel({'Fractional Porosity';['ds = ',num2str(out.ds), ' [cm]']})
ylabel('z-position [cm]')
title({'Healed Porosity:';fnamestr})
colormap(jet)
% plot(out.Phall,out.Z,'k--','LineWidth',1.5)
% xlim([0 1]); box on; grid on;

% Statistical Analysis
% subplot(2,4,3)
% boxplot(out.Ph,MyFormatBox{::})
% ylim([0 1])

```

```

%
%     ylabel('Fractional Porosity []')
%     title('Healed - Per Bitmap')
%     hold on
%     plot(xx,mean(out.Ph),MyFormatLine{::})
%     % plot(xx,mean(out.Phall)*ones(size(xx)), 'k--','LineWidth',1.5)
% subplot(2,4,4)
%     boxplot(out.Phs,MyFormatBox{::})
%     hold on
%     ylim([0 1])
%     title('Healed - Smoothed')
%     plot(xx,mean(out.Phs),MyFormatLine{::})
%     % plot(xx,mean(out.Phall)*ones(size(xx)), 'k--','LineWidth',1.5)
% subplot(2,4,7)
%     boxplot(out.Po,MyFormatBox{::})
%     hold on
%     ylabel('Fractional Porosity []')
%     ylim([0 1])
%     title('Open - Per Bitmap')
%     xlabel('Transect')
%     plot(xx,mean(out.Po),MyFormatLine{::})
%     % plot(xx,mean(out.Poall)*ones(size(xx)), 'k--','LineWidth',1.5)
% subplot(2,4,8)
%     boxplot(out.Pos,MyFormatBox{::})
%     hold on
%     ylim([0 1])
%     title('Open - Smoothed')
%     xlabel('Transect')
%     plot(xx,mean(out.Pos),MyFormatLine{::})
%     % plot(xx,mean(out.Poall)*ones(size(xx)), 'k--','LineWidth',1.5)

% 5.2 TRANSECT Position QA
figure
% image(Xh,Yh,Mh);
image(Xh(1,:),Yh(:,1),Mh);

%     surface('XData',[0 1; 0 1],'YData',[0 0; 1 1],...
%             'ZData',[1 1; 1 1],'CData',flipdim(imageData,1),...
%             'FaceColor','texturemap','EdgeColor','none');

set(gca,'YDir','normal') % affects axes properties within figures
hold on
plot(out.x, out.yy,'.'), axis equal, grid on
xlabel('x'), ylabel('y')
title('Scanline Positions on Healed Porosity Map')
cmap = [0.0 0.4 0.0 ;0.1 0.5 1.0; 1.0 1.0 0.7];
colormap(cmap)

% 5.3 TRANSECT Position QA - 3D
figure
subplot(1,2,1)
plot3(out.x,out.yy,out.Pos,'-'), box on, grid on
set(gca, 'DataAspectRatio', ...
    [diff(get(gca, 'XLim')) diff(get(gca, 'XLim')) ...
    diff(get(gca, 'ZLim'))])
xlabel('x'), ylabel('y'), zlabel('Open Porosity');
view(-50, 70)

subplot(1,2,2)
% Plot the image for reference
surface(Xh(1,:),Yh(:,1),zeros(size(Xh)),Mh, ...
    'FaceColor','texturemap','EdgeColor','none');
% ALTERNATE Plotting options

```

```

% image(Xh(:, :, 1), Yh(:, 1), Mh);
% surface(Xh, Yh, zeros(size(Xh)), ...
%     Mh, 'FaceColor', 'texturemap', 'EdgeColor', 'none');
% surf(Xh, Yh, zeros(size(Xh)), Mh);

set(gca, 'YDir', 'normal') % affects axes properties within figures
cmap = [0.0 0.4 0.0 ; 0.1 0.5 1.0; 1.0 1.0 0.7];
colormap(cmap)
hold on
xlabel('x'), ylabel('y')
title('Scanline Positions on Healed Porosity Map')
view(-50, 70)

% Plot the scanlines
plot3(out.x, out.yy, out.Phs, '-'), box on, grid on
set(gca, 'DataAspectRatio', ...
    [diff(get(gca, 'XLim')) diff(get(gca, 'XLim')) ...
    diff(get(gca, 'ZLim'))])
xlabel('x'), ylabel('y'), zlabel('Healed Porosity');
view(-50, 70)
end

end

%%%%%%%%%%%%%
%% SUBFUNCTIONS
%%%%%%%%%%%%%
-----%
%% SUBFUNCTION: Transect position sampler
function [xy, ds, dx] = TransectSampler(diameter, ds, dx)

% Assume that [x,y] = [0,0] is diameter/2 from the center
xc = diameter/2;

% Find corners of square inscribed within circular cross section of sample
xl = s;
xr = diameter - s;

% Have transects overlap by 1/2 the window length
if isempty(dx) == 1
    dss = 0.5*ds;
else
    dss = dx;
end
xrange = [xll:dss:xur];

% cross correlate to fully sample region at equal intervals (lots of
% transects!
[X, Y] = meshgrid(xrange, yrangle);
% reshape into a simple vector of xy pairs
[m, n] = size(X);
x = reshape(X, m*n, 1);
y = reshape(Y, m*n, 1);
xy= [x, y];

end % end of subfunction

-----%
%% SUBFUNCTION: Import data
-----PLACEHOLDER-----

```


Micro CT Script

```
% TITLE: MASTER_XRayCT_Analysis.m
%
%
%
% AUTHOR: Justin Roth/Nick Davatzes
% DATE Created: 1/14/13
% DATE: 2013-05
%
%
% PURPOSE: This script will import and analyse binary bitmaps that
% represent 1) healed porosity with a greater density than the surrounding
% rock, and 3) open porosity within host rock, 2) healed porosity with less
% density than the surrounding host the host rock.
%
%
% NOTES on pre-process of input data: CTan was used to apply thresholding
% of the voxels. Variables can be created, saved, and added together to
% easily plot healed,open, and skeletal porosity.
%
%
% NOTES:
% - Be sure all bitmaps are exactly the same size (i.e., have the same
% number of rows and columns of pixels) and the 1,1 pixel is registered to
% equivalent x,y coordinate positions. If not, two problems are likely:
% 1. The images might be slightly shuffled so that pixels in adjacent
% layers are mismatched
% 2. Filtering of the data (i.e., sub-sampling) for the transect based on
% a spatial position (rather than a row x column position) might
% occasionally lead to different numbers of pixels within adjacent layers
% being chosen, thus leading to a concatenation problem in assembling the
% 3D matrix, Xfin,Yfin,Zfin.
%
%
% Future Modifications: Eventually, each subtask should be written as a
% function called from a MASTER file.
clear all, close all

%%%%%%%%%%%%%%%
%% STEP 1: Preliminary Selections and Variables: %%
%%%%%%%%%%%%%%%

%% STEP 1.1: Set resolution
DP = 374; % resolution as dots per cm
DPz= DP;

Zc3Din = [];
% Pre-Condition Spatial Transformation Variables
Xin =[];Yin =[];Zin=[];
Xfin=[];Yfin=[];Zfin[];

% Pre-Condition Filtering Variables
% Spatial Filtering

% Transect position:
Xt = 2.54/2; % x-coordinate of the z-axis parallel transect [cm]
Yt = 2.54/2; % y-coordinate of the z-axis parallel transect [cm]

% Data analyzed within given distances to transect position:
% Note: These do not correspond to the size of the transect window, which
% is given as ds below
dx = .1; % half length of data sampled from XRayCT scan in X,Y only [cm]
dy = dx;

% Scan Window
```

```

ds = 20/DP; % defines box size (cm) and is about 0.05 cm
% ds = .2 %defines box size (cm)
% ds = 5/DP % equates to a 0.0013 cm window
% ds = sqrt(700)/DP % equivalent number of points as in point count

%% STEP 1.2: CHOOSE THE SAMPLE To ANALYZE: File Name Bitmap Specification

%% -----SAMPLE N2-4302FA (18-1557)-----%%
% SampleBaseNameGR = 'N24302FA_TrimBin_Denser'; %"GR" for 'greater density'
% SampleBaseNameLS = 'N24302FA_TrimBin_Less'; %"LS" for 'less-density'
% SampleBaseNameOP = 'N24302FA_TrimBin_Open'; %"OP" for 'open porosity'
%
% imin = 179; %arbitrary desired minumum for the trimmed dataset
% imax = 1295; %arbitrary desired maximum for the trimmed dataset

%% -----SAMPLE N2-3937F (18-2082)-----%%
SampleBaseNameGR = 'N23937F_TrimBin_Denser'; %"GR" for 'greater density'
SampleBaseNameLS = 'N23937F_TrimBin_Less'; %"LS" for 'less-density'
SampleBaseNameOP = 'N23937F_TrimBin_Open'; %"OP" for 'open porosity'

imin = 18; %Not yet trimmed
imax = 2082; %Not yet trimmed

%% -----SAMPLE N2-4306F (18-2088)-----%%
% SampleBaseNameGR = 'N24306F_TrimBin_Denser'; %"GR" for 'greater density'
% SampleBaseNameLS = 'N24306F_TrimBin_Less'; %"LS" for 'less-density'
% SampleBaseNameOP = 'N24306F_TrimBin_Open'; %"OP" for 'open porosity'
%
% imin = 18; %Not yet trimmed
% imax = 2088; %Not yet trimmed

%% -----SAMPLE N2-4306F (18-2088)-----%%
% SampleBaseNameGR = 'N24306F_TrimBin_Denser'; %"GR" for 'greater density'
% SampleBaseNameLS = 'N24306F_TrimBin_Less'; %"LS" for 'less-density'
% SampleBaseNameOP = 'N24306F_TrimBin_Open'; %"OP" for 'open porosity'
%
% imin = 18; %Not yet trimmed
% imax = 2088; %Not yet trimmed

%%-----Sample Specification-----%%
%% STEP 1.3: SELECTING THE PLOTS AND ANALYSES TO OUTPUT
% 1.3A: "GR" for thresholded 'greater density bitmaps'
% Plots the untrimmed bitmaps (greater density than host)
    Plot_greater_bitmaps = 'N';
% Plots the selected trimmed volume of interest (greater density than host)
    Plot_greater_bitmapsSUB = 'N';
% Plots selected trimmed volume of interest (greater density than host)
% as a surf plot (faster)
    Plot_greater_bitmapsSUBCAT = 'N';
% Plots a transect of information about the selected trimmed volume of
% interest (greater density than host)
    Plot_Greater_Transect = 'Y';

% 1.3B: "LS" for thresholded 'less-density bitmaps'
% Plots the untrimmed bitmaps (less density than host)
    Plot_lesser_bitmaps = 'N';
% Plots the selected trimmed volume of interest (less density than host)
    Plot_lesser_bitmapsSUB = 'N';
% Plots selected trimmed volume of interest (less density than host)
% as a surf plot (faster)
    Plot_lesser_bitmapsSUBCAT = 'N';

```

```

% Plots a transect of information about the selected trimmed volume of
% interest (less density than host)
    Plot_lesser_Transect = 'N';

% 1.3C: "OP" for thresholded 'open porosity bitmaps'
% Plots the untrimmed bitmaps (open porosity)
    Plot_open_bitmaps = 'N';
% Plots the selected trimmed volume of interest (open porosity)
    Plot_open_bitmapsSUB = 'N';
% Plots selected trimmed volume of interest (open porosity)
% as a surf plot (faster)
    Plot_open_bitmapsSUBCAT = 'N';
% Plots a transect of information about the selected trimmed volume of
% interest (open porosity)
    Plot_open_Transect = 'N';

% MORE RELEVANT PLOT OPTIONS:
    Plot_ALL_POROSITY_TRANSECT = 'Y';
    Plot_HEALED_POROSITY_TRANSECT = 'N';
    Plot_OPEN_POROSITY_TRANSECT = 'N';
    Plot_SKEL_POROSITY_TRANSECT = 'N';

%%%%%%%%%%%%%%%
%% STEP 2: Materials with a GREATER DENSITY than host rock      %%
%%%%%%%%%%%%%%%

% Parameters for number of bit map files to read
imod = imin-1;
nf = imax - imod; % number of files to read and analyze

%% STEP 2.1: IMPORT and Trim the dataset
folder = [pwd,'/DATA_XCT/'];
for i = imin:imax
    %% DATA IMPORT
    % pad with zeros to preserve number of digits representing count
    if i<=9 % could also use: dummy = num2str(n), nn=numel(n),
        % then directly consider the number of digits defining i
        pad = '000';
    elseif i<=99
        pad = '00';
    elseif i <=999
        pad = '0';
    else
        pad = '';
    end
    filename = [folder,SampleBaseNameGR,pad,num2str(i),'.bmp'];
    Zc3Din(:,:,:i-imod) = imread(filename);
    Zc3Din(:,:,:i-imod) = imread(filename);

%% STEP 2.2: CONVERT TIFF/BITMAP IMAGE COORDINATES TO SPATIAL COORDINATES
% Convert the row x colum x page positions to X x Y x Z positions giving a
% total of 4 3D matrices where every row,colum,page position gives a
% single value of x,y,z,color of the desired pixel

    % call to function for scaling based on pixel density
    [Xin(:,:,:i-imod),Yin(:,:,:i-imod)]=ScaleImage(Zc3Din(:,:,:i-imod),DP);
    Zin(:,:,:i-imod) = ones(size(Xin(:,:,:i-imod)))*(i-imod-1)*(1/DP));

    % DELETE data that is not needed useing the CLEAR VAR function
    % (doing the clear within the loop as a series of analyses reduced the
    % amount of data needed to be maintained in memory
end

```

```

%% STEP 2.3: Determine coordinates defining edges of the scan
% Capture the coordinate defining the physical edges of the scan (i.e., min
% and max x,y,z, which define the shape of the volume scanned in XRayCT,
% which includes the entire core and some surrounding empty sapce).
edge.all.x = [min(min(Xin)) max(max(Xin))]; % edge.x = [minx maxx];
edge.all.y = [min(min(Yin)) max(max(Yin))];
edge.all.z = [0 DPz*nf];
if isempty(dx)==1
    edge.scan.x = edge.all.x;
    edge.scan.y = edge.all.y;
    edge.scan.z = edge.all.z;
else
    edge.scan.x = [Xt-dx Xt+dx];
    edge.scan.y = [Yt-dy Yt+dy];
    edge.scan.z = edge.all.z;
end

%% STEP 2.4: VISUALIZATION
if Plot_greater_bitmaps == 'Y';
    %% Plotting of Data in Physical coordinates:
    figure
    [mx,my] = size(Xin(:,:,1));
    for i=1:imax-imod
        dummyx = reshape(Xin(:,:,i),mx*my,1);
        dummyy = reshape(Yin(:,:,i),mx*my,1);
        dummyz = reshape(Zin(:,:,i),mx*my,1);
        dummyzc = reshape(Zc3Din(:,:,i),mx*my,1);
        scatter3(dummyx,dummyy,dummyz,...
            3*ones(size(dummyz)),dummyzc,'filled')
        hold on
    end
    xlabel('Xin'); ylabel('Yin'), zlabel('Zin')
    box on, colorbar
    colormap bone
    %axis equal
end

%% STEP 2.5a: SCANLINE ATTRIBUTES
% STRATEGY:
% Query scanline volumes to find desired indices (FIND) within range of x,y
% x,y footprint of "box" (assume dx=dy) should match choice for 2D thin
% section analysis so they can be compared. Z could be single layer
% (e.g., just like a thinsection) or larger, such as equal to dx (the
% half-length of the box's edge)
%
% Since all bmp images are aligned, the Indices within the x and y ranges
% will be the same for all bitmap images. Thus we only need to perform the
% querry once to find the indices, and then subsample via FOR loop all the
% pages within the 3D matrices.

% Box Size: (full width, not half width): Specified above as part of Step 1
% Variable selection; e.g., ds = 20/DP;
% ds defines box halflength size (cm) and is about 0.05 cm

% Defining a center point in the square bitmap; scan along centerpoint
% center x point:
xmid = max(max(Xin(:,:,1)))/2; %center x point
ymid = max(max(Yin(:,:,1)))/2; %center y point

% NOTE: find works only on 2D matrices
% % Defines a box based on ds above
% I = find (Xin>=(xmid-0.5*ds) & Xin<=(xmid+0.5*ds) & ...

```

```

%     Yin >= (ymid-0.5*ds) & Yin<= (ymid+0.5*ds));
% - For 3d matrices, we use ind2sub (linear indices I, or row x column x
%   page as [Im,In,Ip]

% Finds pixels within a box based on the definition of "ds" above
Isub = ind2sub(size(Xin),...
    find(Xin>=(xmid-0.5*ds) & Xin<=(xmid+0.5*ds) & ...
    Yin >= (ymid-0.5*ds) & Yin<= (ymid+0.5*ds)));
% [Im,In,Ip] = ind2sub(size(Xin),...
%     find(Xin>=(xmid-0.5*ds) & Xin<=(xmid+0.5*ds) & ...
%     Yin >= (ymid-0.5*ds) & Yin<= (ymid+0.5*ds)));
xsub = Xin(Isub); %
ysub = Yin(Isub);
zsub = Zin(Isub);
csub = Zc3Din(Isub); %

%% STEP 2.5b: VISUALIZATION
if Plot_greater_bitmapsSUB == 'Y';
figure
[mx,my] = size(xsub,:,:1);
if Plot_greater_bitmapssUBCAT == 'Y';
    % reshape 2D matrices to vectors so that the variables are
    % compatible with teh limitations ofthe "scatter3" plotting
    % function
    scatter3(xsub,ysub,zsub,3*ones(size(zsub)),csub,'filled')
else % Plot as images
    for i=1:imax-imod % same as 1:nf
        % gives the number of elements in a sub-area of a single bitmap
        Iz = find(zsub==((i-1)*(1/DP)));
        % number of elements within the sub area
        nz = numel(Iz);
        % assumes have same number of elements in the x,y directions
        nzs = sqrt(nz);
        dummyx = reshape(xsub(Iz),nzs,nzs);
        dummyy = reshape(ysub(Iz),nzs,nzs);
        dummyz = reshape(zsub(Iz),nzs,nzs);
        dummyzc= reshape(csub(Iz),nzs,nzs);
        surf(dummyx,dummyy,dummyz,dummyzc)
        hold on
        shading flat
    end
end
axis equal
colorbar

xlabel('X-Position (cm)');...
ylabel('Y-Position (cm)'),zlabel('Z-Position (cm)')
box on, colorbar
colormap bone
title 'Volume of Chosen Sub-region (greater dense thresh)'
end

%% STEP 2.6: Calculating porosity %
Pgr = zeros(nf,1);
for i = 1:nf
    % Count number of pixels within the sub-area of each bitmap image
    % (i.e., page of teh 3D matrix)
    sub_area_pixels = numel(Isub)/nf; % note, if I/nf is not an integer,
                                         % will get an error in subsequent
                                         % calculations
    % Count number of pixels that are pores
    % Ipores = numel(find(csub(:,:,i)==0));% or write as
    Iz = find(zsub==((i-1)*(1/DP)));

```

```

I pores = sum(csub(Iz)==0);
% Calculate porosity in each bitmap (i.e., page of the 3D matrix)
% to give a per bitmap transect of porosity
Pgr(i) = I pores/sub_area_pixels;
end
zp = (1:nf)*(1/DP); % z-position of the porosity measurement
Pgr_ds_smooth = smooth(zp,Pgr,ds,'moving'); % moving window mean

% VISUALIZATION
if Plot_Greater_Transect == 'Y';
    figure
    h1 = plot(Pgr,zp,'k.');
    hold on
    h2 = plot(Pgr_ds_smooth,zp,'-r','LineWidth',1.5);
    xlabel('Healed Porosity (Greater Density Than Host)')
    ylabel('Z-position [cm]')
    box on, grid on
    h3 = plot([mean(Pgr) mean(Pgr)],[0 nf/DP],'r--','LineWidth',1.5);
    legend([h1 h2 h3],'2D porosity of each scan',...
        'Moving cube: ds',...
        'Mean of Volume')
    title(...,'Transect for Healed Minerals with Greater Density than Host Rock')
end

%%%%%%%%%%%%%
%% STEP 3: Materials with a LESS DENSITY than host rock %%
%%%%%%%%%%%%%

%% STEP 3.1: IMPORT and Trim the dataset
for i = imin:imax %trims the dataset
    %% DATA IMPORT
    % pad with zeros to preserve number of digits representing count
    if i<=9 % could also use: dummy = num2str(n), nn=numel(n),
        % then directly consider the number of digits defining i
        pad = '000';
    elseif i<=99
        pad = '00';
    elseif i <=999
        pad = '0';
    else
        pad = '';
    end
    filename = [folder,SampleBaseNameLS,pad, num2str(i),'.bmp'];
    % Zc3Din(:,:,i-imod) = imread(filename);
    Zc3Din(:,:,i-imod) = imread(filename);

    %% STEP 3.2: SPATIAL COORDINATES
    % Convert the row x colum x page positions to X x Y x Z positions giving a
    % total of 4 3D matrices where every row,colum,page position gives a
    % single value of x,y,z,color of the desired pixel

    % call to function for scaling based on pixel density
    [Xin(:,:,i-imod),Yin(:,:,i-imod)]=ScaleImage(Zc3Din(:,:,i-imod),DP);
    Zin(:,:,i-imod) = ones(size(Xin(:,:,i-imod)))*((i-imod-1)*(1/DP));

    % DELETE data that is not needed using the CLEAR VAR function
    % (doing the clear within the loop as a series of analyses reduced the
    % amount of data needed to be maintained in memory
end

%% STEP 3.3: Determine coordinates defining edges of the scan
% Capture the coordinate defining the physical edges of the scan (i.e., min

```

```

% and max x,y,z, which define the shape of the volume scanned in XRayCT,
% which includes the entire core and some surrounding empty sapce).
edge.all.x = [min(min(Xin)) max(max(Xin))]; % edge.x = [minx maxx];
edge.all.y = [min(min(Yin)) max(max(Yin))];
edge.all.z = [0 DPz*nf];
if isempty(dx)==1
    edge.scan.x = edge.all.x;
    edge.scan.y = edge.all.y;
    edge.scan.z = edge.all.z;
else
    edge.scan.x = [Xt-dx Xt+dx];
    edge.scan.y = [Yt-dy Yt+dy];
    edge.scan.z = edge.all.z;
end

%% STEP 3.4: VISUALIZATION
if Plot_lesser_bitmaps == 'Y';
    %% Plotting of Data in Physical coordinates:
    figure
    [mx,my] = size(Xin(:,:,1));

    for i=1:imax-imod
        dummyx = reshape(Xin(:,:,i),mx*my,1);
        dummyy = reshape(Yin(:,:,i),mx*my,1);
        dummyz = reshape(Zin(:,:,i),mx*my,1);
        dummyzc = reshape(Zc3Din(:,:,i),mx*my,1);
        scatter3(dummyx,dummyy,dummyz, ...
            3*ones(size(dummyz)),dummyzc,'filled')
        hold on
    end
    xlabel('Xin'); ylabel('Yin'), zlabel('Zin')
    box on, colorbar
    colormap bone
    %axis equal
end

%% STEP 3.5a: SCANLINE ATTRIBUTES
% Box Size Attributes
% --> Defined in STEP 2.5a --> Same for all sub-analyses
% ds = .05; %defines box size (cm)

%Defining a center point in the square bitmap:
% --> Defined in STEP 2.5a --> Same for all sub-analyses
% Center x point (already defined above):
%     % xmid = max(max(Xin(:,:,1)))/2; %center x point
%     % ymid = max(max(Yin(:,:,1)))/2; %center y point
% Defines a box based on ds above
Isub = ind2sub(size(Xin),...
    find(Xin>=(xmid-0.5*ds) & Xin<=(xmid+0.5*ds) & ...
    Yin >= (ymid-0.5*ds) & Yin<= (ymid+0.5*ds)));
xsub = Xin(Isub); %
ysub = Yin(Isub);
zsub = Zin(Isub);
csub = Zc3Din(Isub);

%% STEP 3.5b: VISUALIZATION
if Plot_lesser_bitmapsSUB == 'Y';
    figure
    [mx,my] = size(xsub(:,:,1));
    if Plot_lesser_bitmapsSUBSCAT == 'Y';
        % reshape 2D matrices to vectors so that the variables are
        % compatible with teh limitations ofthe "scatter3" plotting
        % function

```

```

        scatter3(xsub,ysub,zsub,3*ones(size(zsub)),csub,'filled')
    else % Plot as images
        for i=1:imax-imod % same as 1:nf
            % gives the number of elements in a sub-area of a single bitmap
            Iz = find(zsub==((i-1)*(1/DP)));
            % number of elements within the sub area
            nz = numel(Iz);
            % assumes have same number of elements in the x,y directions
            nzs = sqrt(nz);
            dummyx = reshape(xsub(Iz),nzs,nzs);
            dummyy = reshape(ysub(Iz),nzs,nzs);
            dummyz = reshape(zsub(Iz),nzs,nzs);
            dummyzc= reshape(csub(Iz),nzs,nzs);
            surf(dummyx,dummyy,dummyz,dummyzc)
            hold on
            shading flat
        end
    end
    axis equal
    colorbar
    xlabel('X-Position (cm)');...
    ylabel('Y-Position (cm)'),zlabel('Z-Position (cm)')
    box on, colorbar
    colormap bone
    title 'Volume of Chosen Sub-region (less dense thresh)'
end

%% STEP 3.6: Calculating porosity %
Pls = zeros(nf,1);
for i = 1:nf
    % Count number of pixels within the sub-area of each bitmap image
    % (i.e., page of teh 3D matrix)
    % note, if I/nf is not an integer, will get an error in subsequent
    % calculations
    sub_area_pixels = numel(Isub)/nf;
    % Count number of pixels that are pores
    % Ipores = numel(find(csub(:,:,i)==0));% or write as
    Iz = find(zsub==((i-1)*(1/DP)));
    Ipores = sum(csub(Iz)==0);
    % Calculate porosity in each bitmap (i.e., page of the 3D matrix)
    % to give a per bitmap transect of porosity
    Pls(i) = Ipores/sub_area_pixels;
end
zp = (1:nf)*(1/DP); % z-position of the porosity measurement
Pls_ds_smooth = smooth(zp,Pls,ds,'moving'); % moving window mean
if Plot_lesser_Transect == 'Y';
figure
    h1 = plot(Pls,zp,'k.');
    hold on
    h2 = plot(Pls_ds_smooth,zp,'-r','LineWidth',1.5);
    xlabel('Healed Porosity (LESS Density Than Host)')
    ylabel('Z-position [cm]')
    box on, grid on
    h3 = plot([mean(Pls) mean(Pls)],[0 nf/DP],'r--','LineWidth',1.5);
    legend([h1 h2 h3],'2D porosity of each scan',...
        'Moving cube: ds',...
        'Mean of Volume')
    title(...%
        'Transect for Healed Minerals with Less Density than Host Rock')
end

%%%%%%%%%%%%%

```

```

%% Step 4: Materials That are OPEN POROSITY
%%%%%
%% STEP 4.1: IMPORT and Trim the dataset
for i = imin:imax %trims the dataset
    %% DATA IMPORT
    % pad with zeros to preserve number of digits representing count
    if i<=9 % could also use: dummy = num2str(n), nn=numel(n),
        % then directly consider the number of digits defining i
        pad = '000';
    elseif i<=99
        pad = '00';
    elseif i <=999
        pad = '0';
    else
        pad = '';
    end
    filename = [folder,SampleBaseNameOP,pad, num2str(i),'.bmp'];
    Zc3Din(:,:,i-imod) = imread(filename);

%% STEP 4.2: SPATIAL COORDINATES
% Convert the row x colum x page positions to X x Y x Z positions giving a
% total of 4 3D matrices where every row,colum,page position gives a
% single value of x,y,z,color of the desired pixel

    % call to function for scaling based on pixel density
    [Xin(:,:,i-imod),Yin(:,:,i-imod)]=ScaleImage(Zc3Din(:,:,i-imod),DP);
    Zin(:,:,i-imod) = ones(size(Xin(:,:,i-imod)))*((i-imod-1)*(1/DP));

    % DELETE data that is not needed useing the CLEAR VAR function
    % (doing the clear within the loop as a series of analyses reduced the
    % amount of data needed to be maintained in memory
end

%% STEP 4.3: Determine coordinates defining edges of the scan
% Capture the coordinate defining the physical edges of the scan (i.e., min
% and max x,y,z, which define the shape of the volume scanned in XRayCT,
% which includes the entire core and some surrounding empty sapce).
edge.all.x = [min(min(Xin)) max(max(Xin));]; % edge.x = [minx maxx];
edge.all.y = [min(min(Yin)) max(max(Yin));];
edge.all.z = [0 DPz*nf];
if isempty(dx)==1
    edge.scan.x = edge.all.x;
    edge.scan.y = edge.all.y;
    edge.scan.z = edge.all.z;
else
    edge.scan.x = [Xt-dx Xt+dx];
    edge.scan.y = [Yt-dy Yt+dy];
    edge.scan.z = edge.all.z;
end

%% VISUALIZATION
if Plot_open_bitmaps == 'Y';
    %% Plotting of Data in Physical coordinates:
    figure
    [mx,my] = size(Xin(:,:,1));

    for i=1:imax-imod
        dummyx = reshape(Xin(:,:,i),mx*my,1);
        dummyy = reshape(Yin(:,:,i),mx*my,1);
        dummyz = reshape(Zin(:,:,i),mx*my,1);
        dummyzc = reshape(Zc3Din(:,:,i),mx*my,1);
        scatter3(dummyx,dummyy,dummyz, ...

```

```

            3*ones(size(dummyz)),dummyzc,'filled')
            hold on
        end

        xlabel('Xin'); ylabel('Yin'),zlabel('Zin')
        box on, colorbar
        colormap bone
        %axis equal
    end

    %Defining a center point in the square bitmap:
    %center x point:
    xmid = max(max(Xin(:,:,1)))/2; %center x point
    ymid = max(max(Yin(:,:,1)))/2; %center y point

    % Defines a box based on ds above
    Isub = ind2sub(size(Xin),...
        find(Xin>=(xmid-0.5*ds) & Xin<=(xmid+0.5*ds) & ...
        Yin >= (ymid-0.5*ds) & Yin<= (ymid+0.5*ds)));

    xsub = Xin(Isub); %
    ysub = Yin(Isub);
    zsub = Zin(Isub);
    csub = Zc3Din(Isub);

    %% STEP 4.4: VISUALIZATION
    if Plot_open_bitmapsSUB == 'Y';
        figure
        [mx,my] = size(xsub(:,:,1));
        if Plot_open_bitmapsSUBCAT == 'Y';
            % reshape 2D matrices to vectors so that the variables are
            % compatible with teh limitations ofthe "scatter3" plotting
            % function
            scatter3(xsub,ysub,zsub,3*ones(size(zsub)),csub,'filled')
        else % Plot as images
            for i=1:imax-imod % same as 1:nf
                % gives the number of elements in a sub-area of a single bitmap
                Iz = find(zsub==((i-1)*(1/DP)));
                % number of elements within the sub area
                nz = numel(Iz);
                % assumes have same number of elements in the x,y directions
                nzs = sqrt(nz);
                dummyx = reshape(xsub(Iz),nzs,nzs);
                dummyy = reshape(ysub(Iz),nzs,nzs);
                dummyz = reshape(zsub(Iz),nzs,nzs);
                dummyzc= reshape(csub(Iz),nzs,nzs);
                surf(dummyx,dummyy,dummyz,dummyzc)
                hold on
                shading flat
            end
        end
        axis equal
        colorbar
        xlabel('X-Position (cm)'),...
            ylabel('Y-Position (cm)'),zlabel('Z-Position (cm)')
        box on
        colorbar
        colormap bone
        title 'Volume of Chosen Sub-region (open space)'
    end

    %% STEP 4.6: Calculating porosity %

```

```

Popen = zeros(nf,1);
for i = 1:nf
    % Count number of pixels within the sub-area of each bitmap image
    % (i.e., page of teh 3D matrix)
    % note, if I/nf is not an integer, will get an error in subsequent
    % calculations
    sub_area_pixels = numel(Isub)/nf;
    % Count number of pixels that are pores
    % Ipores = numel(find(csub(:,:,i)==0));% or write as
    Iz = find(zsub==((i-1)*(1/DP)));
    Ipores = sum(csub(Iz)==0);
    % Calculate porosity in each bitmap (i.e., page of the 3D matrix)
    % to give a per bitmap transect of porosity
    Popen(i) = Ipores/sub_area_pixels;
end
zp = (1:nf)*(1/DP); % z-position of the porosity measurement
Popen_ds_smooth = smooth(zp,Popen,ds,'moving'); % moving window mean
if Plot_Open_Transect == 'Y';
figure
    h1 = plot(Popen,zp,'k.');
    hold on
    h2 = plot(Popen_ds_smooth,zp,'-r','LineWidth',1.5);
    xlabel('Open Porosity')
    ylabel('Z-position [cm]')
    box on, grid on
    h3 = plot([mean(Popen) mean(Popen)],[0 nf/DP],...
        'r--','LineWidth',1.5);
    legend([h1 h2 h3],'2D porosity of each scan',...
        'Moving cube: 2*ds',...
        'Mean of Volume')
    title 'Transect for Open Porosity'
end

%%%%%%%%%%%%%%%
%% Step 5: Combining variables to obtain desired porosity type %%
%%%%%%%%%%%%%%%
%
% Open and Healed Porosity Calcluations:
% - Thresholding of the XRayCT scans into binary images has the following
% hierarchical scheme
%   (1) Open Porosity == lowest density
%   (2) Lesser porosity == density image contains both the open porosity
%       and healing minerals less dense than the matrix
%       --> for GEO-N2 this largely corresponds to calcite
%   (3) Denser porosity == density image contains minerals more dense than
%       the matrix
%       --> for GEO-N2 this largely corresponds to Fe-Oxids, Pyrite, silica
%
% Therefore:
% - Open Porosity corresponds to (1)
% - Healed porosity (including crack fill and replacement corresponds to:
%   [(2)-(1)] + (3)

%% STEP 5.1: Define Key variables
% Popen = Popen;
Phealed = Pgr+(Pls-Popen);
Pskel = Popen+Phealed;
Phealed_ds_smooth = smooth(zp,Phealed,ds,'moving'); % moving window mean
Pskel_ds_smooth = smooth(zp,Pskel,ds,'moving'); % moving window mean
Popen_ds_smooth = smooth(zp,Popen,ds,'moving'); % moving window mean

%% STEP 5.2: Master CT Figure

```

```

if Plot_ALL_POROSITY_TRANSECT == 'Y';
figure
h1 = plot(Phealed_ds_smooth,zp,'-g','LineWidth',1.5);
hold on
h2 = plot([mean(Phealed) mean(Phealed)],[0 nf/DP],...
    'g--','LineWidth',1.5);
h3 = plot(Popen_ds_smooth,zp,'-b','LineWidth',1.5);
h4 = plot([mean(Popen) mean(Popen)],[0 nf/DP],'b--','LineWidth',1.5);
h5 = plot(Pskel_ds_smooth,zp,'-r','LineWidth',1.5);
h6 = plot([mean(Pskel) mean(Pskel)],[0 nf/DP],'r--','LineWidth',1.5);

box on, grid on
xlabel('Porosity')
ylabel('Z-position [cm]')
title 'Total Porosity Transect'

legend([h1 h2 h3 h4 h5 h6],'Healed moving cube: 2*ds',...
    'Healed Mean of Volume',...
    'Open moving cube: ds',...
    'Open Mean of Volume',...
    'Skeletal moving cube: ds',...
    'Skeletal Mean of Volume')
end

%%%%%%%%%%%%%
%% STEP 6: Constructing More Relevant Plots
%%

if Plot_HEALED_POROSITY_TRANSECT == 'Y';
figure
h1 = plot(Phealed,zp,'k.');
hold on
h2 = plot(Phealed_ds_smooth,zp,'-g','LineWidth',1.5);
xlabel('Healed Porosity')
ylabel('Z-position [cm]')
box on, grid on
h3 = plot([mean(Phealed) mean(Phealed)],[0 nf/DP],...
    'g--','LineWidth',1.5);
legend([h1 h2 h3],'2D porosity (healed) of each scan',...
    'Moving cube: ds',...
    'Mean Healed of Volume')
title('Healed Porosity Transect')
end

if Plot_OPEN_POROSITY_TRANSECT == 'Y';
figure
h1 = plot(Popen,zp,'k.');
hold on
h2 = plot(Popen_ds_smooth,zp,'-b','LineWidth',1.5);
xlabel('Open Porosity')
ylabel('Z-position [cm]')
box on, grid on
h3 = plot([mean(Popen) mean(Popen)],[0 nf/DP],'b--','LineWidth',1.5);
legend([h1 h2 h3],'2D porosity (open) of each scan',...
    'Moving cube: ds',...
    'Mean Open of Volume')
title('Open Porosity Transect')
end

if Plot_SKEL_POROSITY_TRANSECT == 'Y';
figure
h1 = plot(Pskel,zp,'k.');

```

```

hold on
h2 = plot(Pskel_ds_smooth,zp,'-r','LineWidth',1.5);
xlabel('Skeletal Porosity')
ylabel('Z-position [cm]')
box on, grid on
h3 = plot([mean(Pskel) mean(Pskel)],[0 nf/DP],'r--','LineWidth',1.5);
legend([h1 h2 h3], '2D porosity (skeletal) of each scan',...
    'Moving cube: ds',...
    'Mean Skeletal of Volume')
title('Skeletal Porosity Transect')
end

%%%%%%%%%%%%% STEP 7: DEFINING VARIABLES AND Saving to .mat - PLACE HOLDER: %%%
%%%%%%%%%%%%% STEP 8: REDEFINING VARIABLES TO BE USED IN A FUNCTION & Saving to .mat%
% Define file name and name of structured array for export to .mat file
[pathstr, fname, ext] = fileparts(filename);

CT_Open_Avg = mean(Popen);
CT_Open_Trans = Popen_ds_smooth;
CT_Healed_Avg = mean(Phealed);
CT_Healed_Trans = Phealed_ds_smooth;
CT_Skel_Avg = mean(Pskel);
CT_Skel_Trans = Pskel_ds_smooth;
CT_Pixel_Size = 1/DP;
CT_Cube_Side_Size = ds;
CT_DP = DP;
CT_Z_Posit = zp;
CT_Open_everypage = Popen;
CT_Healed_everypage = Phealed;
CT_Skel_everypage = Pskel;

if Data_export == 'Y'
    % Save selected variables from workspace to current working direction
    % Modifiable to save specific variables from workspace
    save([pathstr,'GEON2_',fname,'.mat'],['GEON2_',fname],'-mat');
end

```

Micro CT Function (1 of 2): Function to Analyze X-Ray CT Binary Porosity Data

```
function [out,hf,ht] = ...
PorosityXRayCTAnalysisV2(fname,DP,range,dz,ds, ...
Plotting,T_sensitivity,Data_export,Figure_export)

% TITLE: MASTER_CT_DRAFT.m
%
%
% AUTHOR: Justin Roth/Nick Davatzes
% DATE: last modified 05/05/13
%
% PURPOSE: This script will import and analyse binary bitmaps that
% represent 1) healed porosity with a greater density than the surrounding
% rock, and 3) open porosity within host rock, 2) healed porosity with less
% density than the surrounding host the host rock.
%
%
% NOTES: CTan was used to apply thresholding of the voxels.
% Variables can be created, saved, and added together to easily plot
% healed, open, and skeletal porosity.
%
%
% INPUTS
% fname represents the file path and "base" file name shared by three
% sets of bitmap files:
% (1) bitmaps of open pores ('_Open')
% (2) bitmaps of material with greater density than the matrix
% ('_Denser')
% (3) bitmaps of material with less density than the matrix (and
% which also includes the open pores ('_Less')
% Within each set, but maps are sequentially numbered
--> The most likely problems running this script are due to
either:
% (A) incorrect application of the file naming convention
% (B) insufficient memory availability and saturation of swap
space
% DP = bitmap resolution in dots per cm
% range = [imin imax] is the range of bitmaps to use in the analysis
% ds = window size in cm
% dz = step distance along transect; not fully implemented and
% currently taken at 1 per bitmap
% Plotting = 'Y' or 'N': create plots of the analysis
% T_sensitivity = 'Y' or 'N': run sensitivity to window size analysis
% Data_export = 'Y' or 'N': write a structure array of the variables to
% a d.mat file in which the file is named as:
% ['GEON2_',FNAME,'.mat']
% and in which the variables have the name:
% 'GEON2_',FNAME
% Figure_export = 'Y' or 'N': print PDF files of figures if generated
--> Not implemented in this version
%
% OUTPUTS
% out is a structured array of the porosity analysis
%
% hf contains the handles to the figure axes
% ht contains the handl to the QA figure axes
%
%
% EXAMPLE SYNTAX and SAMPLE ANALYSIS:
```

```

%
%
%
% NOTES:
% - Be sure all bitmaps are exactly the same size (i.e., have the same
% number of rows and columns of pixels) and the 1,1 pixel is registered to
% equivalent x,y coordinate positions. If not, two problems are likely:
% 1. The images might be slightly shuffled so that pixels in adjacent
% layers are mismatched
% 2. Filtering of the data (i.e., sub-sampling) for the transect based on
% a spatial position (rather than a row x column position) might
% occasionally lead to different numbers of pixels within adjacent layers
% being chosen, thus leading to a concatenation problem in assembling the
% 3D matrix, Xfin,Yfin,Zfin.

% To Do: Eventually, each subtask should be written as a function called
% from a MASTER file.

%%%%%%%%%%%%%%%
%% STEP 0: I/O Check: %%
if nargin < 2
    error('Image Resolution in dots per cm (DP) unspecified')
end
if nargin < 3
    imin = 18;      %Not yet trimmed
    imax = 2082;    %Not yet trimmed
else
    imin = range(1);
    imax = range(2);
end

%%%%%%%%%%%%%%%
%% STEP 1: Preliminary Selections and Variables: %%
%%%%%%%%%%%%%%%

% STEP 1.1: Define Optional Tests
% Scanline parameters
% Optimal window size should consider the following factors
% 1. Property attempting to characterize
% 2. Grain size: probably not meaningful unless window size >= 10 X grain
%    size
% 3. Pore size and distribution
% 4. Pixel resolution: want at least 10 X pixel resolution sized box
% 5. Run Time --> practical
% 6. If comparing to other data: e.g., x-ray CT, it should be conducted
%    a consistency sample volume across both analyses
if nargin <4
    ds = 20/DP; % defines box size (cm) and is about 0.05 cm
    % Scan Window
    % ds = .2 %defines box size (cm)
    % ds = 5/DP % equates to a 0.0013 cm window
    % ds = 20/DP; % defines box size (cm) and is about 0.05 cm
    % ds = sqrt(700)/DP %equivalent to number of points used in point count
elseif nargin <5
    dy = 0.002; % window step increment in cm, 0,.01 preferred?
                % Should be less than ds --> dy = ds/5;
end
% - Plot Results?
if exist('Plotting') == 0

```

```

    Plotting = 'N';
end
% - Test Transect Sensitivity
if exist('T_sensitivity') == 0
    T_sensitivity = 'N'; % 'Y' = yes, 'N' = no
end
% - Write WORKSPACE Variables to File
if exist('Data_export') == 0
    Data_export = 'N';
end
% - Export Transect Figure as .fig file
if exist('Figure_export') == 0
    Figure_export = 'N';
end
if Plotting == 'N' && Figure_export == 'Y';
    error(['Figures are not generated and cannot be exported: ...
            'Set PLOTTING to ''Y'' or FIGURE_EXPORT to ''N'''']);
end

% Split up File Path, File Name, and Extension from the input fname
%   variable
[pathstr, fname, ext] = fileparts(fname);

% Resolution as dots per cm;
% DP = 374; % Typical resolution of MicroXRayCt at max settings is 374

%% -----%%
%% STEP 1.2: Set resolution
DPz= DP; % set as part of INPUT to FUNCTION above

% Pre-Condition Spatial Transformation Variables
Zc3Din = [];
Xin =[]; Yin =[]; Zin=[];
Xfin=[]; Yfin=[]; Zfin=[];

% Pre-Condition Filtering Variables - PLACEHOLDER
% Spatial Filtering - PLACEHOLDER

% Transect position:
Xt = 2.54/2; % x-coordinate of the z-axis parallel transect [cm]
Yt = 2.54/2; % y-coordinate of the z-axis parallel transect [cm]

% Data analyzed within given distances to transect position:
% Note: These do not need correspond to the size of the transect window,
% which is given as ds below
% --> creating an automatic scaling to the transect window and less than
% the dimensions of the core scanned would be desirable in the long term
dx = 2*ds; % ds is the half-length of sample window used to query data
            % in the XRayCT scan in X and Y only [cm]
dy = dx;    % square window

%% -----%%
%% STEP 1.3: SELECTING THE PLOTS AND ANALYSES TO OUTPUT
% "GR" for 'greater density bitmaps'
% "LS" for 'less-density bitmaps'
% "OP" for 'open porosity bitmaps'

% Summary Plots
Plot_SUMMARY_POROSITY_TRANSECT = 'Y';
Plot_ALL_DERIVATIVE_TRANSECTS = 'N';

% QA Plots
% Plot the untrimmed bitmaps (greater density than host)

```

```

    Plot_bitmaps = 'N';
% Plot the selected trimmed volume of interest (greater density than host)
    Plot_bitmapsSUB = 'N';
% Plot selected trimmed volume of interest (greater density than host)
% as a surf plot (faster)
    Plot_bitmapsSUBSCATTER = 'N';
% Plot a transect of information about the selected trimmed volume of
% interest (greater density than host)
    Plot_Transect = 'N';
% Export data (See also "out" and "Data_export")
    Export_BitmapSub = 'N';

%% SAMPLES Summary Table of Bitmaps and Trimming Parameters to mitigate
% edge effects
%
%% -----SAMPLE N2-4302FA (18-1557)-----%%
% SampleBaseNameGR = 'N24302FA_TrimBin_Denser'; %"GR" for 'greater density'
% SampleBaseNameLS = 'N24302FA_TrimBin_Less'; %"LS" for 'less-density'
% SampleBaseNameOP = 'N24302FA_TrimBin_Open'; %"OP" for 'open porosity'
%
% imin = 179; %arbitrary desired minumum for the trimmed dataset
% imax = 1295; %arbitrary desired maximum for the trimmed dataset
%
%% -----SAMPLE N2-3937F (18-2082)-----%%
% SampleBaseNameGR = 'N23937F_TrimBin_Denser'; % "GR" for 'greater density'
% SampleBaseNameLS = 'N23937F_TrimBin_Less'; % "LS" for 'less-density'
% SampleBaseNameOP = 'N23937F_TrimBin_Open'; % "OP" for 'open porosity'
%
% imin = 18; %Not yet trimmed
% imax = 2082; %Not yet trimmed
%
%% -----SAMPLE N2-4306F (18-2088)-----%%
% SampleBaseNameGR = 'N24306F_TrimBin_Denser'; %"GR" for 'greater density'
% SampleBaseNameLS = 'N24306F_TrimBin_Less'; %"LS" for 'less-density'
% SampleBaseNameOP = 'N24306F_TrimBin_Open'; %"OP" for 'open porosity'
%
% imin = 18; %Not yet trimmed
% imax = 2088; %Not yet trimmed
%
%% -----SAMPLE N2-4306F (18-2088)-----%%
% SampleBaseNameGR = 'N24306F_TrimBin_Denser'; %"GR" for 'greater density'
% SampleBaseNameLS = 'N24306F_TrimBin_Less'; %"LS" for 'less-density'
% SampleBaseNameOP = 'N24306F_TrimBin_Open'; %"OP" for 'open porosity'
%
% imin = 18; %Not yet trimmed
% imax = 2088; %Not yet trimmed

%%%%%%%%%%%%%%%
%% STEP 2: Set up loop to analyze three End Member Densities      %%
%%%%%%%%%%%%%%%

%% 2.1 Parameters for number of bit map files to read
imod = imin-1;
nf = imax - imod; % number of files to read and analyze
suffix = {'_Denser'; '_Less'; '_Open'};

if nf <= 30
    c = [1:1:nf+1];
else
    c = [1:50:nf+1];
end

```

```

for k = 1:length(suffix); % Loop across three image files

% NOTE: Currently loads all bitmaps at once --> most memory intensive
fname_full = [fname,suffix{k}];
i = [imin:1:imax];
P = zeros(nf,1); % initialize variable in which porosity data is stored

for j = 1:nf;
    %% STEP 2.1: DATA IMPORT
    if i(j) == imin
        tstart = tic;
    end

    % Pad with zeros to preserve number of digits representing count
    if i(j)<=9 % ALTERNATE: could also use: dummy = num2str(n), nn=numel(n)
        % then directly consider the number of digits defining i
        pad = '000';
    elseif i(j)<=99
        pad = '00';
    elseif i(j) <=999
        pad = '0';
    else
        pad = '';
    end
    filename = [pathstr,'/',fname_full,pad,num2str(i(j)),'.bmp'];

    %% STEP 2.2: CONVERT TIFF/BITMAP IMAGE COORDINATES TO SPATIAL
    % COORDINATES
    % Convert the row x colum x page positions to X x Y x Z positions
    % giving a total of 4 3D matrices where every row,colum,page position
    % gives a single value of x,y,z,color of the desired pixel
    if Plot_bitmaps == 'N' &&...
        Plot_bitmapsSUB == 'N' && ...
        Plot_bitmapsSUBSCATTER == 'N' &&...
        Export_BitmapSub == 'N';
        % Matrix of color values
        Zc3Din = imread(filename);
        % Call to function for scaling based on pixel density
        [Xin,Yin]=ScaleImage(Zc3Din,DP);
        Zin = ones(size(Xin))*((j-1)*(1/DP));
    else
        Zc3Din(:,:,j) = imread(filename);
        % Call to function for scaling x and y coordinates of pixel colors
        % based on pixel density
        [Xin(:,:,j),Yin(:,:,j)]=ScaleImage(Zc3Din(:,:,j),DP);
        % Determining z coordinate of bitmap plane
        Zin(:,:,j) = ones(size(Xin(:,:,j)))*((j-1)*(1/DP));
    end

    %% STEP 2.3: Determine coordinates defining edges of the scan
    % Capture the coordinate defining the physical edges of the scan (i.e., min
    % and max x,y,z, which define the shape of the volume scanned in XRayCT,
    % which includes the entire core and some surrounding empty sapce).
    edge.all.x = [min(min(Xin)) max(max(Xin));]; % edge.x = [minx maxx];f
    edge.all.y = [min(min(Yin)) max(max(Yin));];
    edge.all.z = [0 DPz*nf];
    if isempty(dx)==1
        edge.scan.x = edge.all.x;
        edge.scan.y = edge.all.y;
        edge.scan.z = edge.all.z;
    else
        edge.scan.x = [Xt-dx Xt+dx];

```

```

    edge.scan.y = [Yt-dy Yt+dy];
    edge.scan.z = edge.all.z;
end

% Evaluating progress: Print progress to Command Line
if isempty(find(i(j) == (c+imod))) == 0
    % print filename to screen to keep track of progress
    str = ['Finished file: ', num2str(k), ': ', num2str(i(j)), ...
        ' of ', num2str(imax), ' files'];
    disp(str);
    if i(j) == imin
        ttotal = toc(tstart);
    else
        ttotal = toc(tstart)-ttotal;
    end
    str = ['      Import Elapsed Time = ', num2str(ttotal), ' [sec]'];
    disp(str);
end
if i == imax
    % print to screen status update: import process complete
    str = {[ 'All ', num2str(nf+1), ...
        ' Coordinate Calculations complete.' ]; [' ']};
    fprintf('%s\n', str{:});
end

%% STEP 2.5a: SCANLINE ATTRIBUTES
% Strategy:
%% Query scanline volumes to find desired indices (FIND) within range x,y
% x,y footprint of "box" (assume dx=dy) should match choice for 2D thin
% section analysis so they can be compared. Z could be single layer
% (e.g., just like a thinsection) or larger, such as equal to dx (the
% half-length of the box's edge)
%
% Since all bmp images are aligned, the Indices within the x and y ranges
% will be the same for all bitmap images. Thus we only need to perform the
% querry once to find the indices, and then subsample via FOR loop all the
% pages within the 3D matrices.

% Box Size: (full width, not half width): Defined as input above
% ds = 20/DP; % defines box halflength size (cm) and is about 0.05 cm

% Defining a center point in the square bitmap; scan along centerpoint
% Center x point:
xmid = max(max(Xin(:,:,1)))/2; %center x point
ymid = max(max(Yin(:,:,1)))/2; %center y point

% NOTE: find works only on 2D matrices
%     I = find (Xin>=(xmid-0.5*ds) & Xin<=(xmid+0.5*ds) & ...
%             Yin >= (ymid-0.5*ds) & Yin<= (ymid+0.5*ds));
% Defines a box based on ds above for 3d matrices, we use ind2sub
% (linear indices I, or row x colum x page as [Im,In,Ip])
Isub = ind2sub(size(Xin), find(Xin>=(xmid-0.5*ds) ...
    & Xin<=(xmid+0.5*ds) ...
    & Yin >= (ymid-0.5*ds) & Yin<= (ymid+0.5*ds)));
xsub = Xin(Isub); %
ysub = Yin(Isub);
zsub = Zin(Isub);
csub = Zc3Din(Isub); %

%% STEP 2.6: Calculating porosity
% P = zeros(nf,1);

```

```

% Count number of pixels within the sub-area of each bitmap image
% (i.e., page of teh 3D matrix)
[m,n,o] = size(Isub);
sub_area_pixels = m*n; % sub_area_pixels = numel(Isub)/nf;
% NOTE: if I/nf is not an integer, will get an error in subsequent
% calculations
% Count number of pixels that are pores
% Ipores = numel(find(csub(:,:,i)==0));% or write as
Iz = find(zsub==((j-1)*(1/DP)));
Ipores = sum(csub(Iz)==0);
% Calculate porosity in each bitmap (i.e., page of the 3D matrix)
% to give a per bitmap transect of porosity
P(j) = Ipores/sub_area_pixels;
end

zp = (1:nf)*(1/DP); % z-position of the porosity measurement
P_ds_smooth = smooth(zp,P,ds,'moving'); % moving window mean

%% Recording Porosity for later use
if k == 1;
    Pgr = P;
    Pgr_ds_smooth = P_ds_smooth;
elseif k == 2
    Pls = P;
    Pls_ds_smooth = P_ds_smooth;
else
    Popen = P;
    Popen_ds_smooth = P_ds_smooth;
end

% Clear or reset unnecessary variables
clear P P_ds_smooth sub_area_pixels Xin Yin Zin Cin
Zc3Din = [];

if Plot_bitmaps == 'N' &&...
    Plot_bitmapsSUB == 'N' && ...
    Plot_bitmapsSUBSCATTER == 'N' ...
    Export_BitmapSub == 'N';

    clear Iz Ipores mx my xsub ysub zsub csub
else
    DATA.xsub = xsub; clear xsub;
    DATA.ysub = ysub; clear ysub;
    DATA.zsub = zsub; clear zsub;
    DATA.csub = csub; clear csub;
    DATA.sub_area_pixels = sub_area_pixels; clear sub_area_pixels;
    DATA.Ipores = Ipores; clear Ipores;
    DATA.P = P; clear P;
    DATA.zp = zp; clear zp;
    DATA.P_ds_smooth = P_ds_smooth; clear P_ds_smooth;

    clear Iz Ipores mx my xsub ysub zsub csub
end
end

%%%%%%%%%%%%%%%
%% Step 3: Combining variables to obtain desired porosity type %%%
%%%%%%%%%%%%%%%
%
% Open and Healed Porosity Calcluations:
% - Thresholding of the XRayCT scans into binary images has the following
% hierarchical scheme

```

```

% (1) Open Porosity == lowest density
% (2) Lesser porosity == density image contains both the open porosity
%     and healing minerals less dense than the matrix
%     --> for GEO-N2 this largely corresponds to calcite
% (3) Denser porosity == density image contains minerals more dense than
%     the matrix
%     --> for GEO-N2 this largely corresponds to Fe-Oxids, Pyrite, silica
%
% Therefore:
% - Open Porosity corresponds to (1)
% - Healed porosity (including crack fill and replacement corresponds to:
%   [(2)-(1)] + (3)

%% STEP 3.1: Define Key variables
span = ds*DP;
% Popen = Popen;
Phealed = Pgr+(Pls-Popen);
Pskel = Popen+Phealed;
Phealed_ds_smooth = smooth(zp,Phealed,ds,'moving');% moving window mean
Popen_ds_smooth = smooth(zp,Popen,ds,'moving'); % moving window mean
Pskel_ds_smooth = Phealed_ds_smooth + Popen_ds_smooth;

%%%%%%%%%%%%%
%% STEP 4: REDEFINING VARIABLES TO BE USED IN A FUNCTION & Saving to .mat %
%%%%%%%%%%%%%
% Define file name and name of structured array for export to .mat file

% eval(['GEON2_',fname,'.CT_Open_Avg = mean(Popen)'']);
% eval(['GEON2_',fname,'.CT_Open_Trans = Popen_ds_smooth''']);
% eval(['GEON2_',fname,'.CT_Healed_Avg = mean(Phealed)'']);
% eval(['GEON2_',fname,'.CT_Healed_Trans = Phealed_ds_smooth''']);
% eval(['GEON2_',fname,'.CT_Skel_Avg = mean(Pskel)'']);
% eval(['GEON2_',fname,'.CT_Skel_Trans = Pskel_ds_smooth''']);
% eval(['GEON2_',fname,'.CT_Pixel_Size = 1/DP''']);
% eval(['GEON2_',fname,'.CT_Cube_Size = ds''']);
% eval(['GEON2_',fname,'.CT_DP = DP''']);
% eval(['GEON2_',fname,'.CT_Z_Posit = zp''']);
% eval(['GEON2_',fname,'.CT_Open_everypage = Popen''']);
% eval(['GEON2_',fname,'.CT_Healed_everypage = Phealed''']);
% eval(['GEON2_',fname,'.CT_Skel_everypage = Pskel''']);

eval(['DATA_XCT_',fname,'.CT_Open_Avg = mean(Popen);']);
eval(['DATA_XCT_',fname,'.CT_Open_Trans = Popen_ds_smooth;']);
eval(['DATA_XCT_',fname,'.CT_Healed_Avg = mean(Phealed);']);
eval(['DATA_XCT_',fname,'.CT_Healed_Trans = Phealed_ds_smooth;']);
eval(['DATA_XCT_',fname,'.CT_Skel_Avg = mean(Pskel);']);
eval(['DATA_XCT_',fname,'.CT_Skel_Trans = Pskel_ds_smooth;']);
eval(['DATA_XCT_',fname,'.CT_Pixel_Size = 1/DP;']);
eval(['DATA_XCT_',fname,'.CT_Cube_Size = ds;']);
eval(['DATA_XCT_',fname,'.CT_DP = DP;']);
eval(['DATA_XCT_',fname,'.CT_Z_Posit = zp;']);
eval(['DATA_XCT_',fname,'.CT_Open_everypage = Popen;']);
eval(['DATA_XCT_',fname,'.CT_Healed_everypage = Phealed;']);
eval(['DATA_XCT_',fname,'.CT_Skel_everypage = Pskel;']);
eval(['DATA_XCT_',fname,'.zp = zp;']);
eval(['DATA_XCT_',fname,'.ds = ds;']);
eval(['DATA_XCT_',fname,'.dx = dx;']);
eval(['DATA_XCT_',fname,'.dy = dy;']);
eval(['DATA_XCT_',fname,'.DP = DP;']);
eval(['DATA_XCT_',fname,'.DPz= DPz;']);

% eval(['out = GEON2_',fname]);
eval(['out = ','DATA_XCT_',fname,';']);

```

```

if Data_export == 'Y'
    % Save selected variables from workspace to current working direction
    % Modifiable to save specific variables from workspace
    save([pathstr,'/','DATA_XCT_',fname,'.mat'],...
        ['DATA_XCT_',fname],'-mat');
end

%%%%%%%%%%%%%%%
%% STEP 3: VISUALIZATION
%%%%%%%%%%%%%%%
if Plotting == 'Y';

    %% STEP 2.4: VISUALIZATION for QA: Very Slow
    % Entire imported set of bitmaps
    if Plot_bitmaps == 'Y';
        %% Plotting of Data in Physical coordinates:
        figure
        [mx,my] = size(Xin(:,:,1));
        for ii=1:imax-imod
            dummyx = reshape(Xin(:,:,ii),mx*my,1);
            dummyy = reshape(Yin(:,:,ii),mx*my,1);
            dummyz = reshape(Zin(:,:,ii),mx*my,1);
            dummyzc = reshape(Zc3Din(:,:,ii),mx*my,1);
            scatter3(dummyx,dummyy,dummyz, ...
                3*ones(size(dummyz)),dummyzc,'filled')
            hold on
        end
        xlabel('Xin'); ylabel('Yin'), zlabel('Zin')
        box on, colorbar
        colormap bone
        %axis equal
    end

    %% STEP 2.5b: VISUALIZATION for QA: VERY SLOW!!!
    % Sub region of the set of imported bitmaps
    if Plot_bitmapsSUB == 'Y';
        figure
        [mx,my] = size(xsub(:,:,1));
        if Plot_bitmapsSUBSCATTER == 'Y';
            % reshape 2D matrices to vectors so that the variables are
            % compatible with teh limitations ofthe "scatter3" plotting
            % function
            scatter3(xsub,ysub,zsub,3*ones(size(zsub)),csub,'filled')
        else % Plot as images
            for ii=1:imax-imod % same as 1:nf

                Iz = find(zsub==((ii-1)*(1/DP))); % Iz gives the number of
                                                % elements in a sub-area for a
                                                % single bitmap (2D)
                nz = numel(Iz); % number of elements within the sub area
                nzs = sqrt(nz); % assumes have same number of elements in the
                                % x and y directions
                dummyx = reshape(xsub(Iz),nzs,nzs);
                dummyy = reshape(ysub(Iz),nzs,nzs);
                dummyz = reshape(zsub(Iz),nzs,nzs);
                dummyzc= reshape(csub(Iz),nzs,nzs);
                surf(dummyx,dummyy,dummyz,dummyzc)
                hold on
                shading flat
            end
        end
        box on, axis equal
    end
end

```

```

colorbar
 xlabel('X-Position (cm)'); ylabel('Y-Position (cm)');
 zlabel('Z-Position (cm)')
 box on, colorbar
 colormap bone
 title('Volume of Chosen Sub-region (greater dense thresh)')
end

% 2.6 VISUALIZATION: QA on the density-based plots
% Within Transect Window Slice-by-slice porosity and volume
% smoothed curve
Pnames = {'Pgr';'Pls';'Popen'}; % 'Pskeletal','Phealed','Popen'
Psnames= {'Pgr_ds_smooth';'Pls_ds_smooth';'Popen_ds_smooth'};

if Plot_Transect == 'Y';
 titlestr = ...
 {[ 'Transect for Healed Minerals with Greater Density',...
 'than Host Rock'][;...
 {[ 'Transect for Healed Minerals with Lesser Density',...
 'than Host Rock'][;...
 {[ 'Transect for Healed Minerals with Zero Density (Open)' ]};

nc = 3;
Nf = figure;
[hf,ax] = GenerateTransectAxesColumns(Nf,nc);
for ii = 1:nc
 axes(ax(ii))
 P = eval(Pnames{ii});
 P_ds_smooth = eval(Psnames{ii});

 h1 = plot(P,zp,'k.');
 hold on
 h2 = plot(P_ds_smooth,zp,'-r','LineWidth',1.5);
 xlabel('Healed Porosity (Greater Density Than Host)')
 ylabel('Z-position [cm]')
 box on, grid on
 h3 = plot([mean(P) mean(P)], [0 nf/DP], 'r--','LineWidth',1.5);
 legend([h1 h2 h3], '2D porosity of each scan',...
 'Moving cube: ds',...
 'Mean of Volume')
 title(titlestr{ii})
end
end

%% 2.7 Side-by-Side Plot of All Derivative Transects
% Within Transect Window Slice-by-slice porosity and volume
% smoothed curve
if Plot_ALL_DERIVATIVE_TRANSECTS == 'Y'
 nc = 4;
Nf = figure;
[hf,ax] = GenerateTransectAxesColumns(Nf,nc);
titlestr = {'Healed Porosity Transect',...
 'Open Porosity Transect',...
 'Skeletal Porosity Transect',...
 'All'};
P_str = {'Phealed';'Pskel';'Popen'};
Ps_str= {'Phealed_ds_smooth';'Pskel_ds_smooth';'Popen_ds_smooth'};

for ii = 1:nc
 P = P_str{ii};
 Ps = Ps_str{ii};
 axes(ax(ii));

```

```

h1 = plot(P,zp,'k.');
hold on
h2 = plot(Ps,zp,'-g','LineWidth',1.5);
if ii == 1
    xlabel('Fractional Porosity []')
    ylabel('Z-position [cm]')
end
mP = mean(p);
h3 = plot([mP mP],[0 nf/DP],'g--','LineWidth',1.5);

box on, grid on
title(titlestr{i})
if ii == 3
    legend([h1 h2 h3],'2D porosity (skeletal) of each scan',...
        'Moving cube: ds',...
        'Mean Skeletal of Volume')
end
if ii == 4
    axes(ax(1));
    MyLineProp = {'LineWidth',1.5};
    h1 = plot(Phealed_ds_smooth,zp,'-g','LineWidth',1.5);
    hold on
    h2 = plot([mean(Phealed) mean(Phealed)],[0 nf/DP],...
        'g--',MyLineProp{:});
    h3 = plot(Popen_ds_smooth,zp,',...
        '-b','LineWidth',1.5);
    h4 = plot([mean(Popen) mean(Popen)],[0 nf/DP],...
        'b--',MyLineProp{:});
    h5 = plot(Pskel_ds_smooth,zp,',...
        '-r',MyLineProp{:});
    h6 = plot([mean(Pskel) mean(Pskel)],[0 nf/DP],...
        'r--',MyLineProp{:});

    box on, grid on
    title(titlestr{ii})
    legend([h1 h2 h3 h4 h5 h6],'Healed moving cube: 2*ds',...
        'Healed Mean of Volume',...
        'Open moving cube: ds',...
        'Open Mean of Volume',...
        'Skeletal moving cube: ds',...
        'Skeletal Mean of Volume')
end
end
end

%% 2.8 Summary Transect Plot
if Plot_SUMMARY_POROSITY_TRANSECT == 'Y';
figure
h1 = plot(Phealed_ds_smooth,zp,'-g','LineWidth',1.5);
hold on
h2 = plot([mean(Phealed) mean(Phealed)],[0 nf/DP],'g--',MyLineProp{:});
h3 = plot(Popen_ds_smooth,zp,'-b',MyLineProp{:});
h4 = plot([mean(Popen) mean(Popen)],[0 nf/DP],'b--',MyLineProp{:});
h5 = plot(Pskel_ds_smooth,zp,'-r','LineWidth',1.5);
h6 = plot([mean(Pskel) mean(Pskel)],[0 nf/DP],'r--',MyLineProp{:});

box on, grid on
xlabel('Porosity')
ylabel('Z-position [cm]')
title 'Total Porosity Transect'

legend([h1 h2 h3 h4 h5 h6],'Healed moving cube: 2*ds',...
    'Healed Mean of Volume',...

```

```
'Open moving cube: ds',...
'Open Mean of Volume',...
'Skeletal moving cube: ds',...
'Skeletal Mean of Volume')
end

%% Skeletal Bitmamp-based 3D image
% --> Currently no skeletal bitmap is built up

%-----PLACEHOLDER-----
```

end

Micro CT Function (2 of 2): Function to Analyze X-Ray CT Binary Porosity Data Along Multiple Scan-Lines to Assess Statistics of Porosity Structure

```
function [out] = ...
PorosityXRayCTMultiScanline(fname,DP,ds,dy,diameter,range,Plotting)

% AUTHOR: NCD
% CREATED: 2013-05-07
% VERSION: v1
%
% PURPOSE:
%   Generate and visualize multiple scanlines for the analysis of the
%   statistical variation in porosity as per the FUNCTION:
%   PorosityXRayCTAnalysisV2
%
% INPUTS
%   fname   = base name of bitmap files (can include path to subdirectory
%             containing the files
%   DP      = resolution of bitmaps in dots/cm
%   ds      = window edge length
%   dy      = 0.5 % spacing between transects in cm
%             if dy is:
%               scalar - specifies the spacing between transects
%               'Single' - requests a single transect at the center of the
%                           bitmap
%               [x,y] - specifies a location for a single transect
%               [x1,y1;x2,y2;...] - is a column vector of x and y
%                           coordinates explicitly specifying the positions of
%                           transects
%
%   diameter of sample for use in constructing transects
%   range   = [imin imax]; min and max indices of bitmaps comprising the CT
%             scan
%   Plotting= 'Y' creates summary plots including
%             (1) plots the open and healed transect data including
%                 boxplots summarizing the variation in porosities by
%                 bitmap and by smoothed transect corresponding to the
%                 "moving" average represented by a cube with sides ds
%                 moving through the sample at a spacing of 1/2*ds
%             (2) location of the transects within a wireframe of the
%                 outline of the bitmaps
%
% OUTPUTS
%   out     is a structured array containing the following variables:
%   out.xy = trans_xy;
%   out.Z  = Z; clear Z
%   out.Po = Po; clear Po
%   out.Ph = Pg + (Pl - out.Po); clear Pg Pl
%   out.Pos= Pos; clear Pos % smoothed data
%   out.Phs= Pgs + (Pls - out.Pos);
%   out.ds = ds;
%   out.DP = DP;
%   out.nf = nf;
%   out.N  = N; % number of pixels within each window; sqrt(N) is the
%             % number of pixles in the row or column directions, which
%             % also correspond to the x and y directions in the local
%             % coordinate system
%   out.edge=edge; clear edge
%
%   out.Poall = Poall; clear Poall
%   out.Phall = Pgall + (Plall - out.Poall); clear Pgall Plall
```

```

%     out.Nall  = Nall; clear Nall
%
% SYNTAX
%     fname = ['DATA_XCT/N23937F_TrimBin'];
%     DP    = 374.5; % bitmap resolution in points/cm; assumed same in x,y,z
%                 % for a 26.7 micron pixel
%     ds    = 0.05; % window edge length
%     dy    = 0.8; %0.5; % space between transects in the x and y direction
%     diameter= 1.5; % sample diameter in cm
%     % range = [18 2082]; % range of bitmaps to read: nearly all, but will
%                           % show edge affects in statistics
%     range = [58 1570]; % minimizes edge affects in statistics
%     % range = [18 200]; % range of bitmaps to read: short to run fast
%     Plotting= 'Y';
%     [out] = ...
%         PorosityXRayCTMultiScanline(fname,DP,ds,dy,diameter,range,Plotting)
%
% NOTE: Resolution in vertical and horizontal:
%       DP and DP assumed to be the same

%% STEP 1.0: I/O
if exist('DP','var') == 0
    DP = 374 % max resolution of Temple Micro-XRayCT
end
if exist('dy','var') == 0
    dy = 10*ds % max resolution of Temple Micro-XRayCT
end
if exist('range','var') == 0
    imin = 18
    imax = 2082
else
    imin = range(1);
    imax = range(2);
end
if exist('Plotting') == 0
    Plot = 'N'
end
% File base name
[pathstr,fnamestr,ext] = fileparts(fname);

%%%%%%%%%%%%%%%
%% 2.0 Derive Scanline Data
%% STEP 2.1: Define Series of x,y coordinates for axis parallel transects
[ndy] = numel(dy);
if ndy > 1
    if dy == 'Single'
        trans_xy = [diameter/2, diameter/2];
    elseif isstr(dy) == 0 && ischar == 0
        % Specify either a [x,y] position for a transect or a set of x,y
        % pairs for transects
        trans_xy = dy;
    else
        error(['dy is incorrectly specified please choose: ...
            'dy = scalar, dy = ''Single'', dy = [x,y], ...
            'dy = [x1,y1;x2,y2;...]''])
    end
end
trans_xy = TransectSampler(diameter,ds,dy);
[m,n]      = size(trans_xy);

```

```

if isempty(trans_xy)
    error(['No transect [x,y] coordinates generated for dy = ',...
        num2str(dy)])
end

%% STEP 2.2: Calculate Porosity

% Determine number of measurements
imod = imin-1;
nf = imax - imod; % number of files to read and analyze
i = [imin:1:imax];

% Precondition variables to accept porosity transect data from each
% transect:
% --> Matrices have rows corresponding to porosity measured at Z and
%     columns corresponding to the transection number (rows of "trans_xy")
Pg = nan(nf,m); % greater density
Pl = Pg; % lower density
Po = Pg; % open porosity

Pgs = Pg; Pls = Pg; Pos = Pg;
Pgall = nan(nf,1); Plall = Pgall; Poall = Pgall;

% Naming convention to distinguish three porosity images used
suffix = {'_Denser'; '_Less'; '_Open'};

for i = 1:m
    tstart = tic;
    for j = 1:3 % number of binary images used to determine porosity
        % Set fname
        F = [pathstr,'/',fnamestr,suffix{j}];
        % Run the porosity transect function for each of the
        [P,Z,Ps,N,edge,Pall,Nall] = ...
            PorosityXRayCTtransect(F,DP,trans_xy(i,:),ds,imin,imax);
        if j == 1
            Pg(:,i) = P;
            Pgs(:,i) = Ps;
            Pgall = Pall;
        elseif j == 2
            Pl(:,i) = P;
            Pls(:,i) = Ps;
            Plall = Pall;
        else
            Po(:,i) = P;
            Pos(:,i) = Ps;
            Poall = Pall;
        end
    end
    tpast = toc(tstart);
    disp(['--> Transect ',num2str(i), ' of ', num2str(m), ...
        ' complete after ', num2str(tpast), ' seconds'])
end

%% STEP 2.3: Calculate Open, Healed, Skeletal Porosity for output & export
out.xy = trans_xy;
out.Z = Z; clear Z
out.Po = Po; clear Po
out.Ph = Pg + (Pl - out.Po); clear Pg Pl
out.Pos= Pos; clear Pos % smoothed data
out.Phss= Pgs + (Pls - out.Pos);
out.ds = ds;
out.DP = DP;

```

```

out.nf = nf;
out.N = N;% number of pixels within each window; sqrt(N) is the number
        % of pixels in the row or column directions, which also
        % correspond to the x and y directions in the local
        % coordinate system
out.edge = edge; clear edge

out.Poall = Poall; clear Poall
out.Phall = Pgall + (Plall - out.Poall); clear Pgall Plall
out.Nall = Nall; clear Nall
out.fnamestr = fnamestr;

%% STEP 4.0: Export
save([pathstr,'/','XCTmulti_',fnamestr,'.mat'],['out'],'-mat')

%%%%%%%%%%%%%
%% 3.0 PLOT of RESULTS
%%%%%%%%%%%%%
% 3.1: Basic plots of porosity transects and statistical analysis of
%       variability
if Plotting == 'Y'
    figure
    [m,n] = size(out.xy);
    xx = [1:1:m];
    subplot(1,4,1)
    % for i = 1:m
    %     plot(out.Po(:,i),out.Z,'co','MarkerSize',2);
    %     hold on
    %     plot(out.Pos(:,i),out.Z,'-', 'LineWidth',1.5,'Color',[0 0.6 0]);
    %     plot(out.Ph(:,i),out.Z,'o','MarkerSize',2,'Color',[0 0.9 0]);
    %     plot(out.Phs(:,i),out.Z,'-', 'LineWidth',1.5,'Color',[0 0.7 0]);
    % end
    MyColor = jet(2*m);
    plot(out.Po,out.Z,'o','MarkerSize',2);
    hold on
    plot(out.Pos,out.Z,'-', 'LineWidth',2);
    xlabel({'Fractional Porosity';['ds = ',num2str(out.ds),' [cm]']})
    ylabel('z-position [cm]')
    title({'Open PorosityL:';fnamestr})
    colormap(jet)
    plot(out.Poall,out.Z,'k--','LineWidth',1.5)
    xlim([0 1]); box on; grid on;
    subplot(1,4,2)
    plot(out.Ph,out.Z,'v','MarkerSize',2);
    hold on
    plot(out.Phs,out.Z,'-', 'LineWidth',2);
    xlabel({'Fractional Porosity';['ds = ',num2str(out.ds),' [cm]']})
    ylabel('z-position [cm]')
    title({'Healed Porosity:';fnamestr})
    colormap(jet)
    plot(out.Phall,out.Z,'k--','LineWidth',1.5)
    xlim([0 1]); box on; grid on;

    MyFormatBox = ...
        {'notch','on','colorgroup',[],'outliersize',3,'symbol','k+'};
    MyFormatLine = ...
        {'d-k','MarkerSize',10,'MarkerFaceColor','y','LineWidth',1.5};
    % Statistical Analysis
    subplot(2,4,3)
    boxplot(out.Ph,MyFormatBox{:})
    ylim([0 1])
    ylabel('Fractional Porosity []')

```

```

title('Healed - Per Bitmap')
hold on
plot(xx,mean(out.Ph),MyFormatLine{::})
plot(xx,mean(out.Phall)*ones(size(xx)), 'k--', 'LineWidth',1.5)
subplot(2,4,4)
boxplot(out.Phs,MyFormatBox{::})
hold on
ylim([0 1])
title('Healed - Smoothed')
plot(xx,mean(out.Phs),MyFormatLine{::})
plot(xx,mean(out.Phall)*ones(size(xx)), 'k--', 'LineWidth',1.5)
subplot(2,4,7)
boxplot(out.Po,MyFormatBox{::})
hold on
ylabel('Fractional Porosity []')
ylim([0 1])
title('Open - Per Bitmap')
xlabel('Transect')
plot(xx,mean(out.Po),MyFormatLine{::})
plot(xx,mean(out.Poall)*ones(size(xx)), 'k--', 'LineWidth',1.5)
subplot(2,4,8)
boxplot(out.Pos,MyFormatBox{::})
hold on
ylim([0 1])
title('Open - Smoothed')
xlabel('Transect')
plot(xx,mean(out.Pos),MyFormatLine{::})
plot(xx,mean(out.Poall)*ones(size(xx)), 'k--', 'LineWidth',1.5)

% 3.2 TRANSECT Position QA
figure
dummy = ones(size(out.Z));
xx = nan(out.nf,m);
yy = xx;
zz = xx;
% build matrices
for i = 1:m
    xx(:,i) = dummy*out.xy(i,1);
    yy(:,i) = dummy*out.xy(i,2);
    zz(:,i) = out.Z;
end
plot3(xx,yy,zz,'-o','LineWidth',2,'MarkerSize',3);
hold on

% Plot outline corresponding to extent of bitmaps in x,y,z
llb = [out.edge.xmin,out.edge.ymin,out.edge.zmin];%low-left-bottom
lrb = [out.edge.xmax,out.edge.ymin,out.edge.zmin];
urb = [out.edge.xmax,out.edge.ymax,out.edge.zmin];
ulb = [out.edge.xmin,out.edge.ymax,out.edge.zmin];
llt = [out.edge.xmin,out.edge.ymin,out.edge.zmax];
lrt = [out.edge.xmax,out.edge.ymin,out.edge.zmax];
urt = [out.edge.xmax,out.edge.ymax,out.edge.zmax];
ult = [out.edge.xmin,out.edge.ymax,out.edge.zmax];
% Build outline of 3D box from sequency of corners
B = [llb;lrb;urb;ulb;llb;...
      llt;lrt;lrb;lrt;...
      urt;urb;urt;...
      ult;ulb;ult;...
      llt];
plot3(B(:,1),B(:,2),B(:,3),'om--','MarkerSize',3);

% Plot windows around one transect
dss = 0.5*out.ds;

```

```

    xx = out.xy(1,1);
    yy = out.xy(1,2);
    llb = [xx-dss, yy-dss];
    lrb = [xx+dss, yy-dss];
    urb = [xx+dss, yy+dss];
    ulb = [xx-dss, yy+dss];
    zzb = max(out.Z);
    Bwin= [llb,zzb;lrb,zzb;urb,zzb;ulb,zzb;llb,zzb];

    plot3(Bwin(:,1),Bwin(:,2),Bwin(:,3), 'k-', 'LineWidth',2);

    xlabel('x [cm]')
    ylabel('y [cm]')
    zlabel('z [cm]')
    title('Transect Positions')

    box on, grid on, axis equal
    xspan = 0.5*(out.edge.xmax - out.edge.xmin);
    xlim([out.edge.xmin - xspan, out.edge.xmax + xspan])
    ylim([out.edge.ymin - xspan, out.edge.ymax + xspan])
    zlim([out.edge.zmin - xspan, out.edge.zmax + xspan])
end

end % end of primary function

%%%%%%%%%%%%%%%
%% 4.0 SUBFUNCTIONS
%%%%%%%%%%%%%%%
%-----%
%% 4.1 SUBFUNCTION: Transect position sampler
function [xy,ds,dy] = TransectSampler(diameter,ds,dy)

% 1. Assume that [x,y] = [0,0] is diameter/2 from the center
xc = diameter/2; yc = xc;

% 2. Find corners of square inscribed within circular cross section of
%     sample
s = diameter*sin(pi/4); % length of the side of the inscribed square
xll = xc - 0.5*s;
xur = xll + s;
yll = yc - 0.5*s;
yur = yll + s;

% 3. Have transects overlap by 1/2 the window length
if isempty(dy) == 1
    dss = 0.5*ds;
else
    dss = dy;
end
xrange = [xll+0.5*ds:dss:xur-0.5*ds];
yrange = [yll+0.5*ds:dss:yur-0.5*ds];

% 4. Cross correlate to fully sample region at equal intervals (lots of
% transects!
[X,Y] = meshgrid(xrange,yrange);
% reshape into a simple vector of xy pairs
[m,n] = size(X);
x = reshape(X,m*n,1);
y = reshape(Y,m*n,1);
xy= [x,y];

```

```

end % end of subfunction

%-----
%% 4.2 SUBFUNCTION: Import data
function [P,Z,Ps,N,edge,Pall,Nall] = ...
    PorosityXRayCTtransect(fname,DP,xy,ds,imin,imax)
%
% PURPOSE
%   Collect porosity measurements along a single transect traversing a set
%   of binary bitmap images
%   NOTE: 3D imaging of the pore structure or pore shape attributes is not
%         attempted
%
% INPUTS
%   fname file base name with path
%   xy  coordinates of axis-parallel transect --> [x,y]
%   ds  window edge length
%   dy  smoothing increment
%
%
% OUTPUTS
%   p   fraction of pixels in each slice
%   ps  smoothed fraction of pixels for given smoothing window
%   z   z-position of each p measurement
%   zs  z-position of each ps measurement (same as z)
%   xy  xy position of transect (same as input)
%

[pathstr,fnamestr,ext] = fileparts(fname);

% Parameters for number of bit map files to read
imod = imin-1;
nf = imax - imod; % number of files to read and analyze
i = [imin:1:imax];
P = nan(nf,1); % initialize variable in which porosity data is stored
Pall = P;
% Z = P;
Z = [1:1:nf]'*(1/DP);

for j = 1:nf;
    % pad with zeros to preserve number of digits representing count
    if i(j)<=9
        % ALTERNATIVE: could also use: dummy = num2str(n), nn=numel(n),
        % then directly consider the number of digits defining i
        pad = '000';
    elseif i(j)<=99
        pad = '00';
    elseif i(j) <=999
        pad = '0';
    else
        pad = '';
    end
    filename = [pathstr,'/',fnamestr,pad,num2str(i(j)), '.bmp'];
    M = imread(filename); % matrix of color values
    M = M./max(max(M)); % ensure normalization between 0 and 1

    Nall = numel(M);
    Pall(j) = sum(sum(M==0)/Nall);

    % Scale Image to physical coordinates
    % - Call to function for scaling of image to true physical dimensions
    % based on pixel density
    [X,Y]=ScaleImage(M,DP);

```

```

% Z(j) = ((j-1)*(1/DP));

% Find points within a square window aligned with x and y within 0.5*ds
% of the transect position
% Define a box based on ds above
Isub = ind2sub(size(X), find(X>=(xy(1)-0.5*ds) & X<=(xy(1)+0.5*ds) &...
    Y >= (xy(2)-0.5*ds) & Y<= (xy(2)+0.5*ds)));
xsub = X(Isub); %
ysub = Y(Isub);
csub = M(Isub); %
% determine number of points within window for a single bitmap
[m,n,o] = size(Isub);
% determine fraction of window consisting of pores
N = m*n;
P(j) = sum(csub == 0)/N; % pores are zeros in normalized binary image
end
% Find max and min dimensions of bitmap for export
edge.xmin = min(min(X));
edge.xmax = max(max(X));
edge.ymin = min(min(Y));
edge.ymax = max(max(Y));
edge.zmin = min(Z);
edge.zmax = max(Z);
% Smooth the data as per a 3D cube but through an average of the averages
span = floor(DP*ds);
% moving window mean one calculation per bitmap
Ps = smooth(Z,P,span,'moving');

end % end of subfuction
%-----

```

Supporting Scripts and Functions (1 of 8): Formatting Plotting Axes

```
function [h,ax] = GenerateTransectAxes(figureN)

% AUTHOR: Nicholas C. Davatzes
% DATE CREATED: 2012-11-14
% MATLAB VERSION: 2010a
%
%
% PURPOSE:
%   Script to explicitly define plot size and subplot regions with handles
%   - Creates print and PDF compatible axes for plotting the Brady's
%     pumping records.
%   - Dimensions are scaled to an 11 x 8.5 page so that the way the figure
%     looks on screen corresponds to its characteristics using the save as
%     PDF functionality
%   - The current version produces all axes at the start (as opposed to
%     SUBPLOT), use the command: axes(ax(i)), where i is the axis number from
%     left to right, preceding the PLOT function to plot to a specific axis
%
%
% INPUTS:
%   figureN = figure number
%
%
% OUTPUT
%   h = figure handle
%   ax = vector of axes handles, where n = the number of axes
%
%
% NOTES:
%   (1) 5 panels organized in columns
%   (2) 8.5 x 11 Landscape View
%
%--> With a series of for loops, this FUNCTION could be modified to
% automatically set up axes. The best combination of inputs would be:
%   PaperSize = [xSize ySize]
%   Margins   = [xmargin ymargin]
%   array      = [m n] % number of rows by columns of evenly sized axes
%   sep       = [xsep ysep] % separation between axes in x and y
%--> defaults could be set up using IF statements
%
%
%-----%
% % EXAMPLE
% % Size the Figure window to print
% set(gcf,'PaperUnits','centimeters')
% xSize = 8; ySize = 12;
% xLeft = (21-xSize)/2; yTop = (30-ySize)/2;
% set(gcf,'PaperPosition',[xLeft yTop xSize ySize])
% set(gcf,'Position',[X Y xSize*50 ySize*50])
%
% % Then to plot to a specific axis:
% axes(ax.ax1); % call axes handle for desired plot, if previously used
% %               % plotted to a different figure, call: figure(hf)
%
%%%%%%%%%%%%%
% Create figure window
if nargin == 1
```

```

    h = figure(FigureN);
else
    h = figure;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Size Figure Window to Print (paper) Dimensions

set(gcf,'PaperUnits','inches'); % Set paper units, gcf = get current figure
% Variables defining paper dimensions and margins
xSize = 11; ySize = 8.5; % Paper Dimensions
% All dimensions within the figure are expressed in normalized units within
% Matlab
% Define margins
xmargin = 1/xSize; % 1 inch on x-dir, normalized
ymargin = 1/ySize; % 1 inch on y-dir, normalized

% set paper size; a PDF of the figure derived from this plot will have the
% papersize dimensions
set(gcf,'papersize',[xSize,ySize])

% Set position of the Figure on the Paper
xLeft = xmargin; yBottom = ymargin; % lower left corner
set(gcf,'PaperPosition',[xLeft yBottom (xSize-2*xmargin) (ySize-2*ymargin)]) % Coordinates to center figure on paper

% Position Figure on the Screen
scrsz = get(0,'ScreenSize'); % gives get screen size in pixels
dpi = get(0,'ScreenPixelsPerInch');
% X = scrsz(3)/2; Y = scrsz(4)/2;
X = 0; Y = 0;
if xSize <= dpi*scrsz(3) && ySize <= dpi*scrsz(4)
    s = dpi;
else
    s = 50;
end
% set figure position on screen; x and y correspond to the lower left hand
% corner
set(gcf,'Position',[X Y xSize*s ySize*s])

% The figure window has coordinates that range from 0 to 1 in the x- and
% y-direction. The lower left corner is (0,0), and the upper right corner
% is (1,1).

% % Axes can be set up with relative sizing or through absolute positions
% % Relative sizing of axis positions:
% x = 1.5/xSize; y = 2/ySize; width = 3/xSize; height = 2/ySize;

% Set up columns for plotting
n = 5; % number of columns
xsep = 0.25/xSize; % Normalized separation between columns in x-direction

% Dimension of axes for each column
width = (1 - ((n-1)*xsep + 2*xmargin))/n; % normalized figure units
height = 1 - 2*ymargin; % normalized figure units

% Position of lower left corner of each axes
ax = [];
for i = 1:n
    % axes('position',[xLowerLeft yLowerLeft Width Height]);
    x = (i-1)*(width + xsep)+xmargin;
    y = ymargin;
    dummy = axes('position',[x y width height]); %

```

```
ax = [ax dummy];

% If all data has the same y-axis then only the left-most y-axis is
% needed
if i ~= 1
    set(gca, 'YTickLabel', [])
end
box on, hold on
end
```

Supporting Scripts and Functions (2 of 8): Formatting Plotting Axes for Specified Number of Columns

```

function [h,ax] = GenerateTransectAxesColumns (FigureN,nc,xSize,ySize)

% [h,ax] = GenerateTransectAxesColumns (FigureN,nc,xSize,ySize)
%
% AUTHOR: Nicholas C. DAvatzes
% DATE CREATED: 2012-11-14
% MATLAB VERSION: 2010a
%
%
% PURPOSE:
%   Script to explicitly define plot size and subplot regions with handles
%   - Creates print and PDF compatible axes for plotting the Brady's
%     pumping records.
%   - Dimensions are scaled to an 11 x 8.5 page so that the way the figure
%     looks on screen corresponds to its characteristics using the save as
%     PDF functionality
%   - The current version produces all axes at the start (as opposed to
%     SUBPLOT), use the command: axes(ax(i)), where i is the axis number from
%     left to right, preceding the PLOT function to plot to a specific axis
%
%
% INPUTS:
%   FigureN = figure number
%   nc      = number of columns
%
% OUTPUT
%   h      = figure handle
%   ax    = vector of axes handles, where n = the number of axes
%
%
% NOTES:
%   (1) 5 panels organized in columns
%   (2) 8.5 x 11 Landscape View
%
%--> With a series of for loops, this FUNCTION could be modified to
% automatically set up axes. The best combination of inputs would be:
%   PaperSize = [xSize ySize]
%   Margins   = [xmargin ymargin]
%   array     = [m n] % number of rows by columns of evenly sized axes
%   sep       = [xsep ysep] % separation between axes in x and y
%--> defaults could be set up using IF statements
%
%
%-----%
% % EXAMPLE
% % Size the Figure window to print
% set(gcf,'PaperUnits','centimeters')
% xSize = 8; ySize = 12;
% xLeft = (21-xSize)/2; yTop = (30-ySize)/2;
% set(gcf,'PaperPosition',[xLeft yTop xSize ySize])
% set(gcf,'Position',[X Y xSize*50 ySize*50])
%
% % Then to plot to a specific axis:
% axes(ax.ax1); % call axes handle for desired plot, if previously used
%                 % plotted to a different figure, call: figure(hf)

%%%%%%%%%%%%%%%
%% Create figure window

```

```

if nargin <= 1
    h = figure;
else
    h = figure(FigureN);
end
if nargin < 2
    nc = 5;
end

%%%%%%%%%%%%%%%
%% Size Figure Window to Print (paper) Dimensions

set(gcf,'PaperUnits','inches'); % Set paper units, gcf = get current figure
% Variables defining paper dimensions and margins
if nargin <=2
    xSize = 11; ySize = 8.5; % Paper Dimensions
end

% Define margins
xmargin = 1/xSize; % 1 inch on x-dir, normalized
ymargin = 1/ySize; % 1 inch on y-dir, normalized

% set paper size; a PDF of the figure derived from this plot will have the
% papersize dimensions
set(gcf,'papersize',[xSize,ySize])

% Set position of the Figure on the Paper
xLeft = (xSize-xmargin)/2; yTop = (ySize-ymargin)/2;
set(gcf,'PaperPosition',[xLeft yTop (xSize-xmargin) (ySize-ymargin)]) %
Coordinates to center figure on paper

% Position Figure on the Screen
scrsz = get(0,'ScreenSize'); % gives get screen size in pixels
dpi = get(0,'ScreenPixelsPerInch');
% X = scrsz(3)/2; Y = scrsz(4)/2;
X = 0; Y = 0;
if xSize <= dpi*scrsz(3) && ySize <= dpi*scrsz(4)
    s = dpi;
else
    s = 50;
end
% set figure position on screen; x and y correspond to the lower left hand
% corner
set(gcf,'Position',[X Y xSize*s ySize*s])

% The figure window has coordinates that range from 0 to 1 in the x- and
% y-direction. The lower left corner is (0,0), and the upper right corner
% is (1,1).

% % Axes can be set up with relative sizing or through absolute positions
% % Relative sizing of axis positions:
% x = 1.5/xSize; y = 2/ySize; width = 3/xSize; height = 2/ySize;

% Set up columns for plotting
% nc = 5; % number of columns
xsep = 0.25/xSize; % Normalized separation between columns in x-direction

% Dimension of axes for each column
width = (1 - ((nc-1)*xsep + 2*xmargin))/nc; % normalized figure units
height = 1 - 2*ymargin; % normalized figure units

% Position of lower left corner of each axes
ax = [];

```

```
for i = 1:nc
    % axes('position',[xLowerLeft yLowerLeft Width Height]);
    x = (i-1)*(width + xsep)+xmargin;
    y = ymargin;
    dummy = axes('position',[x y width height]); %
    ax = [ax dummy];

    % If all data has the same y-axis then only the left-most y-axis is
    % needed
    if i ~= 1
        set(gca,'YTickLabel',[])
    end
    box on, hold on
end
```

Supporting Scripts and Functions (3 of 8)

Scale Image to Physical Dimensions from Pixel Size

```
function [X,Y] = ScaleImage(M,DP)

% AUTHOR:
% DATE CREATED:
% DATE MODIFIED:
%
%
% PURPOSE:
%
%
%
% INPUTS:
%   M = m by n matrix of color values as per a TIFF file,
%       where the number of columns is the number of pixels in the
%       x-direction
%       where the number of rows is the number of pixels in the y-direction
%   DP =dots per unit length, if unspecified, unit lenght is assumed to be
%       an inch
%   UNIT= dimensions of length used in conjunction with DP
%
% OUTPUTS:
%   X = m by n matrix of x positions the same size as M
%   Y = m by n matrix of y positions the same size as M
%
% --> for X and Y it is assumed that the pixels are of equal size and
% equally distributed along both dimensions
%
% EXAMPLE:
%
%
%
%
%
%
% RELATED FUNCTIONS/NOTES:
%   - M is assumed to be a monochromatic color matrix, i.e., there is only
%     a single value specifying color. However, note that if the image is RGB
%     or CMYK, the same X and Y matrices can be used to correspond to the
%     positions for each of the matrices corresponding to the additional
%     color scales since all are indexed to corrresponding matrix positions.

%% 1.0 I/O Check
% Number of Input Arguments
if nargin < 2
    error('insufficient number of input arguments')
end

%% 2.0 Calculate Scale Factor
%      (all steps separated to make explicit as an example of scripting)
s = size(M); % determine size of the matrix of color values
m = s(1); % number of rows of M
n = s(2); % number of columns of M
a = n/DP; % largest x-position; corresponds to:
            % [number of points]/([number points][length]^(-1)) = [length]
b = m/DP; % largest y-position

%% 3.0 Create the Matrices X and Y
```

```
x = linspace(0,a,n);
y = linspace(0,b,m);
% In the matrix M of color values:
%   - columns correspond to constant values of x positions in which the
%     minimum value of x is 0 and the maximum value of x is a;
%   - rows correspond to constant values of y positions so that 0 <= y <= b
[X,Y] = meshgrid(x,y);
```

Supporting Scripts and Functions (4 of 8): Compute Scanline Porosity in Moving Window

```
function [P] = ScanlineY(X,Y,M,x,dy,ds,Index)

% AUTHOR: Nicholas C. Davatzes
% DATE CREATED: 2012-11-01
% DATE MODIFIED:
% MABLAD VERSION: 2010a
%
% PURPOSE:
%   Find values within a moving window of specified size, with dimensions
%   of length, along a specified transect parallel to the y-axis.
%   - This script is primarily called from other scripts controlling the
%     image porosity analysis
%
% INPUTS:
%   X = m x n matrix of X positions as per meshgrid
%   Y = m x n matrix of Y positions as per meshgrid
%   M = m x n matrix of Z (color) values corresponding to X,Y and represens
%       a binary image data set
%   x = scalar spaciifying the x position of a scanline parallel to the
%       y-axis
%   dy= if scalar: incremental distance advanced between each box
%       if vector: taken to be y coordinates of box centers
%   ds= edge length of a square sample area
%       --> note: rather than using a number of pixels to specify the box
%       dimensions, this FUNCTION specifies a physical dimension to define
%       box size.
%   Index = determines whether or not the indices of the pixels associated
%           with each sampling window are collected into the structure array
%           'N' indicates do not collect and is the default, P.in still exists,
%           but as a cell array of empty matrices
%           'Y' implements collection of the indices
%   P.wn= Nnumber of pixels (elements) within each window
%
% NOTE: This algorithm assumes the dimensions of x, dy, ds are given in
% the same dimensions as the queried data set
%
% OUTPUTS:
%   P is a structured array
%   P.x =
%   P.y =
%   P.in= cell array of matrix indices of values for each window
%   P.m = cell array of matrix values within each window
%   P.p0= vector of calculated fractional porosity values for each window
%   P.p1= vector of calculated fractional matrix (non-pore space) values
%         for each window
%
% EXAMPLE:
%
%
% RELATED FUNCTIONS/NOTES:
%
%
%% 1.0 I/O Check
% Number of Input Arguments
if nargin < 5
    error('insufficient number of input arguments')
elseif nargin < 6
    ds = 0.5*ds;
```

```

    Index = 'N';
elseif nargin < 7
    Index = 'N';
end

% test matrices X,Y,M have same dimensions
% -----PLACEHOLDER-----
% test that M is a binary image of zeros and 1's
% -----PLACEHOLDER-----

M = M/max(max(M)); % normalizes the matrix of binary data, M,
% so that its values range from 0 to 1

%% 2.0 DEFINE VECTOR OF WINDOW CENTERS
if length(dy)==1;
    P.y = [min(min(Y))+ds:dy:max(max(Y))-ds];
else
    d = size(dy);
    if d(2)>d(1)
        P.y = dy';
    else
        P.y = dy;
    end
end
P.x = x*ones(size(P.y));

% Create place-holder vectors
P.in= cell(size(P.y));
P.m = P.in;
P.p = zeros(size(P.y))*nan;
P.wn= P.p; % number of pixels (elements) within each window

dss = 0.5*ds; % is the window half length

for i = 1:length(P.y)
    I = find(Y>=(P.y(i)-dss) & Y<=(P.y(i)+dss) & X>=(x-dss) & X<=(x+dss));
    if Index == 'Y'
        P.in{i}= I;
        P.m{i} = M(I); %redundant with P.in
    end
    P.wn(i)=numel(I);
    P.p1(i)= sum(sum(M(I)))/numel(I); % assumes the high value corresponds
% to the pore space
    P.p0(i)= 1-P.p1(i); % redundant as compliment to P.p1
end

```

Supporting Scripts and Functions (5 of 8): Scale image by Pixel Dimension

```
function [out,AX,H] = VisualizeMultiTransect(in,AX);

% function [out] = VisualizeMultiTransect(in
%
% AUTHOR: NCD
% CREATED: 2013-06-11
% VERSION: v1
%
% PURPOSE:
%   Provide a quick summary plot of multi-transect data to visualize
%   variation in porosity along different transects.
%   Data must be derived from the functions:
%     (1) PorosityImageMultiScanline.m
%     (2) PorosityXRayCTMultiScanline.m
%
% NOTE: This script does not magically force the alignment of the
% positions in the individual scanlines that correspond to crossing the
% fracture. That must be set up externally for instance via:
%   PlotRotatedScanlines
% And taking advantage of both the Rotational and Translational (ytrans,
% out. yapp) options.
%
% INPUTS
%   in      is the output structured array, OUT, from:
%           (1) PorosityXRayCTMultiScanline
%           (2) PorosityImageMultiScanline
%   in.Po    open porosity on a slice-by-slice basis
%   in.Pos   smoothed open porosity
%   in.Ph    healed porosity
%   in.Phs   smoothjed healed porosity
%   in.ds    window edge length
%
% OUTPUTS
%   figure   data plot
%   AX       Axes of plots
%   H        Handles of plots
%
% SYNTAX and EXAMPLE
%   figure
%   VisualizeMultiTransect(out)

%% 1.0 I/O

%% 2.0 Find Limits of Image Data
% Based on pixel size and pixel resolution
[m,n] = size(in.Pos);
out.Pos.Min = min(in.Pos)';
out.Pos.Max = max(in.Pos)';
out.Pos.Med = median(in.Pos)';
out.Pos.Avg = mean(in.Pos)';
out.Pos.std = std(in.Pos)';
out.Pos.Twentyfifth = zeros(m,1);
out.Pos.Seventyfifth = zeros(m,1);
for i = 1:m
    dummy = in.Pos(i,:)';
    I = find(dummy <= out.Pos.Med(i));
    J = find(dummy > out.Pos.Med(i));
    out.Pos.Twentyfifth(i) = median(dummy(I));
    out.Pos.Seventyfifth(i) = median(dummy(J));
```

```

    end

    out.Phs.Min = min(in.Phs)';
    out.Phs.Max = max(in.Phs)';
    out.Phs.Med = median(in.Phs)';
    out.Phs.Avg = mean(in.Pos)';
    out.Phs.std = std(in.Phs)';
    out.Phs.Twentyfifth = zeros(m,1);
    out.Phs.Seventyfifth = zeros(m,1);
    for i = 1:m
        dummy = in.Phs(i,:);
        I = find(dummy <= out.Phs.Med(i));
        J = find(dummy >= out.Phs.Med(i));
        out.Phs.Twentyfifth(i) = median(dummy(I));
        out.Phs.Seventyfifth(i) = median(dummy(J));
    end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% 3.0 PLOT DATA
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% 3.1 COMMON PLOT FORMATTING
%F.IndLines = {'.', 'LineWidth', 0.5, 'Color', [0.75 0.75 0.75]};
F.IndLines = {'-', 'LineWidth', 0.5, 'Color', [0.75 0.75 0.75]};
F.Min = {'-', 'LineWidth', 0.75, 'Color', [0 0 1]};
F.Max = {'-', 'LineWidth', 1, 'Color', [1 0 0]};
F.Med = {'-', 'LineWidth', 1.5, 'Color', [0 0 0]};

%% 3.2 Open Porosity: in.Z, in.Po
out.Z = in.Z;
Z = [out.Z; flipud(out.Z)];
% Make envelope of the based on extremes
% Xos= [out.Pos.Min; flipud(out.Pos.Max)];
% Xhs= [out.Phs.Min; flipud(out.Phs.Max)];

%% 3.3 Make envelop from the 25th and 75th percentiles of
% both Open and Healed porosity
Xos= [out.Pos.Twentyfifth; flipud(out.Pos.Seventyfifth)];
Xhs= [out.Phs.Twentyfifth; flipud(out.Phs.Seventyfifth)];
% Build envelope from mean +/- standard deviation
% Xos= [out.Pos.Avg+out.Pos.std; flipud(out.Pos.Avg-out.Pos.std)];
% Xhs= [out.Pos.Avg+out.Phs.std; flipud(out.Phs.Avg-out.Phs.std)];
subplot(1,3,1)
hf = patch(Xos,Z,'y');
set(hf,'EdgeColor','flat','LineWidth',1)
hold on
hl = plot(in.Pos,out.Z,F.IndLines{:});
hmin = plot(out.Pos.Min,out.Z,F.Min{:});
hmax = plot(out.Pos.Max,out.Z,F.Max{:});
hmed = plot(out.Pos.Med,out.Z,F.Med{:});
xlabel('Open Porosity')
ylabel('Z [cm]')
box on, grid on
legend([hf hl(1) hmin hmax hmed], '25th-75th %',...
    'Individual Scans',...
    'Min',...
    'Max',...
    'Median')

%% 3.4 Healed Porosity: in.Z, in.Ph
subplot(1,3,2)
patch(Xhs,Z,'y')

```

```

hold on
plot(in.Phs,out.Z,F.IndLines{:});
plot(out.Phs.Min,out.Z,F.Min{:})
plot(out.Phs.Max,out.Z,F.Max{:})
plot(out.Phs.Med,out.Z,F.Med{:})
xlabel('Healed Porosity')
ylabel('Z [cm]')
box on, grid on

%% Skeletal
%-----PLACEHOLDER-----
%%%%%
%%%%%
%%%%%
SUBFUNCTIONS
%%%%%
%%%%%

```

Supporting Scripts and Functions (6 of 8): Rotate Multi-Transect Data to Common, Fracture Relative Reference Frame

```

function [out] = PlotRotatedScanlines(in,theta,yint);

% [AX,out] = PlotRotatedScanlines(in,theta)
%
% AUTHOR: NCD
% CREATED: 2013-06-11
% VERSION: v1
%
% PURPOSE:
%   In most cases the fracture of interest makes a non-90 degree angle to
%   the axis (y-direction) of the thin section image. Thus in comparing the
%   porosity distribution along transects distributed in the x-direction,
%   the locust of perturbed porosity associated with the fracture is
%   offset. For simple fractures this can be partly accounted for by
%   rotating the data by the angle fracture makes the the x-axis of the
%   iamge. A further translation will be needed depending on the
%   y-intercept of the fracture in the image.
%
%
% INPUTS
%   in      is the output structured array, OUT, as created by:
%           (1) PorosityXRayCTMultiScanline
%           (2) PorosityImageMultiScanline
%   in.Po    open porosity on a slice-by-slice basis
%   in.Pos   smoothed open porosity
%   in.Ph    healed porosity
%   in.Phs   smoothjed healed porosity
%   in.ds    window edge half-length
%   theta   is the angle of rotation given in degrees
%
% OUTPUTS
%   figure   data plot
%   AX       Axes of plots
%   H        Handles of plots
%
% SYNTAX and EXAMPLE
%   figure
%   [out) = VisualizeMultiTransect(in,theta)
%
% NOTE:
%   Must be run in association with: (1) PorosityXRayCTMultiScanline;
%   (2) PorosityImageMultiScanline

%%%%%%%%%%%%%%%
%          FUNCTION          %%
%%%%%%%%%%%%%%%
%%%% 1.0 I/O

%% 2.0 Manually setable plotting options within script
Plot3DCheck = 'N'; % options: 'Y' or 'N'

%% 3.0 Rotate Data via SUBFUNCTION
out = VisualizeMultiTransectROTATE(in,theta,yint);

%% 4.0 Plotting
% 4.1 Basic Plot
figure

```

```

[m,n] = size(out.x);

% COMMON PLOT FORMATTING
xx = [1:1:n];
MyFormatBox = ...
    {'notch','on','colorgroup',[],'outliersize',3,'symbol','k+'};
MyFormatLine = ...
    {'d-k','MarkerSize',10,'MarkerFaceColor','y','LineWidth',1.5};

yy = out.yyp; % repmat(out.y,1,length(trans_x));
subplot(1,4,1)
    plot(out.Pos,out.y,'-', 'LineWidth',2);
    plot(out.Pos,yy,'-', 'LineWidth',2); box on
    hold on
    xlabel({'Fractional Porosity'; ['ds = ',num2str(out.ds), ' [cm]']})
    ylabel('z-position [cm]')
    title({'Open PorosityL:';out.fnamestr})
    colormap(jet)
    % plot(out.Poall,out.y,'k--','LineWidth',1.5)
    xlim([0 1]); box on; grid on;
subplot(1,4,2)
    plot(out.Phs,out.y,'-', 'LineWidth',2);
    plot(out.Phs,yy,'-', 'LineWidth',2); box on
    hold on
    xlabel({'Fractional Porosity'; ['ds = ',num2str(out.ds), ' [cm]']})
    ylabel('z-position [cm]')
    title({'Healed Porosity:';out.fnamestr})
    colormap(jet)
    % plot(out.Phall,out.Z,'k--','LineWidth',1.5)
    xlim([0 1]); box on; grid on;

    % Statistical Analysis
subplot(2,4,3)
    boxplot(out.Ph,MyFormatBox{::})
    ylim([0 1])
    ylabel('Fractional Porosity []')
    title('Healed - Per Bitmap')
    hold on
    plot(xx,mean(out.Ph),MyFormatLine{::})
    % plot(xx,mean(out.Phall)*ones(size(xx)),'k--','LineWidth',1.5)
subplot(2,4,4)
    boxplot(out.Phs,MyFormatBox{::}), box on
    hold on
    ylim([0 1])
    title('Healed - Smoothed')
    plot(xx,mean(out.Phs),MyFormatLine{::})
    % plot(xx,mean(out.Phall)*ones(size(xx)),'k--','LineWidth',1.5)
subplot(2,4,7)
    boxplot(out.Po,MyFormatBox{::})
    hold on
    ylabel('Fractional Porosity []')
    ylim([0 1])
    title('Open - Per Bitmap')
    xlabel('Transect')
    plot(xx,mean(out.Po),MyFormatLine{::})
    % plot(xx,mean(out.Poall)*ones(size(xx)),'k--','LineWidth',1.5)
subplot(2,4,8)
    boxplot(out.Pos,MyFormatBox{::}), box on
    hold on
    ylim([0 1])
    title('Open - Smoothed')
    xlabel('Transect')
    plot(xx,mean(out.Pos),MyFormatLine{::})

```

```

% plot(xx,mean(out.Poall)*ones(size(xx)),'k--','LineWidth',1.5)

% 4.2 Rotation check plot
figure, plot(out.xp, out.ypp,'.'), axis equal, grid on, box on
xlabel('xp'), ylabel('yp')
title('Rotated Scanlines: out.xp, out.ypp')

% 4.3 3D plot
if Plot3DCheck = 'Y'
    figure
    subplot(2,2,1)
        plot3(out.x,out.yy,out.Pos,'-'), box on, grid on, box on
        set(gca, 'DataAspectRatio',...
            [diff(get(gca, 'XLim')) diff(get(gca, 'XLim')) ...
            diff(get(gca, 'ZLim'))])
        xlabel('x'), ylabel('y'), zlabel('Open Porosity');

    subplot(2,2,2)
        plot3(out.xp,out.ypp,out.Pos,'-'), box on, grid on, box on
        set(gca, 'DataAspectRatio',...
            [diff(get(gca, 'XLim')) diff(get(gca, 'XLim')) ...
            diff(get(gca, 'ZLim'))])
        xlabel('xp'), ylabel('yp'), zlabel('Open Porosity');

    subplot(2,2,3)
        plot3(out.x,out.yy,out.Phs,'-'), box on, grid on, box on
        set(gca, 'DataAspectRatio',...
            [diff(get(gca, 'XLim')) diff(get(gca, 'XLim')) ...
            diff(get(gca, 'ZLim'))])
        xlabel('x'), ylabel('y'), zlabel('Healed Porosity');
    subplot(2,2,4)
        plot3(out.xp,out.ypp,out.Phs,'-'), box on, grid on, box on
        set(gca, 'DataAspectRatio',...
            [diff(get(gca, 'XLim')) diff(get(gca, 'XLim')) ...
            diff(get(gca, 'ZLim'))])
        xlabel('xp'), ylabel('yp'), zlabel('Healed Porosity');
    end
end

% 4.4 Translated alignment plot based on rotation angle and y-intercept
if exist('yint') == 1
    figure, plot(out.xp, out.yypp,'.'), axis equal, grid on, box on
    xlabel('xp'), ylabel('yp')
    title('Rotated Scanlines: out.xp, out.ypp')
end

%%%%%%%%%%%%%
%          SUBFUNCTIONS
%%%%%%%%%%%%%
function [out] = VisualizeMultiTransectROTATE(in,theta,yint)
%
% PURPOSE:
%   Rotate and translate porosity transects into the fracture local
%   reference frame so that xp is aligned with the fracture and ypp is
%   normal to the fracture
%   - If no ytrans is specified, no translation is calcualted or plotted
%
% INPUTS
%   in      as above
%   theta   as above
%   yint   as above
%
```

```

% OUTPUTS
%   in.xp
%   in.yyp
%   in.ytrans

%%%%%%%%%%%%%%%
% 1.0  I/O
if exist('yint') == 0
    yint = 0;
end

% 2.0 Collect basic outputs to maintain integrity of structured array
out = in;
out.theta = theta;

% 3.0 Convert to radians
theta = deg2rad(theta);

% 4.0 Define rotation matrix
rot = [cos(theta) -sin(theta); sin(theta) cos(theta)];

% 5.0 Apply rotation matrix and translation to each set of transect
%     positions recorded in:
%         out.data.x
%         out.data.yy
[m,n] = size(out.x);
out.xp = [];
out.yyp = [];
out.yypp= [];
out.ytrans = [];
for i = 1:n
    xpyp = [out.x(:,i),out.yy(:,i)]*rot';
    out.xp(:,i) = xpyp(:,1);

    % Calculat yyp - rotation only, suitable for plotting ontop of
    % thin section image
    out.yyp(:,i)= xpyp(:,2);
    % define translation
    ytrans_i = -(yint + out.x*tan(theta));
    ytrans = [ytrans;ytrans_i];
    % Calculate yypp: rotation + translation
    out.yypp(:,i)= xpyp(:,2) - ytrans_i;
end
end
%%%%%%%%%%%%%%

```

Supporting Scripts and Functions (7 of 8): Calculate Basic Frequency Distribution Statistics

```
function [out] = quartile(x);
%
%
% AUTHOR: Chris D. Larson
% DATE: 14 Sep 2004 (Updated 17 Sep 2004)
%
% PURPOSE:
%   Calculate basic statistics describing the frequency distribution of the
%   data set in x
%
% SOURCE:
%   http://www.mathworks.com/matlabcentral/fileexchange/
%           5877-quartile-percentile-calculation/content/quartile.m

Nx = length(x);

% compute mean
mx = mean(x);

% compute the standard deviation
sigma = std(x);

% compute the median
medianx = median(x);

% STEP 1 - rank the data
y = sort(x);

% compute 25th percentile (first quartile)
Q(1) = median(y(find(y<median(y))));

% compute 50th percentile (second quartile)
Q(2) = median(y);

% compute 75th percentile (third quartile)
Q(3) = median(y(find(y>median(y))));

% compute Interquartile Range (IQR)
IQR = Q(3)-Q(1);

% compute Semi Interquartile Deviation (SID)
% The importance and implication of the SID is that if you
% start with the median and go 1 SID unit above it
% and 1 SID unit below it, you should (normally)
% account for 50% of the data in the original data set
SID = IQR/2;

% determine extreme Q1 outliers (e.g., x < Q1 - 3*IQR)
iy = find(y<Q(1)-3*IQR);
if length(iy)>0,
    outliersQ1 = y(iy);
else
    outliersQ1 = [];
end

% determine extreme Q3 outliers (e.g., x > Q1 + 3*IQR)
iy = find(y>Q(1)+3*IQR);
if length(iy)>0,
    outliersQ3 = y(iy);
```

```

else
    outliersQ3 = [];;
end

% compute total number of outliers
Noutliers = length(outliersQ1)+length(outliersQ3);

% % display results
% disp(['Mean: ', num2str(mx)]);
% disp(['Standard Deviation: ', num2str(sigma)]);
% disp(['Median: ', num2str(medianx)]);
% disp(['25th Percentile: ', num2str(Q(1))]);
% disp(['50th Percentile: ', num2str(Q(2))]);
% disp(['75th Percentile: ', num2str(Q(3))]);
% disp(['Semi Interquartile Deviation: ', num2str(SID)]);
% disp(['Number of outliers: ', num2str(Noutliers)]);

%% Define outputs
out.N      = Nx;
out.sigma   = sigma;
out.Mean    = mx;
out.Median  = medianx;
out.Q1      = Q(1);
out.Q2      = Q(2);
out.Q3      = Q(3);
out.IQR     = IQR;
out.SID     = SID;

%
% Percentile Calculation Example
%
% define percent
kpercent = 75;
%
% STEP 1 - rank the data
y = sort(x);
%
% STEP 2 - find k% (k /100) of the sample size, n.
k = kpercent/100;
result = k*Nx;
%
% STEP 3 - if this is an integer, add 0.5.
% If it isn't an integer round up.
[N,D] = rat(k*Nx);
if isequal(D,1), % k*Nx is an integer, add 0.5
    result = result+0.5;
else % round up
    result = round(result);
end
%
% STEP 4 - Find the number in this position. If your depth ends
% in 0.5, then take the midpoint between the two numbers.
[T,R] = strtok(num2str(result),'0.5');
if strcmp(R,'.5'),
    Qk = mean(y(result-0.5:result+0.5));
else
    Qk = y(result);
end
%
% display result

```

```
% fprintf(1,['\nThe ',num2str(kpercent),...
%     'th percentile is ',num2str(Qk),'.\n\n']);
```

Supporting Scripts and Functions (8 of 8): Group Subplots Tightly

```
function ha = tight_subplot(Nh, Nw, gap, marg_h, marg_w)
%
% AUTHOR: by Pekka Kumpulainen
% DATE CREATED: 23 Jun 2010
%
% PURPOSE:
%   tight_subplot creates "subplot" axes with adjustable gaps and margins
%
% ha = tight_subplot(Nh, Nw, gap, marg_h, marg_w)
%
%   in: Nh      number of axes in height (vertical direction)
%       Nw      number of axes in width (horizontal direction)
%       gap     gaps between the axes in normalized units (0...1)
%               or [gap_h gap_w] for different gaps in height and width
%       marg_h  margins in height in normalized units (0...1)
%               or [lower upper] for different lower and upper margins
%       marg_w  margins in width in normalized units (0...1)
%               or [left right] for different left and right margins
%
%   out: ha      array of handles of the axes objects
%           starting from upper left corner, going row-wise as in
%           going row-wise as in
%
% Example: ha = tight_subplot(3,2,[.01 .03],[.1 .01],[.01 .01])
%           for ii = 1:6; axes(ha(ii)); plot(randn(10,ii)); end
%           set(ha(1:4),'XTickLabel',''); set(ha,'YTickLabel','')
%
% Pekka Kumpulainen 20.6.2010 @tut.fi
% Tampere University of Technology / Automation Science and Engineering
%
% SOURCE:
%   http://www.mathworks.com/matlabcentral/fileexchange/27991-tight-subplot

if nargin<3; gap = .02; end
if nargin<4 || isempty(marg_h); marg_h = .05; end
if nargin<5; marg_w = .05; end

if numel(gap)==1;
    gap = [gap gap];
end
if numel(marg_w)==1;
    marg_w = [marg_w marg_w];
end
if numel(marg_h)==1;
    marg_h = [marg_h marg_h];
end

axh = (1-sum(marg_h)-(Nh-1)*gap(1))/Nh;
axw = (1-sum(marg_w)-(Nw-1)*gap(2))/Nw;

py = 1-marg_h(2)-axh;

ha = zeros(Nh*Nw,1);
ii = 0;
for ih = 1:Nh
    px = marg_w(1);

    for ix = 1:Nw
```

```
ii = ii+1;
ha(ii) = axes('Units','normalized', ...
    'Position',[px py axw axh], ...
    'XTickLabel','','',...
    'YTickLabel','');
px = px+axw+gap(2);
end
py = py-axh-gap(1);
end
```

Script for Analysis of Surface Roughness (1 of 3): Analysis of Surface Roughness

```
%TITLE: RoughnessCalculations_v4.m
%
%
% Author: NC Davatzes and Justin Roth
% Date: last modified 4/25/13
%
% Purpose:
%   This script imports digitized fracture surfaces in order to calculate
%   and visualize the topographic relief.
%
% INPUTS:
%   Photograph of thin section from sample 3937FA
%   Digitized interpreted fracture surfaces from sample 3937FA
%
% OUTPUTS:
%   Plot of fracture surfaces colored by age
%   Plot of surface topography, statistics of surface length and topography
%   Power Spectral Analysis of surface topography
%
% CRITICAL CALLED FUNCTIONS:
%   QUARTILE
%   ScaleImage
%   ReorderByNearestNeighbor
%
% NOTES:
%
%%%%%
clear all, close all

%%%%%%%%%%%%%% SECTION 1: Import and Visualization %%%%%%
% Section 1.1: Visualizing the fracture in thin section:
%DDefining Variables:
DP = 484; %dots per cm
% Load the JPEG of the thin section image
TSi = imread('3937FAtsimage.jpg');
TScl= imread('CLAddedFlipped.tif');
%image(TSi), axis equal, axis tight

% Section 1.2: SCALE DATA
[Xtsi,Ytsi] = ScaleImage(TSi,DP);
[Xtscl,Ytscl] = ScaleImage(TScl,DP);

% Section 1.3: Plot scaled image
figure(1);
subplot(1,3,1)
    image(Xtsi(1,:),Ytsi(:,1),TSi); % TS image
    xlabel('X-position [cm]')
    ylabel('Y-position [cm]')
    title('Thin Section Image')
    hold on
    set(gca,'YDir','normal') % set axis properties within figures
    axis equal, axis tight, grid on
```

```

title('Thin Section (3523.5WFA)')

subplot(1,3,2)
image(Xtscl(1,:),Ytscl(:,1),TScl); % TS image
xlabel('X-position [cm]')
ylabel('Y-position [cm]')
title('Thin Section Image')
hold on
set(gca,'YDir','normal') % set axis properties within figures
axis equal, axis tight, grid on
title('Thin Section (3523.5WFA)')

%%%%%%%%%%%%%%%
%% SECTION 2: Visualizing the digitized fracture surfaces: %%
%%%%%%%%%%%%%%%

% SECTION 2.1: Load and Organize Data
% data = load('RoughRoughnessData.txt');
data = load('RoughRoughnessData_V3.txt');
% NOTE:
% Colum 1 = x
% Colum 2 = y
% Colum 3 = surface index pairs(color)
% Colum 4 = 1 or 2 corresponding to top or bottom
%
% For proper analysis and plotting, it is necessary to make the order of
% the points defining each surface approximately monotonic within each
% combined data set. The original data was not systematically digitized
% from one end of a fracture to another in order to take advantage of the
% clearest surfaces first, then infill.
% --> A goal here is to always start from one extreme end of the data set.
% OPTION 1:
% data = sortrows(data,[3,4]);
%
% OPTION 2:
% ReorderByNearestNeighbor
% this assumes that a search for the radially
% closest digitized point determines the order. This is only viable if
% the data is digitized at a spacing smaller than the distance between
% faces patches of the surface at bends.

X = data(:,1); %loads the master x-coordinate data
Y = data(:,2); %loads the master y-coordinate data
Si = data(:,3);
Stb= data(:,4);
mc = max(Si);
yoffst = 157*(1/DP);% difference between image and digitized data origin
MyColor = jet(2*mc);
% transform coordinate system due to wrong origin during digitization
realY=((Y-(max(Y))).*-1);
X = X.*(1/DP);
Y = realY.*(1/DP);

% SECTION 2.2: Divide data set into cells by surface number and top or
% bottom and make sure the data is a sequential order along the surface
% (path order and not necessarily monotonic in x)
MySurfaces.obs = cell(2*mc,1);
% upper surfaces are always the odd numbered elements, bottom surfaces are
% in the even numbered elements
for i = 1:mc
    for j = 1:2
        jj = [2,1]; % reverse order of 1 and 2 for logical test
        I = find(Si==i & Stb == jj(j));
    end
end

```

```

%           MySurfaces.obs{ (2*i)-(j-1),1} = [X(I),Y(I)];
x = X(I);
y = Y(I);
% AD Hoc...
% ***Attempt to get close to the lower left-had corner of the data
% to plot (note if the initial data point is actually in the first
% row of the x,y data matrix, comment this line out
A = [x,y]; A = sortrows(A,[2]); x = A(:,1); y = A(:,2);
if i == 6 && j == 1; % 12
    A = [x,y]; A = sortrows(A,[-2]); x = A(:,1); y = A(:,2);
    % x = [0.9462;x]; y = [1.772;y];
elseif i == 7 && j == 1; % 14
    A = [x,y]; A = sortrows(A,[-2]); x = A(:,1); y = A(:,2);
    % x = [0.01784;x]; y = [0.6053;y];
elseif i == 17 && j == 1; % 34
    A = [x,y]; A = sortrows(A,[-2]); x = A(:,1); y = A(:,2);
    % x = [1.111;x]; y = [1.433;y];
end
% [XI,YI] = ReorderByNearestNeighbor(x,y);
[XI,YI] = ReorderByNearestNeighbor2(x,y);
MySurfaces.obs{(2*i)-(j-1),1} = [XI,YI];
end
end

% SECTION 2.3: Plot surface traces
subplot(1,3,3)
dummy = [1;2];
for i = 1:mc-1;
    dummy = [dummy;1;2];
end
MyLine = {'-';'--'};
for i = 1:2*mc
    hold on
%     plot(MySurfaces.obs{i}(:,1),MySurfaces.obs{i}(:,2)+yoffst, ...
%         'LineStyle',MyLine{dummy(i)}, 'Color',MyColor(i,:));
    plot(MySurfaces.obs{i}(:,1),MySurfaces.obs{i}(:,2)+yoffst, ...
        'LineStyle',MyLine{dummy(i)}, 'Color',MyColor(i,:));
end

xlabel('X-position [cm]')
% ylabel('Y-position [cm]')
title('Digitized Fracture Surfaces')
axis equal, grid on, box on
xlim([min(min(Xtsi)) max(max(Xtsi))]);
ylim([min(min(Ytsi)) max(max(Ytsi))]);

% SECTION 2.4: Quality Assessment: Many results depend on the ordering of
% points defining the surface (they are path-dependent)
% --> Plot Test of the ordering of the points defining the surface
figure
subplot(1,2,1)
i = 14;% 12 % 14 % 34;
I = [1:1:length(MySurfaces.obs{i}(:,1))]';
x = MySurfaces.obs{i}(:,1);
y = MySurfaces.obs{i}(:,2)+yoffst;
plot3(x,y,I,'k-');
hold on
%     scatter(MySurfaces.obs{i}(:,1),MySurfaces.obs{i}(:,2)+yoffst, ...
%         ones(size(MySurfaces.obs{i}(:,1)))*4,I);
scatter3(x,y,I, ...
    ones(size(MySurfaces.obs{i}(:,1)))*8,I);
grid on, box on
view([0,90])

```

```

hc = colorbar;
set(get(hc,'ylabel'),'String','Path Sorted Index Position',...
    'FontSize',12,'FontName','Helvetica');

% Reorder by nearest neighbor (careful for "flat, "crack-like"
% protrusions from the surface
% [XI,YI] = ReorderByNearestNeighbor(x,y);
[XI,YI] = ReorderByNearestNeighbor2(x,y);
plot(XI,YI,'r-','LineWidth',1.5);
xlabel('X-component [cm]')
ylabel('Y-component [cm]')
title('Example surface path colored by row position')

subplot(1,2,2)
horder = nan(1,2*mc);
htitle = cell(1,2*mc);
hold on
for i = 1:2*mc
    horder(i) = plot(MySurfaces.obs{i}(:,2),'o-',...
        'Color',MyColor(i,:),'MarkerSize',5);
    htitle{i} = num2str(i);
end
xlabel('Index Position []')
ylabel('y-component [cm]')
legend(horder,htitle{::})
grid on, box on
title('Quality Assessment of Surface Path')

%%%%%%%%%%%%%%%
%% SECTION 3: Import and Visualization %%
%%%%%%%%%%%%%%

%% SECTION 3.1: Solve for Least Squares Best Fits to Each Surface
MySurfaces.model = cell(2*mc,3);
% structure of cell array:
%   column 1: [X,Y]
%   column 2: P (the constants in the polynomial fit)
%   column 3: S
% --> every row of the cell array corresponds to a different surface
% To query data:
%   1st: use curly brackets to grab the appropriate cell in the cell
%   array, e.g., MySurfaces{5,2} ==> cell at row 5, column 2
%   2nd: use normal brackets to query a position in the matrix
%   contained in the cell, e.g., MySurfaces{5,2}(1,2)
% ==> element of matrix at row 1, column 2 in the cell at row 5
% column 2

% APPROACH: Polyfit for each surface using the slope and intercept of the
% oldest surface to determine rotation.

% SECTION 3.1a: Fit each surface independently
N = 1; % polynomial fit degree
for i = 1:2*mc
    %   x = MySurfaces.obs{1}(:,1);
    %   y = MySurfaces.obs{1}(:,2);
    %   [x,I] = sortrows(x);
    %   y = y(I);
    %   [XI,YI] = ReorderByNearestNeighbor(x,y);
    %   [MySurfaces.model{i,2},MySurfaces.model{i,3}] =...
    %       polyfit(XI,YI,N);

    % Least Squares fit using polyfit
    [MySurfaces.model{i,2},MySurfaces.model{i,3}] =...

```

```

polyfit(MySurfaces.obs{i} (:,1),MySurfaces.obs{i} (:,2),N);

% ymodel = m * Xobs + b
ymodel = MySurfaces.model{i,2}(1)*MySurfaces.obs{i} (:,1) + ...
    MySurfaces.model{i,2}(2);
MySurfaces.model{i,1} = [MySurfaces.obs{i} (:,1),ymodel];
end

% SECTION 3.1b: Fit oldest surface to define reference frame and distribute
% into structure array (just so all arrays are the same shape even though
% it is redundant).
MySurfaces.old.model=cell(2*mc,1);

% Least Squares fit using polyfit
% Oldest surface first:
[P,S] = polyfit(MySurfaces.obs{1} (:,1),MySurfaces.obs{1} (:,2),N);

for i = 1:2*mc
    MySurfaces.old.model{i,2} = P;
    MySurfaces.old.model{i,3} = S;
    % ymodel = m * Xobs + b
    ymodel = MySurfaces.old.model{i,2}(1)*MySurfaces.obs{i} (:,1) + ...
        MySurfaces.old.model{i,2}(2);
    MySurfaces.old.model{i,1} = [MySurfaces.obs{i} (:,1),ymodel];
end

%-----%
%% 3.2.1: Rotate individual surface data set to slope = 0
MySurfaces.obsP = cell(2*mc,1);
for i = 1:2*mc
    t = -atan(MySurfaces.model{i,2}(1));
    ROT = [cos(t),sin(t);-sin(t),cos(t)];
    MySurfaces.obsP{i} = ROT'*MySurfaces.obs{i}';
    MySurfaces.obsP{i} = MySurfaces.obsP{i,1}';
    % Sort the data along X to ensure monotonic increase in [x,y]
    % MySurfaces.obsP{i} = sortrows(MySurfaces.obsP{i},[1,2]);
    % MySurfaces.obsP{i} = unique(MySurfaces.obsP{i},'rows');
end

% Plot the rotated data
figure
hold on
h = zeros(size(MySurfaces.obsP));
for i = 1:2*mc
    h(i) = plot(MySurfaces.obsP{i} (:,1),MySurfaces.obsP{i} (:,2),...
        'LineStyle',MyLine{dummy(i),:}, 'Color',MyColor(i,:));
end
xlabel('Xp-position [cm]')
% ylabel('Yp-position [cm]')
box on, grid on

% SECTION 3.2.2: Rotate by Individual YModel and calculate residual
MySurfaces.modelP = cell(2*mc,1);
MySurfaces.residual= cell(2*mc,1); % topography
MySurfaces.residual0=cell(2*mc,1); % topography where the lowest point is 0
MySurfaces.gradient= cell(2*mc,1); % numerical first derivative of
% residuals (i.e., topography)
for i = 1:2*mc
    t = -atan(MySurfaces.model{i,2}(1));
    ROT = [cos(t),sin(t);-sin(t),cos(t)];
    MySurfaces.modelP{i} = ROT'*MySurfaces.model{i}';
    MySurfaces.modelP{i} = MySurfaces.modelP{i}';

```

```

% [X, Yobs - Ymodel]
MySurfaces.residual{i} = [MySurfaces.modelP{i}(:,1), ...
    MySurfaces.obsP{i}(:,2) - MySurfaces.modelP{i}(:,2)];

% Calculate slope (taken to coincide with midpoint of x-values
% --> Note this approach does not achieve even spacing between data
% points nor does it attempt to evaluate the wavelength dependence of
% slope
[m,n] = size(MySurfaces.residual{i}(:,1));
MySurfaces.residual0{i} = [MySurfaces.model{i}(:,1), ...
    MySurfaces.residual{i}(:,2)-min(MySurfaces.residual{i}(:,2))];
MySurfaces.gradient{i}(:,1) = MySurfaces.residual{i}(1:m-1,1) + ...
    diff(MySurfaces.residual{i}(:,1));
% deltaX./deltaY
MySurfaces.gradient{i}(:,2) = diff(MySurfaces.residual{i}(:,2))./...
    diff(MySurfaces.residual{i}(:,1));
end

%-----
%% SECTION 3.3: Rotate by Oldest (i.e., first) YModel & calculate residual
MySurfaces.old.modelP = cell(2*mc,1);
MySurfaces.old.residual= cell(2*mc,1); % topography
MySurfaces.old.residual0=cell(2*mc,1); % topography where lowest point is 0
MySurfaces.old.gradient= cell(2*mc,1); % numerical first derivative of
% residuals (i.e., topography)
MySurfaces.old.obsP = cell(2*mc,1);

t = -atan(MySurfaces.model{1,2}(1));
ROT = [cos(t),sin(t);-sin(t),cos(t)];
for i = 1:2*mc
    MySurfaces.old.obsP{i} = ROT'*MySurfaces.obs{i}';
    MySurfaces.old.obsP{i} = MySurfaces.old.obsP{i,1}';

    MySurfaces.old.modelP{i} = ROT'*MySurfaces.old.model{i}';
    MySurfaces.old.modelP{i} = MySurfaces.old.modelP{i}';

    % [X, Yobs - Ymodel]
    MySurfaces.old.residual{i} = [MySurfaces.old.modelP{i}(:,1), ...
        MySurfaces.old.obsP{i}(:,2) - MySurfaces.old.modelP{i}(:,2)];

    % Calculate slope (taken to coincide with midpoint of x-values
    % --> Note this approach does not achieve even spacing between data
    % points nor does it attempt to evaluate the wavelength dependence of
    % slope
    [m,n] = size(MySurfaces.old.residual{i}(:,1));
    MySurfaces.old.residual0{i} = [MySurfaces.old.model{i}(:,1), ...
        MySurfaces.old.residual{i}(:,2) - ... 
        min(MySurfaces.old.residual{i}(:,2))];
    MySurfaces.old.gradient{i}(:,1) = ...
        MySurfaces.old.residual{i}(1:m-1,1) + ...
        diff(MySurfaces.old.residual{i}(:,1));
    % deltaX./deltaY
    MySurfaces.old.gradient{i}(:,2) = ...
        diff(MySurfaces.old.residual{i}(:,2))./...
        diff(MySurfaces.old.residual{i}(:,1));
end

%%%%%%%%%%%%%
%%          Section 4: Basic Surface Roughness          %%
%%%%%%%%%%%%%

```

```

% SECTION 4.1a:
figure
xmin = 0;
xmax = 3.2;
% use tight subplot instead of subplot
% Data rotated individually
subplot(3,2,1)
hold on
h = zeros(size(MySurfaces.obsP));
for i = 1:2*mc
    h(i) = plot(MySurfaces.residual{i}(:,1),MySurfaces.residual{i}(:,2),...
        'LineStyle',MyLine{dummy(i),:},'Color',MyColor(i,:));
end
title('Individually Evaluated 2D Surfaces')
box on, grid on
% xlabel('Xp-position [cm]')
ylabel('Topography [cm]')
xlim([xmin xmax])
ylim([-0.3 0.2])

% SECTION 4.2a: Slope variation by position; individually rotated data
subplot(3,2,3)
hold on
for i = 1:2*mc
    plot(MySurfaces.gradient{i}(:,1),MySurfaces.gradient{i}(:,2),...
        'LineStyle',MyLine{dummy(i),:},'Color',MyColor(i,:));
    % semilogy(MySurfaces.gradient{i}(:,1),MySurfaces.gradient{i}(:,2),...
    %     'LineStyle',MyLine{dummy(i),:},'Color',MyColor(i,:));
end
ylabel('dTopo/dx []')
xlabel('Xp-position [cm]')
xlim([xmin xmax])
ylim([-10 10])
grid on, box on

% SECTION 4.3a: Data statistics of topography for individually rotated data
subplot(3,2,5)
hold on
s = 0.001;
bins = [0:s:0.25];
% MySurfaces.hist = cell(2*mc,1);
MySurfaces.hist = zeros(2*mc,length(bins))*nan;
for i = 1:2*mc
    % MySurfaces.hist{i} = hist(MySurfaces.residual{i}(:,2) + ...
    %     min(MySurfaces.residual{i}(:,2)),bins);
    % semilogy(bins,MySurfaces.hist{i},...
    %     'LineStyle',MyLine{dummy(i),:},'Color',MyColor(i,:));
    MySurfaces.hist(i,:) = hist(MySurfaces.residual0{i}(:,2),bins);
    semilogy(bins,MySurfaces.hist(i,:),...
        'LineStyle',MyLine{dummy(i),:},'Color',MyColor(i,:));
end
ylabel('log(10) Frequency')
xlabel(['Topography [cm]: bin size = ',num2str(s)])
box on, grid on
xlim([0,max(bins)])

%-----
% SECTION 4.1b: Data rotated by lsq derived from oldest surface
subplot(3,2,2)
hold on

```

```

h = zeros(size(MySurfaces.old.obsP));
for i = 1:2*mc
    h(i) = plot(MySurfaces.old.residual{i}(:,1),...
        MySurfaces.old.residual{i}(:,2),...
        'LineStyle',MyLine{dummy(i),:},'Color',MyColor(i,:));
end
title('Relative to Oldest Evaluated 2D Surfaces')
box on, grid on
% xlabel('Xp-position [cm]')
ylabel('Topography [cm]')
xlim([xmin xmax])
ylim([-0.3 0.2])

% SECTION 4.2b: Data statistics for lsq to oldest surface
subplot(3,2,4)
hold on
for i = 1:2*mc
    plot(MySurfaces.old.gradient{i}(:,1),...
        MySurfaces.old.gradient{i}(:,2),...
        'LineStyle',MyLine{dummy(i),:},'Color',MyColor(i,:));
    % semilogy(MySurfaces.gradient{i}(:,1),MySurfaces.gradient{i}(:,2),...
    % 'LineStyle',MyLine{dummy(i),:},'Color',MyColor(i,:));
end
ylabel('dTopo/dx []')
xlabel('Xp-position [cm]')
xlim([xmin xmax])
ylim([-10 10])
grid on, box on

% SECTION 4.3b: Slope variation by position; lsq to oldest surface
subplot(3,2,6)
hold on
% MySurfaces.hist = cell(2*mc,1);
MySurfaces.old.hist = zeros(2*mc,length(bins))*nan;
for i = 1:2*mc
    % MySurfaces.hist{i} = hist(MySurfaces.residual{i}(:,2) + ...
    % min(MySurfaces.residual{i}(:,2)),bins);
    % semilogy(bins,MySurfaces.hist{i},...
    % 'LineStyle',MyLine{dummy(i),:},'Color',MyColor(i,:));
    MySurfaces.old.hist(i,:) = hist(MySurfaces.old.residual0{i}(:,2),bins);
    semilogy(bins,MySurfaces.old.hist(i,:),...
        'LineStyle',MyLine{dummy(i),:},'Color',MyColor(i,:));
end
ylabel('log(10) Frequency')
xlabel(['Topography [cm]: bin size = ',num2str(s)])
box on, grid on
xlim([0,max(bins)])

%% SECTION 4.4: Position versus standard deviation of topography within bin
-----PLACEHOLDER-----
%%%%%
%%%%% SECTION 5: Relative Surface age versus roughness %%%
%%%%% SECTION 5.1a: Surface age versus roughness statistics in box plots
figure
subplot(3,1,1)
hold on
% Surfaces are ordered from oldest to youngest (best to do in reference to

```

```

% original fracture plane???)
```

```

% Transform data back into a simple two column vector for use with boxplot
% group functionality
MySurfaces.boxplot.data = [];
MySurfaces.boxplot.group= [];
for i = 1:2*mc
    dummyD = MySurfaces.residual0{i}(:,2);
    [m,n] = size(dummyD);
    MySurfaces.boxplot.data = [MySurfaces.boxplot.data;...
        dummyD];
    MySurfaces.boxplot.group = [MySurfaces.boxplot.group;...
        ones(m,n)*i];
end
```

```

MySurfaces.boxplot.labels=cell(1,2*mc);
dummyL = ['T';'B'];
for i = 1:2*mc
    if mod(i,2) == 0
        j = 2;
    else
        j = 1;
    end
    MySurfaces.boxplot.labels{i} = [num2str(ceil(i/2)), '.', dummyL(j)];
end
```

```

h = boxplot(MySurfaces.boxplot.data,MySurfaces.boxplot.group, ...
    'notch','on','colors',MyColor, ...
    'labels',MySurfaces.boxplot.labels,'labelorientation','inline', ...
    'outliersize',4,'symbol','k+');
ylabel('Topographic Relief [cm]')
xlabel('Surface in age order Oldest to Youngest')
box on, grid on
title('Analysis of independently evaluated surfaces')
```

```

%-----
subplot(3,1,2)
hold on
% Surfaces are ordered from oldest to youngest (best to do in reference to
% original fracture plane???)
```

```

% transofrm data back into a simple two column vector
MySurfaces.old.boxplot.data = [];
MySurfaces.old.boxplot.group= [];
for i = 1:2*mc
    dummyD = MySurfaces.old.residual0{i}(:,2);
    [m,n] = size(dummyD);
    MySurfaces.old.boxplot.data = [MySurfaces.old.boxplot.data;...
        dummyD];
    MySurfaces.old.boxplot.group = [MySurfaces.old.boxplot.group;...
        ones(m,n)*i];
end
```

```

h = boxplot(MySurfaces.old.boxplot.data,MySurfaces.old.boxplot.group, ...
    'notch','on','colors',MyColor, ...
    'labels',MySurfaces.boxplot.labels,'labelorientation','inline', ...
    'outliersize',4,'symbol','k+');
ylabel('Topographic Relief [cm]')
xlabel('Surface in age order Oldest to Youngest')
box on, grid on
title('Analysis of surfaces relative to oldest surface')
```

```

%-----
%% SECTION 5.1b: Find ratio of straight line length and surface length
% SECTION 5.1b-1: Find ratios for individually analyzed surfaces
subplot(3,1,3)
MySurfaces.Ltotal = cell(2*mc,1);
MySurfaces.Lmin = cell(2*mc,1);
MySurfaces.Lratio = cell(2*mc,1);
MySurfaces.xmax = ones(2*mc,1)*nan;
for i = 1:2*mc
    xdummy = MySurfaces.obsP{i}(:,1);
    ydummy = MySurfaces.obsP{i}(:,2);

    % To calculate the length between adjacent points (should really be its
    % own function)
    MySurfaces.Ltotal{i} = sum(sqrt((diff(xdummy)).^2+(diff(ydummy)).^2));
    MySurfaces.Lmin{i} = max(MySurfaces.modelP{i}(:,1)) - ...
        min(MySurfaces.modelP{i}(:,1));
    MySurfaces.Lratio{i} = MySurfaces.Ltotal{i}/MySurfaces.Lmin{i};

    hold on
    h3 = plot(i,MySurfaces.Lratio{i},...
        'o','MarkerFaceColor',MyColor(i,:),'MarkerEdgeColor','k',...
        'MarkerSize',9);
    MySurfaces.xmax(i) = max(MySurfaces.modelP{i}(:,1));
end

% SECTION 5.1b-2: Find and plot ratios using oldest surface as reference
MySurfaces.old.Ltotal = cell(2*mc,1);
MySurfaces.old.Lmin = cell(2*mc,1);
MySurfaces.old.Lratio = cell(2*mc,1);
MySurfaces.old.xmax = ones(2*mc,1)*nan;
for i = 1:2*mc
    xdummy = MySurfaces.old.obsP{i}(:,1);
    ydummy = MySurfaces.old.obsP{i}(:,2);

    % To calculate the length between adjacent points (should really be its
    % own function)
    MySurfaces.old.Ltotal{i} = sum(sqrt((diff(xdummy)).^2 + ...
        (diff(ydummy)).^2));
    MySurfaces.old.Lmin{i} = max(MySurfaces.old.modelP{i}(:,1)) - ...
        min(MySurfaces.old.modelP{i}(:,1));
    MySurfaces.old.Lratio{i} = MySurfaces.old.Ltotal{i}/...
        MySurfaces.old.Lmin{i};

    hold on
    h4 = plot(i,MySurfaces.old.Lratio{i},...
        '^','MarkerFaceColor',MyColor(i,:),'MarkerEdgeColor','k',...
        'MarkerSize',9);
    MySurfaces.old.xmax(i) = max(MySurfaces.old.modelP{i}(:,1));
end

h1 = plot([1:2*mc]',cell2mat(MySurfaces.old.Lmin),'-k');
h2 = plot([1:2*mc]',cell2mat(MySurfaces.Lmin),'-r');
legend([h1 h2 h3 h4],'Lmin [cm] (rotated by oldest surfaces)',...
    'Lmin [cm] (individually rotated)',...
    'Lf/Lmin: Individually evaluated',...
    'Lf/Lmin: Referenced to original surface');

set(gca,'XTickLabel',MySurfaces.boxplot.labels,'XMinorTick','on')
set(gca,'XTick',1:length(MySurfaces.boxplot.labels));
grid on, box on
xlabel('Surface in age order Oldest to Youngest')
ylabel('L(total)/L(linear) []')

```

```

title('Comparison of Surface Length to StraightLine Length')

%-----
% SECTION 5.2: Plot in image fracture length versus ratio of
% Ltotal/L(linear)
% --> once fracture links the total length jumps and the roughness suddenly
%     becomes small relative to the new greater length in the plot
% --> possibly add second axis with lengths to previous plot above

%-----PLACEHOLDER-----


%-----
% SECTION 5.3: Age versus standard deviation of topography for each
% surface

%-----PLACEHOLDER-----


%%%%%%%%%%%%%%%
% SECTION 6: Anaylsus of Wavelength and Power %%
%%%%%%%%%%%%%%%
% Surfaces roughness (same could be performed on thickness distribution)

% SECTION 6.1: Data Preparatio: Resample rotated data
% --> resampling to produce a monotonic progression could obscure some
% path-dependent features of the surface texture.
MySurfaces.residualInt = cell(2*mc,1);
MySurfaces.unique.residual = cell(2*mc,1);
% int = 0.001;
int = 0.0005;
MySurfaces.XI = [0:int:max(MySurfaces.old.xmax)]';
% ***** FUTURE IMPROVEMENT: rebuild this to make it more flexible by
%     determining the xmax above in the rotated data above
for i = 1:2*mc
    % Sort the data along X to ensure monotonic increase in [x,y]
    MySurfaces.unique.residual{i} = sortrows(MySurfaces.residual{i},[1,2]);

    % There might be some degredation of the surface since this step
    % enforces a monotic function
    [A,IA] = unique(MySurfaces.unique.residual{i}(:,1));
    MySurfaces.unique.residual{i} = ...
        [MySurfaces.unique.residual{i}(IA,1), ...
        MySurfaces.unique.residual{i}(IA,2)];

    YI = interp1(MySurfaces.unique.residual{i}(:,1), ...
        MySurfaces.unique.residual{i}(:,2),...
        MySurfaces.XI,'linear',nan); % spline smoother, but produces some
        % perturbations, linear for simplest
    MySurfaces.residualInt{i} = [MySurfaces.XI,YI];
end

% Example plot of Original Surfaces and resampled surfaces
figure
for i = 1:2*mc
    h1 = plot(MySurfaces.XI,MySurfaces.residualInt{i}(:,2),...
        'Color',MyColor(i,:));
    hold on
    h2 = plot(MySurfaces.unique.residual{i}(:,1),...
        MySurfaces.unique.residual{i}(:,2),'o',...
        'Color',MyColor(i,:),'MarkerSize',4);
end

```

```

title('Interpolation Test')
xlabel('x-position [cm]')
ylabel('Topography [cm]')
grid on, box on
legend([h1 h2],'Fit','Obs')

%-----
%% SECTION 6.2a: Calculate Power Spectral Density Distribution
figure
subplot(1,2,1)
hold on

MySurfaces.unique.p      = cell(2*mc,1);
MySurfaces.unique.freq = cell(2*mc,1);
for i = 1:2*mc % See: http://faculty.olin.edu/bstorey/Notes/Fourier.pdf
    % APPROACH: Define function, here given as observed roughness
    % (topography as f(position)) [m]
    y    = MySurfaces.unique.residual{i}(:,2)*0.01;
    % ALTERNATE:
    % define function, here given as observed roughness
    % (topography as f(position))
    % y    = MySurfaces.residualInt{i}(:,2);

    I    = isnan(y) == 0; % remove nans
    y    = y(I);
    % x    = MySurfaces.residualInt{i}(:,1); % sample positions
    x    = MySurfaces.unique.residual{i}(:,1)*0.01; % sample positions [m]
    x    = x(I);

    N    = length(x);           % Number of samples within period of signal
    nfft= 10*N;
    L    = (max(x)-min(x));   % Length of signal: Total sampling time:
                               % "time"interval --> length of fracture in m
    T    = L/N;                % sampling spacing (space or time between
                               % samples; should == variable int above)
    fs  = N/L;                % frequency of samples [1/m]
    % nw  = T;                  % time-bandwidth product
    % df  = 1/(N*dt);          % frequency resolution

    % OPTIONAL Tools for calculating PSD from an observed signal:
    % - Basic Calculation from Fast Fourier Analysis (FFT)
    % p    = abs(fft(y))/(N/2); % absolute value of fft
    % p    = p(1:N/2).^2;       % take the power of positive frequency half
    % freq=[0:N/2-1]'/L;       % find the corresponding frequency in 1/cm

    % - FUNCTION: SPECTRUM
    % spectrum
    % - FUNCTION: PERIODOGRAM
    % [p,freq] = periodogram(y,[],'onesided',N,fs); % periodogram

    % - FUNCTION: MULTITAPER
    % p = pmtm(x,nw)           % multitaper
    % [p,freq] = pmtm(y,[],nfft,fs,'onesided'); % multitaper
    [p,freq] = pmtm(y,[],'onesided',N,fs); % multitaper

    % - FUNCTION: WELCH
    % p = pwelch(x);           % welch

    % Plot the data
    plot(log10(freq),log10(p),...
        'LineStyle',MyLine{dummy(i),:}, 'Color',MyColor(i,:));

```

```

    % Store the data
    MySurfaces.unique.p{i} = p; % power spectral density
    MySurfaces.unique.freq{i} = freq; % corresponding frequencies in [1/m]
end
grid on, box on
title('Periodogram')
xlabel('log10(Frequency) [1/m]')
ylabel('log10(Power) [m^2/(1/m)]')
axis equal

% SECTION 6.2b: Critical dimensions
% RREFERENCE OBS LENGTH SCALE: Thin sections length
ts = 2.5*1/100; % width of thin section in [m]
yy = get(gca,'YLim');
plot(log10([1/ts 1/ts]),yy,'r--','LineWidth',1.5)

% RREFERENCE OBS LENGTH SCALE: Minimum sampling distance
xx = fs; % (min(diff(MySurfaces.unique.residual{i}(:,1)))*.01)^-1;
plot(log10([xx xx]),yy,'r--','LineWidth',1.5)

% REFERENCE CHARACTERISTIC DIMENSION: Mean Grain Size
%-----PLACEHOLDER-----

% REFERENCE CHARACTERISTIC DIMENSION: Mean Pore Size
%-----PLACEHOLDER-----

% SECTION 6.2c: Power Spectra Slope and fractal dimension
% Lower bound of data used in fitting:
LowerBound.all = 10^2.5; % 1/(4*10^-4);
% Upper bound of data used in fitting:
UpperBound.all = 10^4.3; % 1/(0.5*10^-4);

LowerBound.low = LowerBound.all;
UpperBound.low = 10^3.774;
LowerBound.high= UpperBound.low;
UpperBound.high= UpperBound.all;

plot(log10([UpperBound.low UpperBound.low]),yy,'k--','LineWidth',1)

N = 1; % degree of polynomial fit
MySurfaces.unique.fit.all = ones(2*mc,2)*nan;
MySurfaces.unique.fit.low = ones(2*mc,2)*nan;
MySurfaces.unique.fit.high = ones(2*mc,2)*nan;

for i = 1:2*mc
    % fit all
    I = find(MySurfaces.unique.freq{i} >= ...
        LowerBound.all & MySurfaces.unique.freq{i} <= UpperBound.all);
    x = log10(MySurfaces.unique.freq{i}(I));
    y = log10(MySurfaces.unique.p{i}(I));
    [lsq,S] = polyfit(x,y,N);
    % polyval(x,lsq,S);
    MySurfaces.unique.fit.all(i,:) = lsq;

    % fit steep interval (lower frequencies)
    I = find(MySurfaces.unique.freq{i} >= ...
        LowerBound.low & MySurfaces.unique.freq{i} <= UpperBound.low);

```

```

x = log10(MySurfaces.unique.freq{i}(I));
y = log10(MySurfaces.unique.p{i}(I));
[lsq,S] = polyfit(x,y,N);
MySurfaces.unique.fit.low(i,:) = lsq;

% fit shallower interval (higher frequencies)
I = find(MySurfaces.unique.freq{i} >= ...
    LowerBound.high & MySurfaces.unique.freq{i} <= UpperBound.high);
x = log10(MySurfaces.unique.freq{i}(I));
y = log10(MySurfaces.unique.p{i}(I));
[lsq,S] = polyfit(x,y,N);
MySurfaces.unique.fit.high(i,:) = lsq;

end
subplot(1,2,2)
% plot fit to all data
h1 = plot([1:2*mc],MySurfaces.unique.fit.all(:,1),...
    'or','MarkerFaceColor','y');
hold on
% plot fit to low frequencies
h2 = plot([1:2*mc],MySurfaces.unique.fit.low(:,1),...
    'sb','MarkerFaceColor','c');
% plot fit to high frequencies
h3 = plot([1:2*mc],MySurfaces.unique.fit.high(:,1),...
    'vk','MarkerFaceColor','r');
% Formatting of plot
legend([h1 h2 h3],...
    ['Whole range (10^{' ,num2str(log10(LowerBound.all)),...
        '} to 10^{' ,num2str(log10(UpperBound.all)),'} )'],...
    ['Low Frequencies (10^{' ,num2str(log10(LowerBound.low)),...
        '} to 10^{' ,num2str(log10(UpperBound.low)),'} )'],...
    ['High Frequencies (10^{' ,num2str(log10(LowerBound.high)),...
        '} to 10^{' ,num2str(log10(UpperBound.high)),'} )']);
set(gca,'XTickLabel',MySurfaces.boxplot.labels,'XMinorTick','on')
set(gca,'XTick',1:length(MySurfaces.boxplot.labels));
grid on, box on
xlabel('Relative surface age')
ylabel('Power Spectral Slope [(m^2/(1/m))/m]')
ylim([-3 0])

%%%%%%%%%%%%%%%
%% Analysis of Slip & Dilatation history (Paired Surfaces Separation) %%
%%%%%%%%%%%%%%%
% Dilatation history analysis of paired surfaces
% --> Use resamples surfaces from wavelength analysis samples at the same
%     x-positions
% -->
-----PLACEHOLDER-----

```

Supporting Function: (2 of 3): Reorder Digitized Line Data by Nearest Neighbor Relationship

```

function [XI,YI,I,P] = ReorderByNearestNeighbor2(x,y,varargin)

% AUTHOR: N.C. Davatzes
% DATE CREATED: 2013-04-30
% DATE MODIFIED: 2013-05-01
% VERSION: 2
%
% PURPOSE
%   Reorder data points by nearest neighbor.
%
% INPUTS
%   x
%   y
%   Start
%   MaxGap
%
% SYNTAX
%   [XI,YI,I,P] = ReorderByNearestNeighbor2(x,y)
%   [XI,YI,I,P] = ReorderByNearestNeighbor2(x,y,'Start',Val)
%   [XI,YI,I,P] = ReorderByNearestNeighbor2(x,y,'MaxGap',Val)
%   [XI,YI,I,P] = ReorderByNearestNeighbor2(x,y,'MaxGap',Val,'Start',Val)
%   [XI,YI,I,P] = ReorderByNearestNeighbor2(x,y,'Start',Val,'MaxGap',Val)
%
% OUTPUTS
%   XI ordered x-component
%   YI ordered y-component
%   I Index order
%   P path length
%   varargin input options related to the dist function
%
% NOTE:
% - Add a maximum gap length enforcement: MaxGap
% - If MaxGap is empty, then set
%   MaxGap = 0.1*max(max(dist));
% - An alternate formulation would also consider the number of samples as
%   a qualitative indicator of data density to define the scaling
%   constant instead of an arbitrary value.
% - Problem: If data is not regularly spaced or in non-monotonic data
%   paths and the sample spacing of the paths is less than the spacing of
%   data points intended to defin the path,
%   --> then this formulation could either (1) prevent convergence to a
%   solution or intersperse the data
% - APPROACH:
%   (1) Use MaxGap Criterion
%   (2) Define path
%   (3) Identify segments that exceed criterion
%   (4) Modify series
%     (a) Identify sub-series of data separated by distances greater
%         than MaxGap
%     (b) Calculate the series length
%     (C) Identify the largest continuous series of data meeting the
%         criterion
%     (b) For separate segments of path data where the length is
%         greater than two, L>=2, split matrix and concatonate in a
%         more appropriate order

```

```

%
% (a) For isolated data points, L==1, within the span of the data
%       series Insert data violating MaxGap into nearest path
%       position based on a dist2 calculation
%
% - ALTERNATE APPROACH
%   - Look into meshing algorithms such as delaunay...
%

%%%%%%%%%%%%%%%
%% 1.0 Organize Data and I/O Check
%%%%%%%%%%%%%%%
%% 1.1 Variable matrix shape
[m,n] = size(x);
if n>m
    x = x';
end
[m,n] = size(y);
if n>m
    y = y';
end

data = [x,y];
% dist = pdist2(data,data,varargin{:});
dist = pdist2(data,data);

%% 1.2 I/O Check
% Input arguments: Test for existence, if do not exist set to default
if nargin < 2
    error('too few input parameters')
end
if nargin == 2;
    MaxGap = [];
    Start = 1;
elseif nargin > 3
    for i = 1:floor(length(varargin))/2
        eval([varargin{i*2-1}, '=' ,num2str(varargin{i*2}),';']);
    end
    if ~exist(MaxGap)
        MaxGap = [];
    elseif isempty(MaxGap) == 1;
        MaxGap = 0.1*max(max(dist));
    end
    if ~exist(Start)
        Start = 1;
    end
end

% ADDITIONAL I/O Checks
% if nargin < 3
%     MaxGap = [];
% elseif isempty(MaxGap) == 1;
%     MaxGap = 0.1*max(max(dist));
% end
%
% if nargin < 4
%     varargin = {};
% end

%%%%%%%%%%%%%%%
%% 2.0 USE DIST and DIST2 To Develop Ordering of Points
%%%%%%%%%%%%%%%
if isempty(MaxGap)
    % Basic ordering algorithm based on output from DIST2
    % --> very sensitive to data spacing and start position

```

```

N = size(data,1);
result = NaN(N,1);
%     result(1) = 1; % first point is first row in data matrix
result(1) = Start; % first point is first row in data matrix

for ii=2:N
    dist(result(ii-1),:) = Inf;
    [~, closest_idx] = min(dist(:,result(ii-1)));
    result(ii) = closest_idx;
end

% Index order of original: rows of I correspond to the ordered indices
% of x and y by nearest neighbor distances
I = result;

% Orderd data output
XI = x(I);
YI = y(I);
else
%% OPTION 2 : Modified ordering algorithm based on output from DIST2 but
% which tests the gap sizes generated
% --> might not converge to a solution

%-----PLACEHOLDER-----
end
end

%%%%%%%%%%%%%
%%% 3.0 SUBFUNCTION
%%%%%
function tf = isdefined(in_var_name)
%% tf = isdefined(in_var_name)
%
% Returns a logical indicating if the variable ?in_var_name? both exists
% and is not empty in the ?caller? workspace.
%
% ?in_var_name? is a text string that specifies the name of the variable to
% search for.
%
% AUTORE: Kent Conover, 12-Mar-08

cmd_txt = ['exist(',in_var_name,',',var,')'];
if evalin('caller', cmd_txt);
    cmd_txt = ['~isempty(',in_var_name, ')'];
    if evalin('caller', cmd_txt);
        tf = true;
    else
        tf = false;
    end
else
    tf = false;
end
end

```

Supporting Function: (3 of 3): Grain and Pore Size Statistics in Support of Surface Roughness Interpretation

```
% TITLE: GrainSizePlot.m
%
% AUTHOR:
% DATE CREATED:
%
% PURPOSE:
%   Examine the correlation between fracture surface topography in
%   sample 3937FA and the size of grains and pores with greater than 0.2 mm
%   dimensions (i.e., phenocrysts as opposed to ground mass).
%
% INPUTS:
%   Compiles files of phenocryst and lare pore long and short axes.
%
% OUTPUTS:
%   Plots and statistical characteristics of grain and pore size frquency
%   distribution.
%
% NOTES:
%   Does not address frequency distribution, and thus potential role of the
%   large number of smaller than 0.2mm grains.

%%%%%%%%%%%%%%%
clear all, close all

%%%%%%%%%%%%%%%
% SECTION 1: Import DATA %
%%%%%%%%%%%%%%%
grains.m = load('Phenocrysts.txt');
pores.m = load('PoresLarge.txt');

grains.l = grains.m(:,2)*0.1; % long axis [cm]
grains.s = grains.m(:,3)*0.1; % short axis [cm]
grains.r = grains.m(:,4); % axial ratio

pores.l = pores.m(:,2)*0.1; % long axis [cm]
pores.s = pores.m(:,3)*0.1; % short axis [cm]
pores.r = pores.m(:,4); % axial ratio

%%%%%%%%%%%%%%%
% SECTION 2: Basic Statistical Characterization %
%%%%%%%%%%%%%%%

% Basic Statistics
grains.stat.l = quartile(grains.l);
grains.stat.s = quartile(grains.s);
grains.stat.r = quartile(grains.r);

pores.stat.l = quartile(pores.l);
pores.stat.s = quartile(pores.s);
pores.stat.r = quartile(pores.r);

% Calculate Histogram
inc = 0.01;
```

```

bins.L = [0.0:inc:1];
grains.stat.l.hist = hist(grains.l,bins.L);
grains.stat.s.hist = hist(grains.s,bins.L);
pores.stat.l.hist = hist(pores.l,bins.L);
pores.stat.s.hist = hist(pores.s,bins.L);

inc = 0.05;
bins.R = [0.0:inc:2];
grains.stat.r.hist = hist(grains.r,bins.R);
pores.stat.r.hist = hist(pores.r,bins.R);

%%%%%%%%%%%%%%%
%% SECTION 3: Plot Results %%
%%%%%%%%%%%%%%%
figure
ax1 = subplot(2,1,1);
hgl = plot(bins.L,grains.stat.l.hist,'r-','LineWidth',2);
hold on
hgs = plot(bins.L,grains.stat.s.hist,'r--','LineWidth',1);
hpl = plot(bins.L,pores.stat.l.hist,'b-','LineWidth',2);
hps = plot(bins.L,pores.stat.s.hist,'b--','LineWidth',1);
% Formatting
ylabel('Frequency')
xlabel('Length [cm]')
% xlim([0,0.2])
grid on, box on
legend([hgl hgs hpl hps],...
    ['Grain Long Axis, N = ',length(grains.l)],...
    'Grain Short Axis',...
    ['Pore Long Axis, N = ',length(pores.l)],...
    'Pore Short Axis')
title('Grains and Pores > 0.02 cm')

% Plot envelopes
yy = get(gca,'Ylim');
plot([grains.stat.l.Q2 grains.stat.l.Q2],yy,'m-','LineWidth',1.5)
plot([grains.stat.l.Q1 grains.stat.l.Q1],yy,'m--')
plot([grains.stat.l.Q3 grains.stat.l.Q3],yy,'m--')

plot([pores.stat.l.Q2 pores.stat.l.Q2],yy,'c-','LineWidth',1.5)
plot([pores.stat.l.Q1 pores.stat.l.Q1],yy,'c---', 'LineWidth',1)
plot([pores.stat.l.Q3 pores.stat.l.Q3],yy,'c---', 'LineWidth',1)

ylim(yy)

subplot(2,1,2)
plot(bins.R,grains.stat.r.hist,'r-','LineWidth',2);
hold on
plot(bins.R,pores.stat.r.hist,'b-','LineWidth',2);
% Formatting
ylabel('Frequency')
xlabel('Axial Ratio []');
grid on, box on

% Plot envelopes
yy = get(gca,'Ylim');
plot([grains.stat.r.Q2 grains.stat.r.Q2],yy,'m-','LineWidth',1.5)
plot([grains.stat.r.Q1 grains.stat.r.Q1],yy,'m--','LineWidth',1)
plot([grains.stat.r.Q3 grains.stat.r.Q3],yy,'m--','LineWidth',1)

plot([pores.stat.r.Q2 pores.stat.r.Q2],yy,'c-','LineWidth',1.5)
plot([pores.stat.r.Q1 pores.stat.r.Q1],yy,'c---','LineWidth',1)

```

```
plot([pores.stat.r.Q3 pores.stat.r.Q3],yy,'c--','LineWidth',1)  
ylim(yy)
```

APPENDIX B

AVERAGE POROSITY VALUES AND METHOD CHECKLIST FOR EACH SAMPLE

Average Porosity Value Results for Each Method

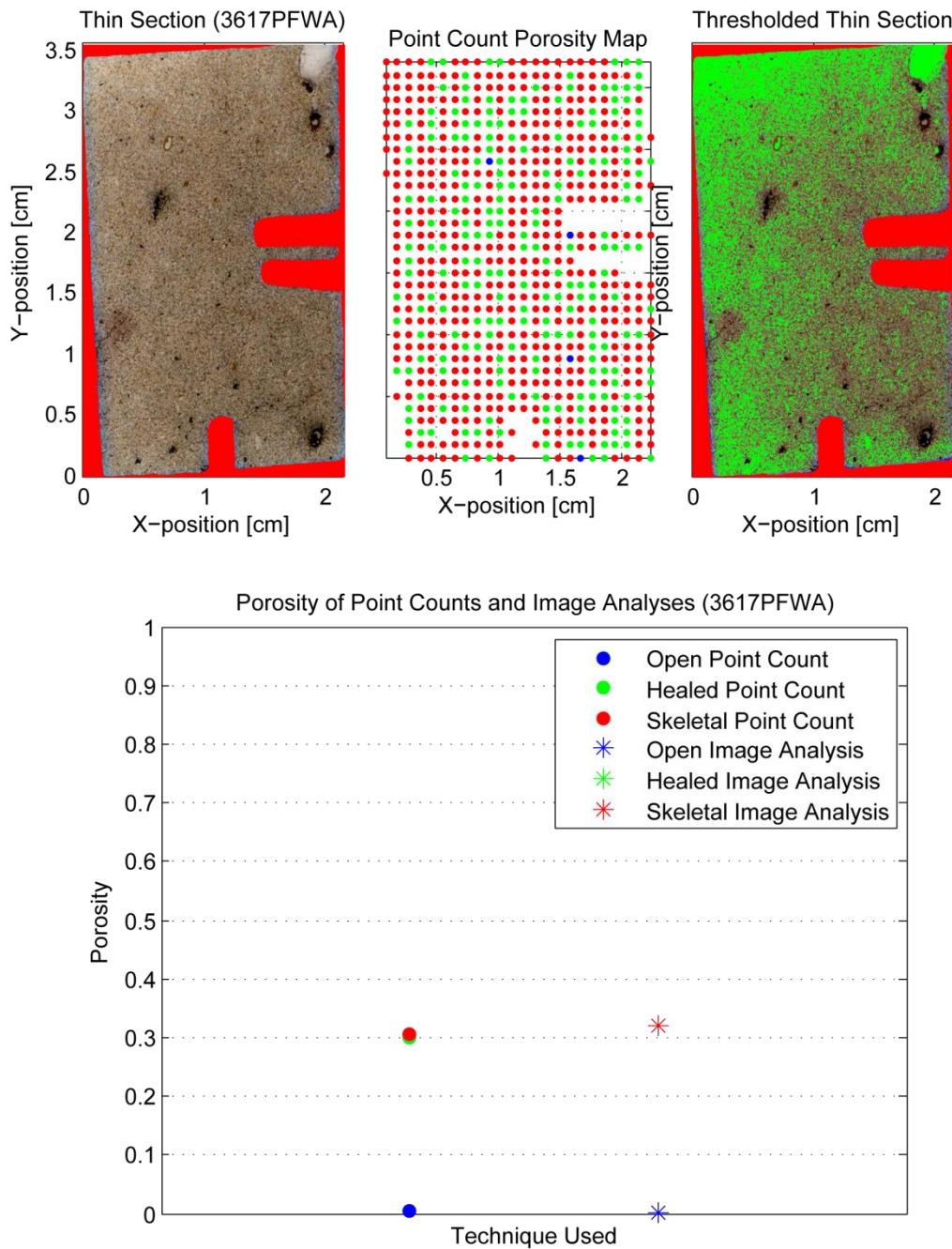
		N2-3617 PFWA	N2-3617 PFWB	N2-3937 FA	N2-4152 F	N2-4267 FA	N2-4303 PFWA	2-3523.5 WF	N2-4302 FA	N2-4306 FA
Point Count										
Open Average	0.60	0.14	0.90	2.70	3.30	6.00	0.60	0.70	3.70	
Healed Average	29.9	23.9	14.2	23.8	27.8	28.4	35.1	44.3	73.3	
Skeletal Average	30.5	24.1	15.1	26.5	31.1	34.4	35.6	45.0	77.0	
Automated Image										
Open Average	0.00	0.00	0.47	1.40	1.70	5.50	0.00	0.60	0.90	
Healed Average	31.9	28.8	24.0	33.7	24.7	38.5	41.0	63.4	63.3	
Skeletal Average	31.9	28.8	24.5	35.1	26.4	44.0	41.0	64.0	64.1	
High Res. Image										
Open Average	0.00	0.00	0.00	1.00	1.70	1.40	0.40	0.10	5.00	
Healed Average	32.1	25.8	13.7	16.4	33.0	22.0	41.8	22.0	56.4	
Skeletal Average	32.1	25.8	13.7	17.4	35.0	23.3	42.2	22.0	61.3	
Micro CT										
Open Average	0.00	0.00	0.00	1.60	0.30	1.00	0.00	0.00	0.20	
Healed Average	0.68	1.2	5.80	9.30	11.3	2.60	2.50	16.3	15.8	
Skeletal Average	0.68	1.2	5.80	10.8	11.7	3.70	2.50	16.3	16.0	
Stage:	2.0	2	2	3	3.1	4	4	4	5	

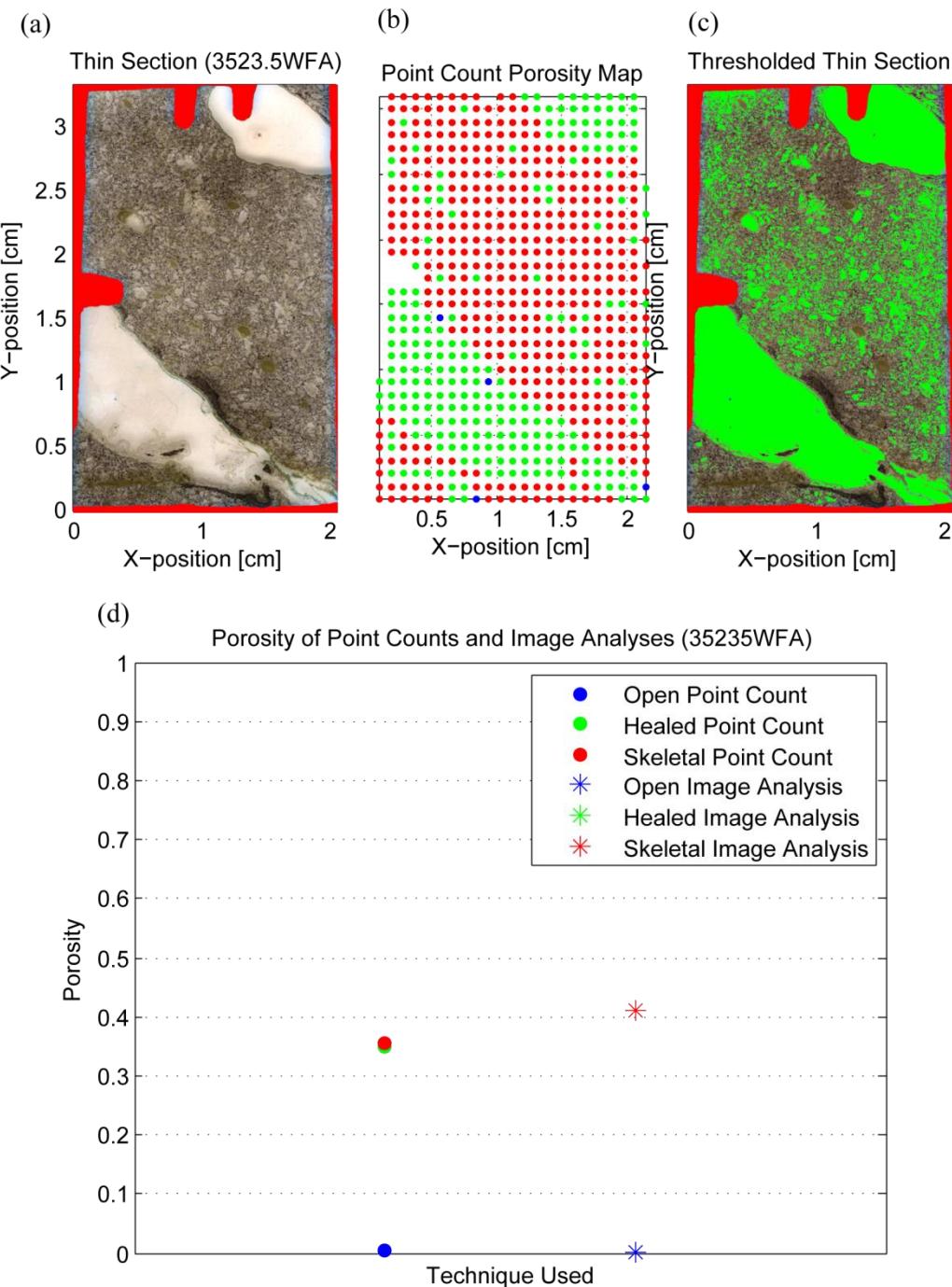
Method Checklist for Each Sample

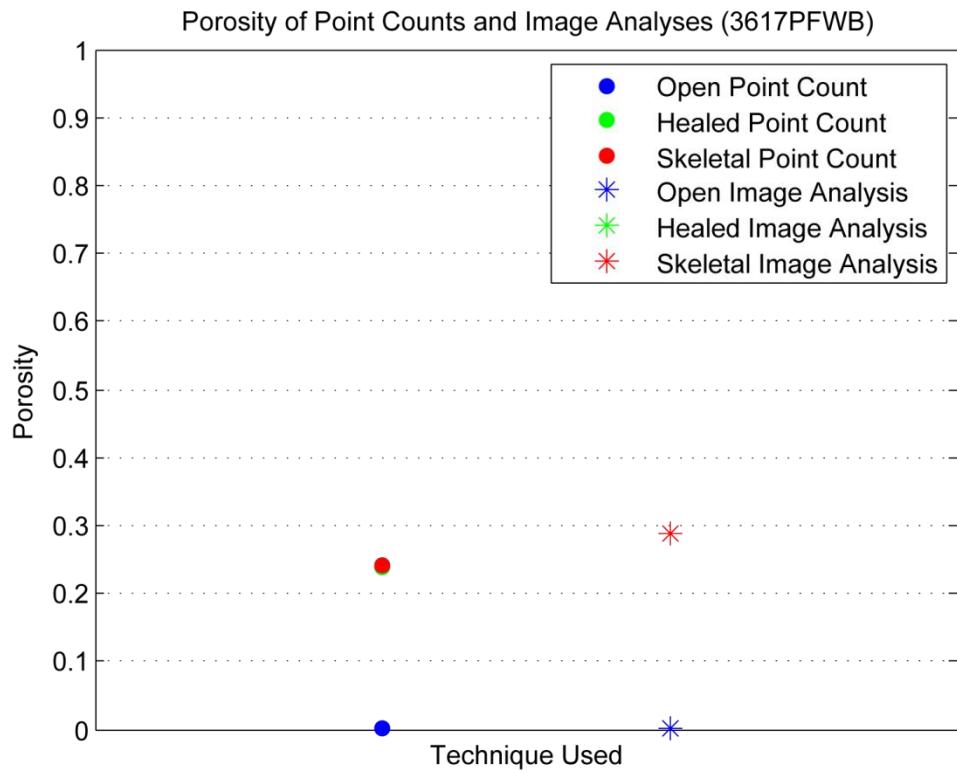
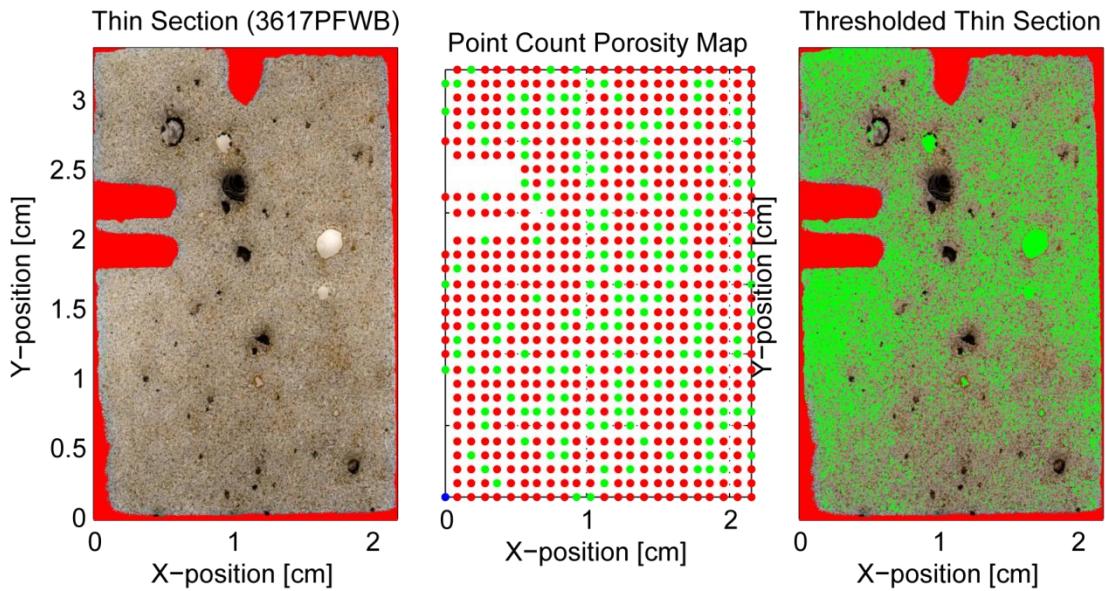
Newberry Core Name	Subcore Name	Fracture Stage	Thin Sectioned?	Point Counted?	Auto-Thresholded?	Auto-Porosity Map?	High Res. Auto	High Res. Manual	High Res. Porosity	High Res. Transect?	Micro CT Scanned?	Micro CT Thresholded?	Micro CT Porosity Transect?
N2-3523.5	N2-3523.5-FA	4	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
N2-3617	N2-3617-PFW	2	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
N2-3617	N2-3617-PFWB	2	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
N2-3937	N2-3937-F	2	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
N2-4152	N2-4152-F	3	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
N2-4267	N2-4267-FA	3 to 4	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
N2-4302	N2-4302-FA	4	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
N2-4303	N2-4303-PFWA	4	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
N2-4306	N2-4306-F	5	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

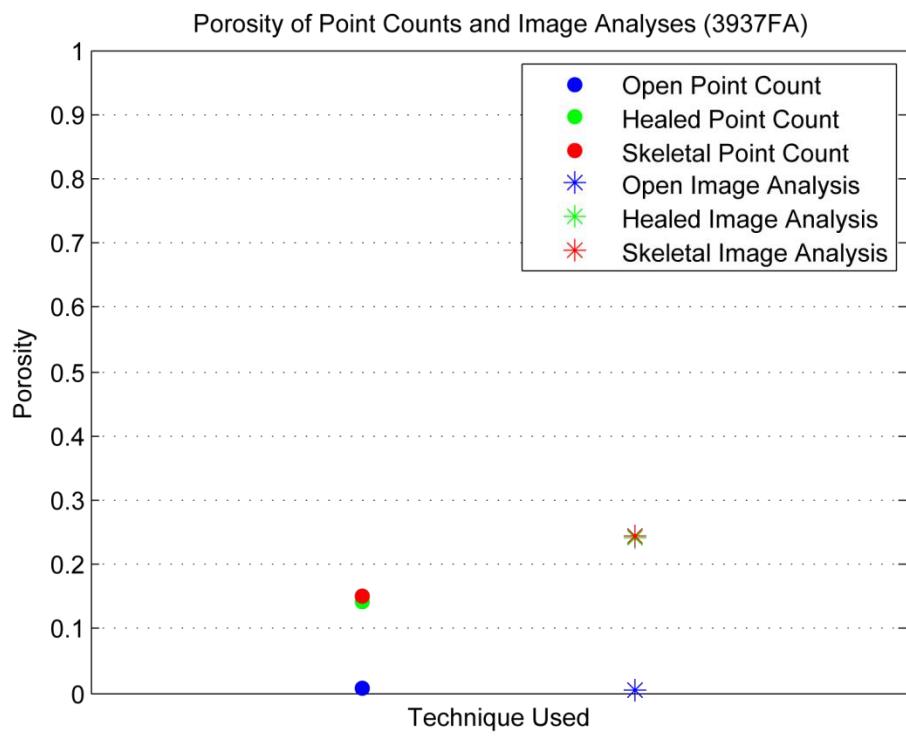
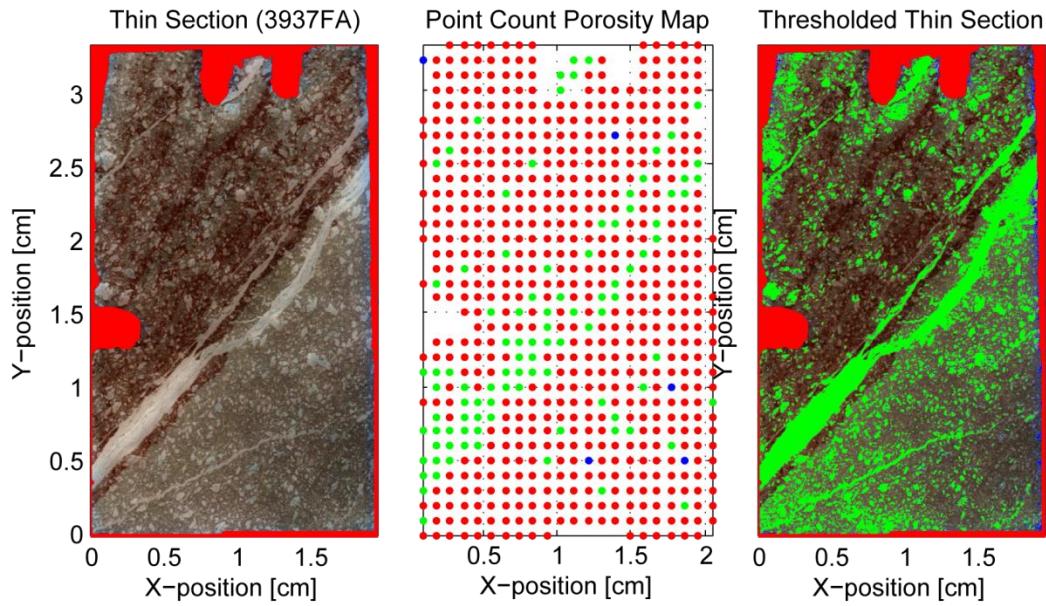
APPENDIX C

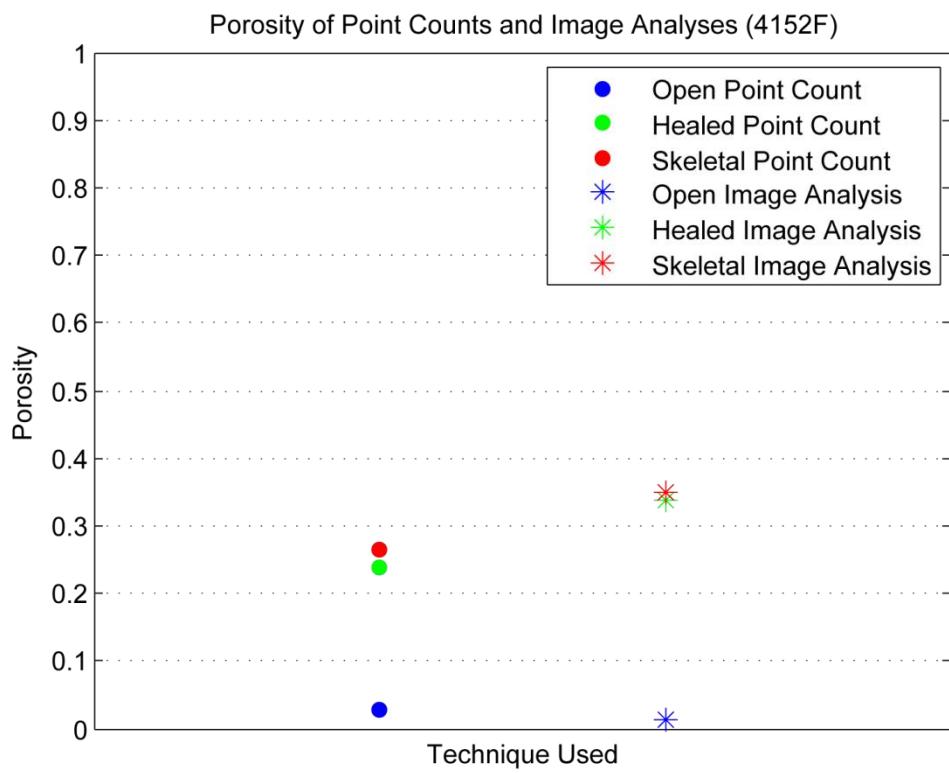
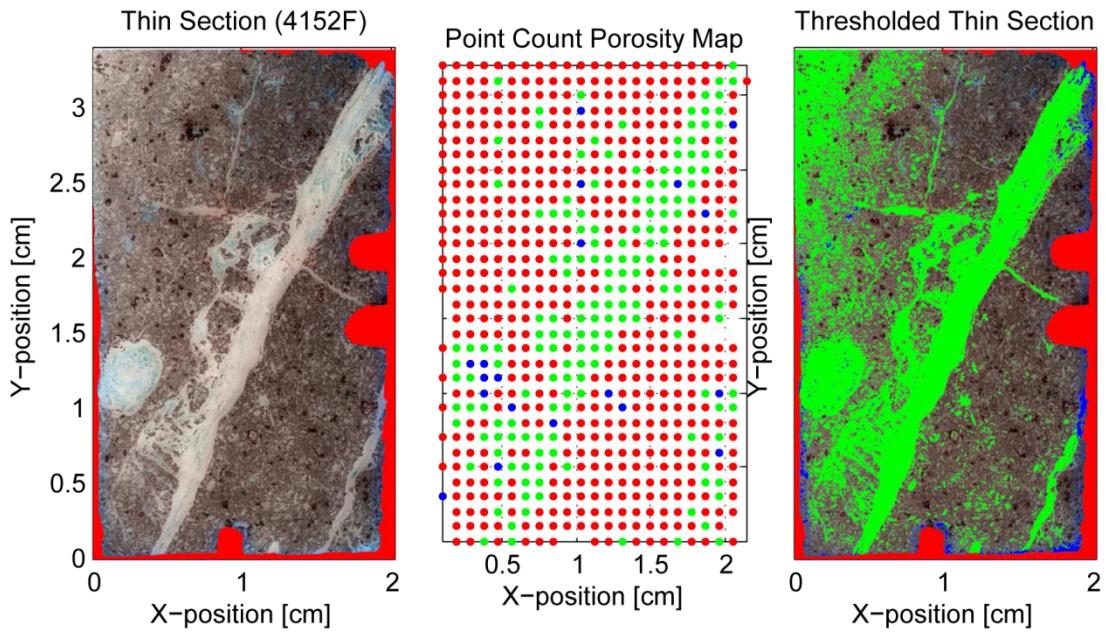
POROSITY MAPS AND PLOTS FOR POINT COUNTS AND AUTOMATED IMAGE ANALYSIS

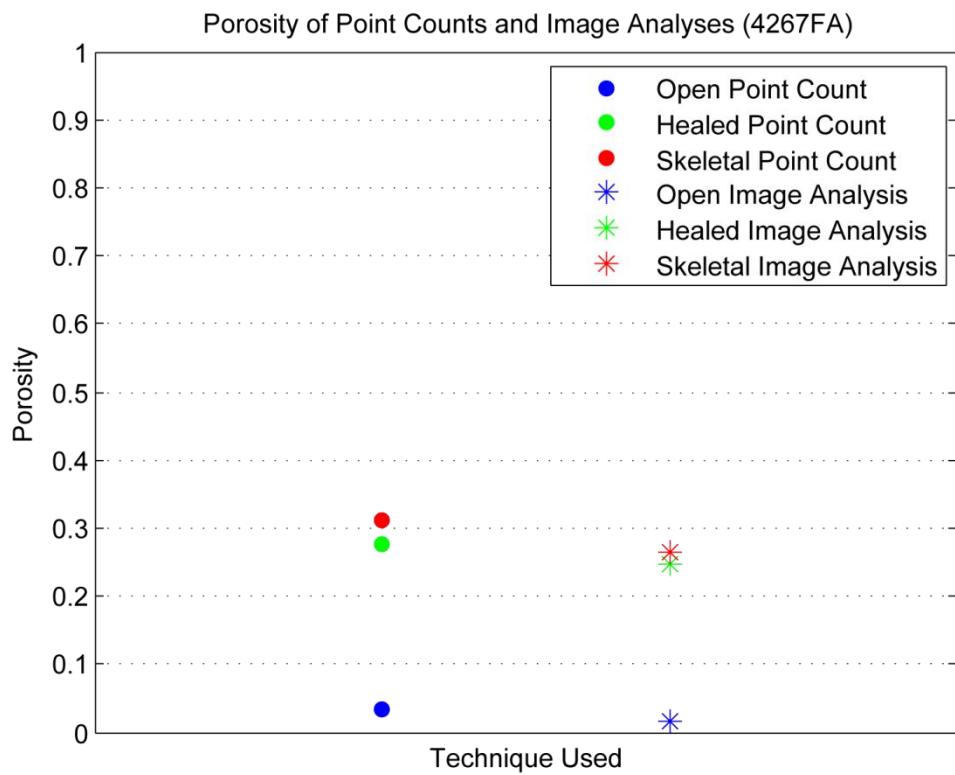
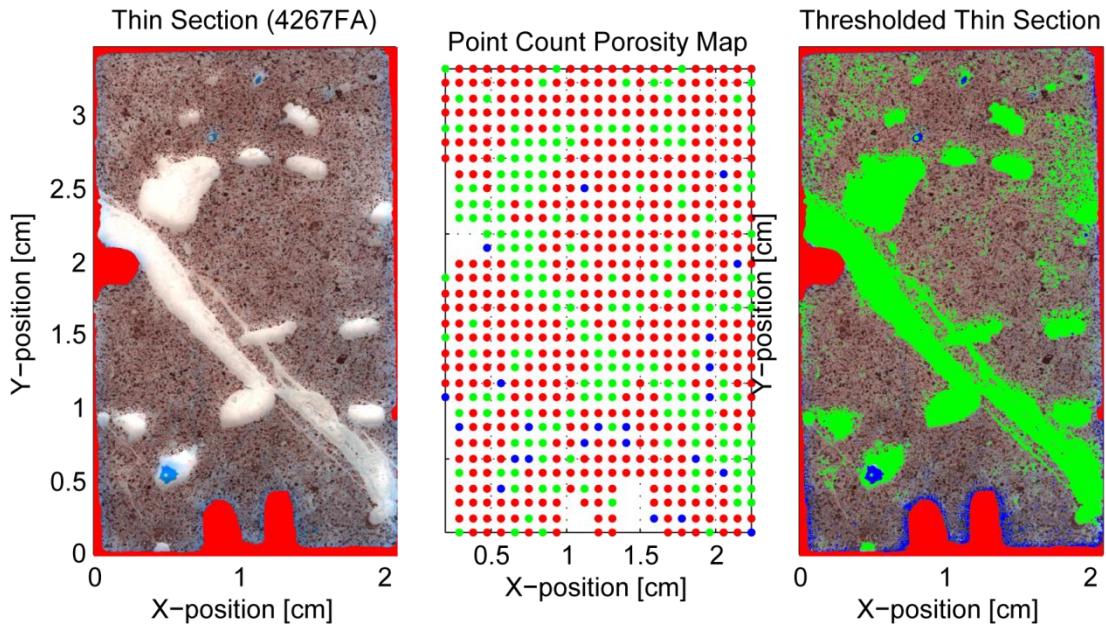


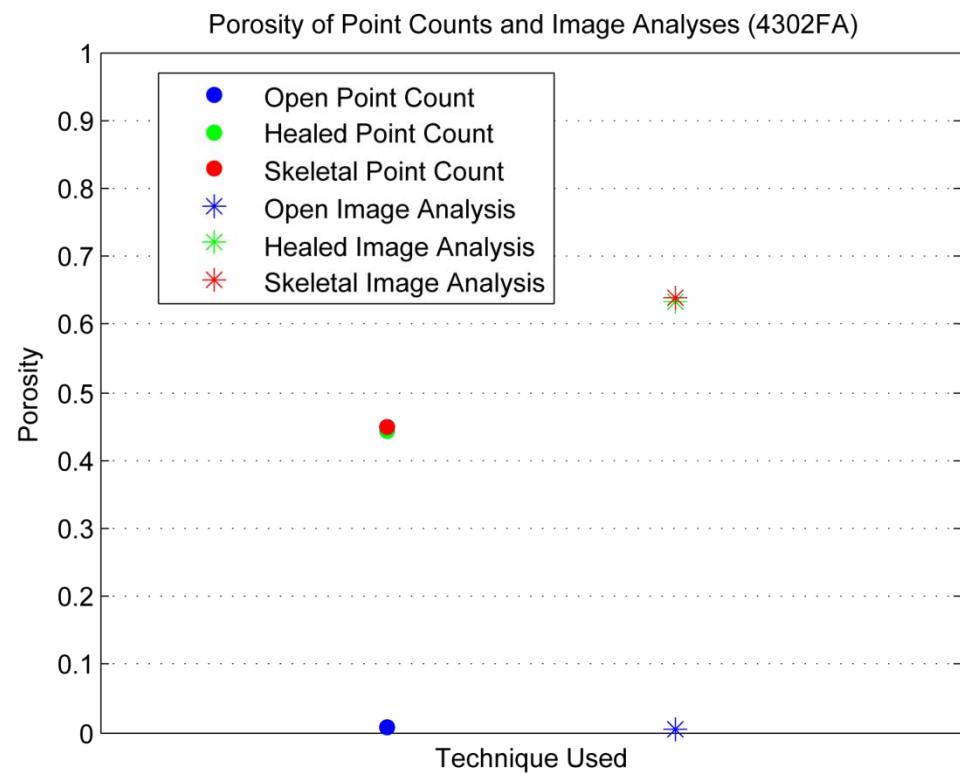
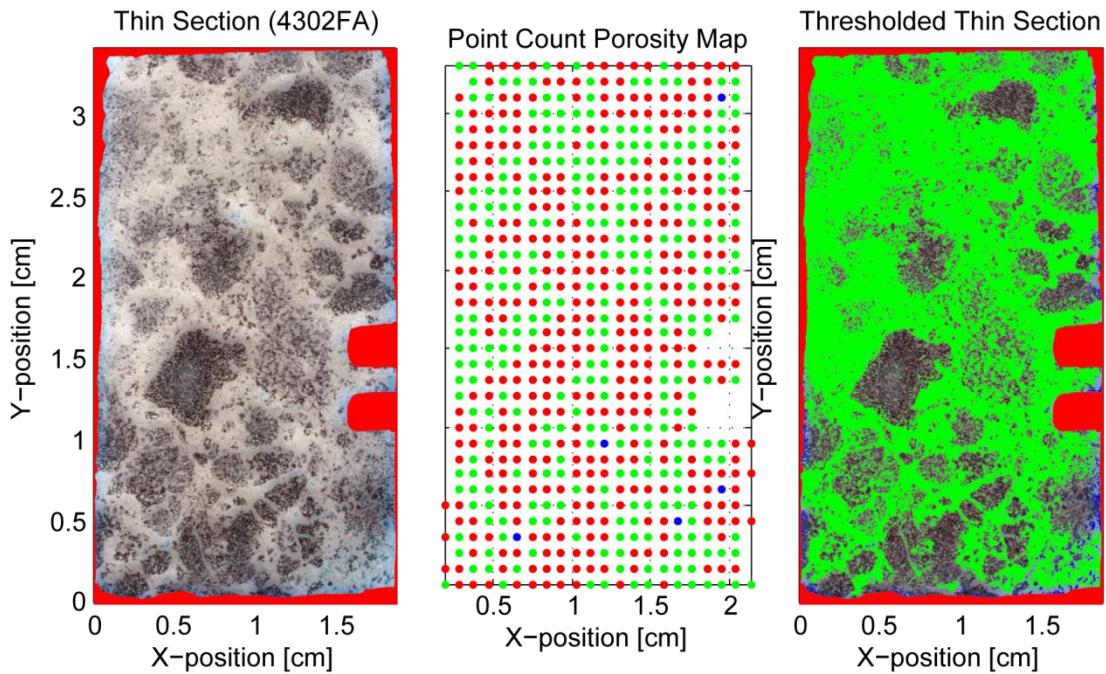


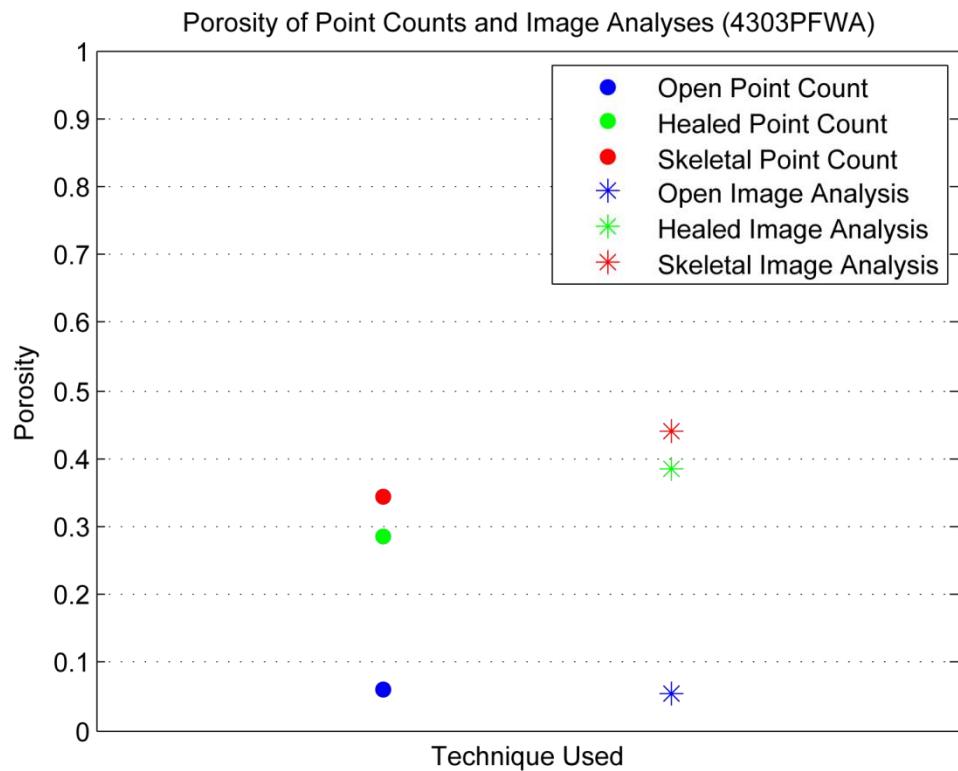
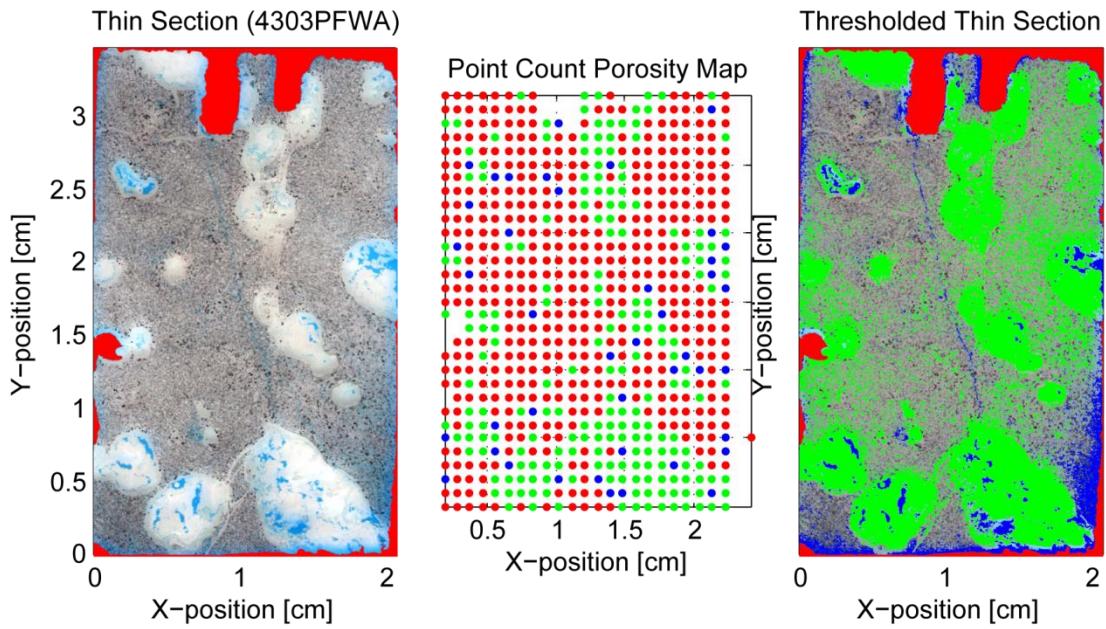


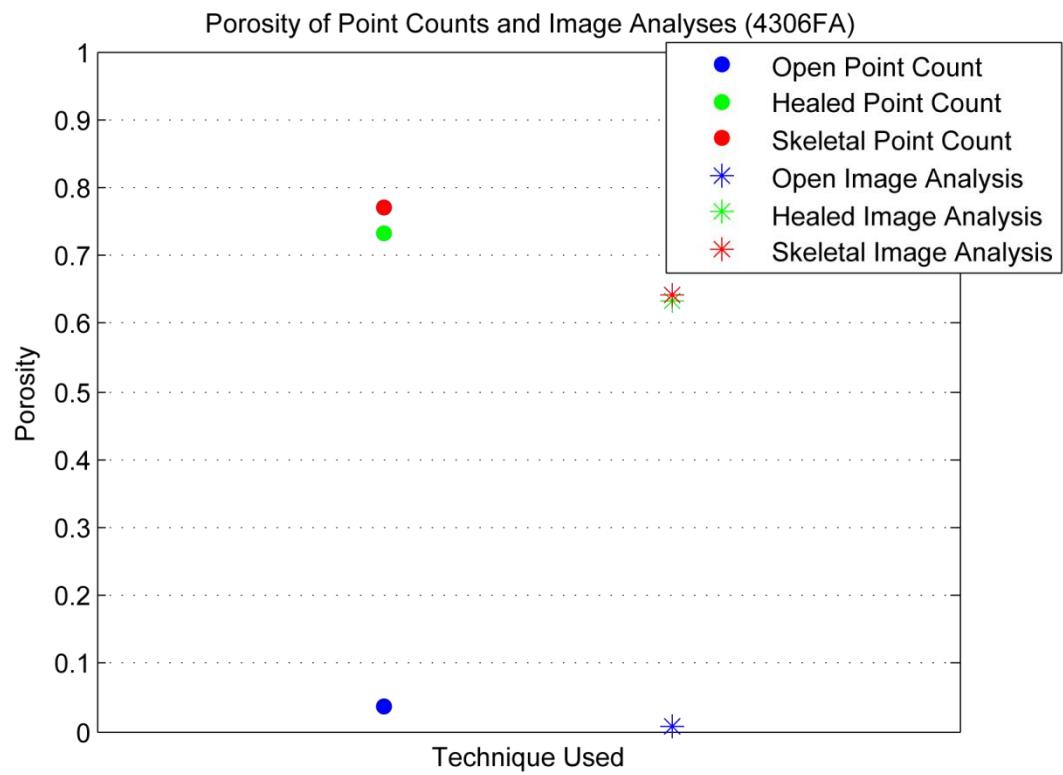
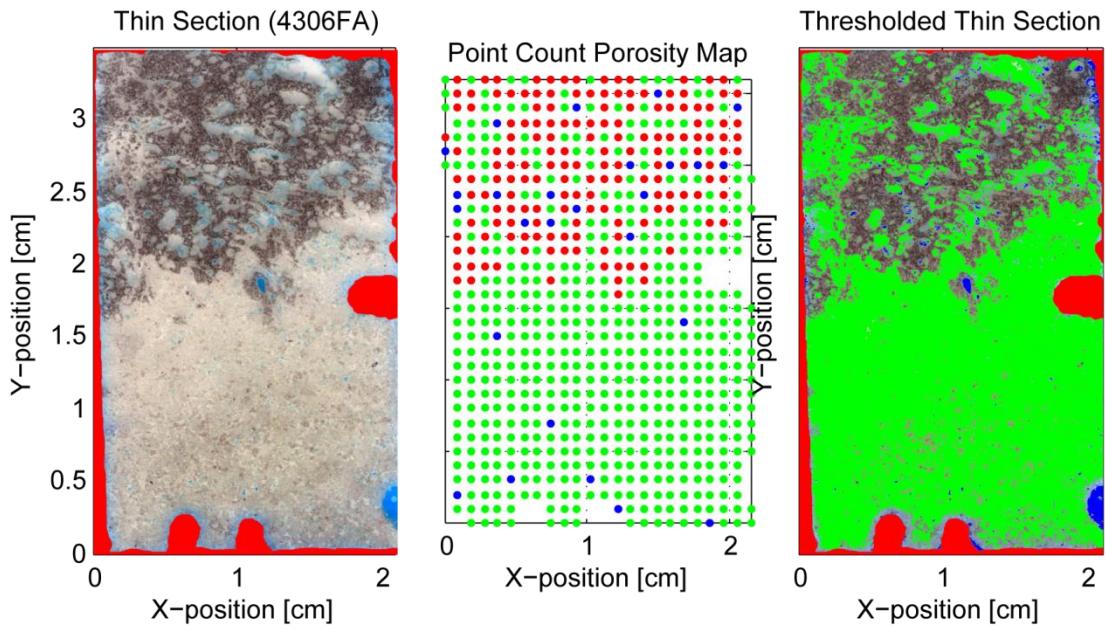






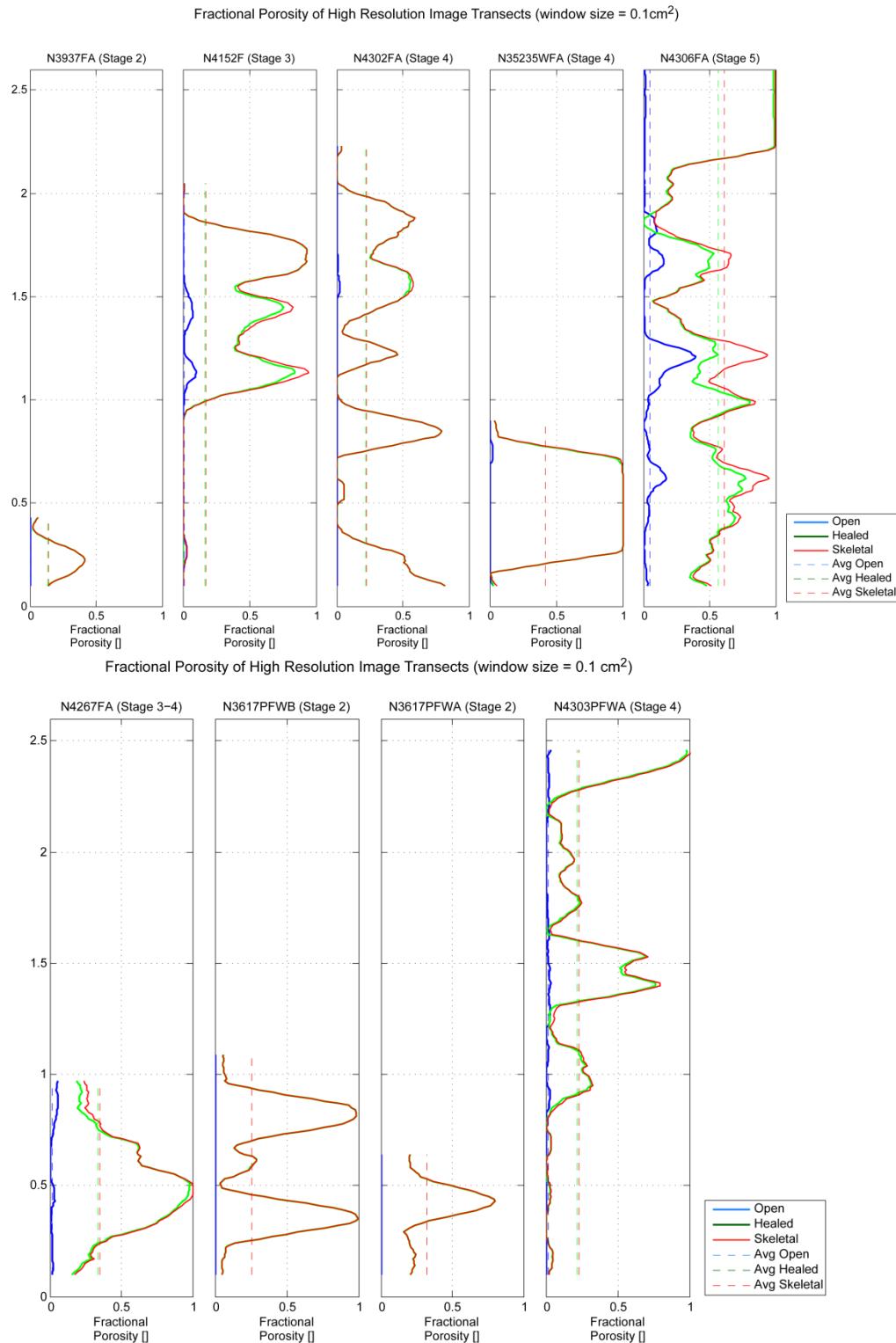




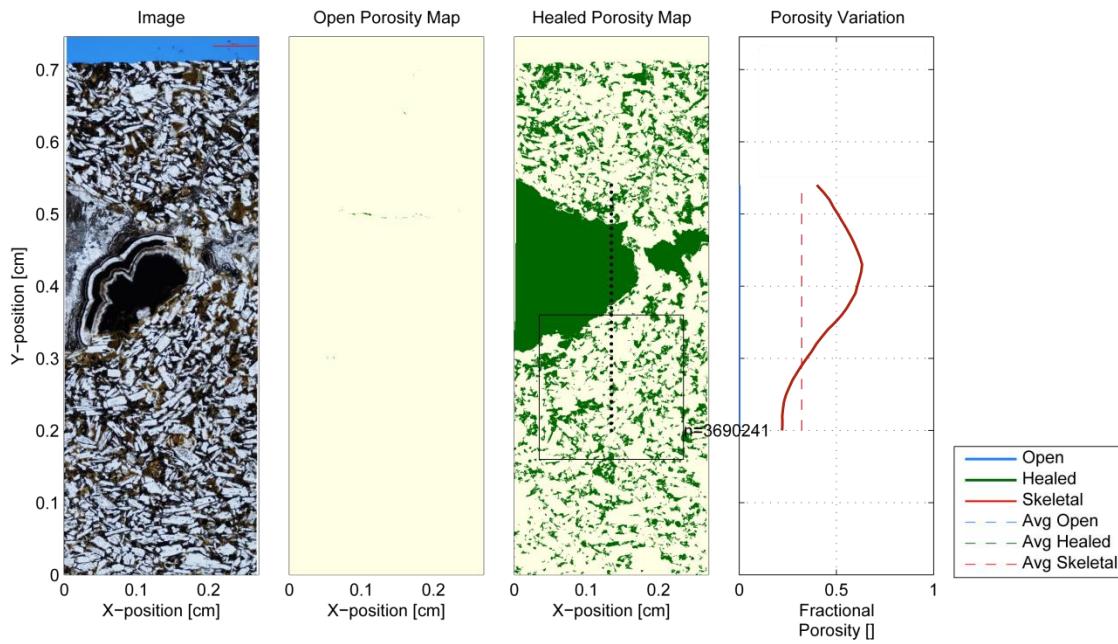


APPENDIX D

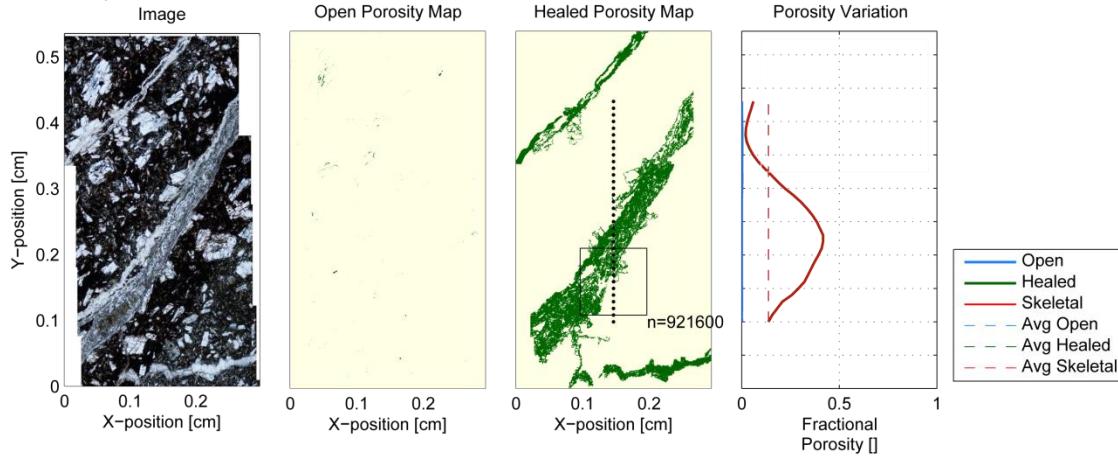
RESULTS OF HIGH RESOLUTION IMAGE ANALYSES

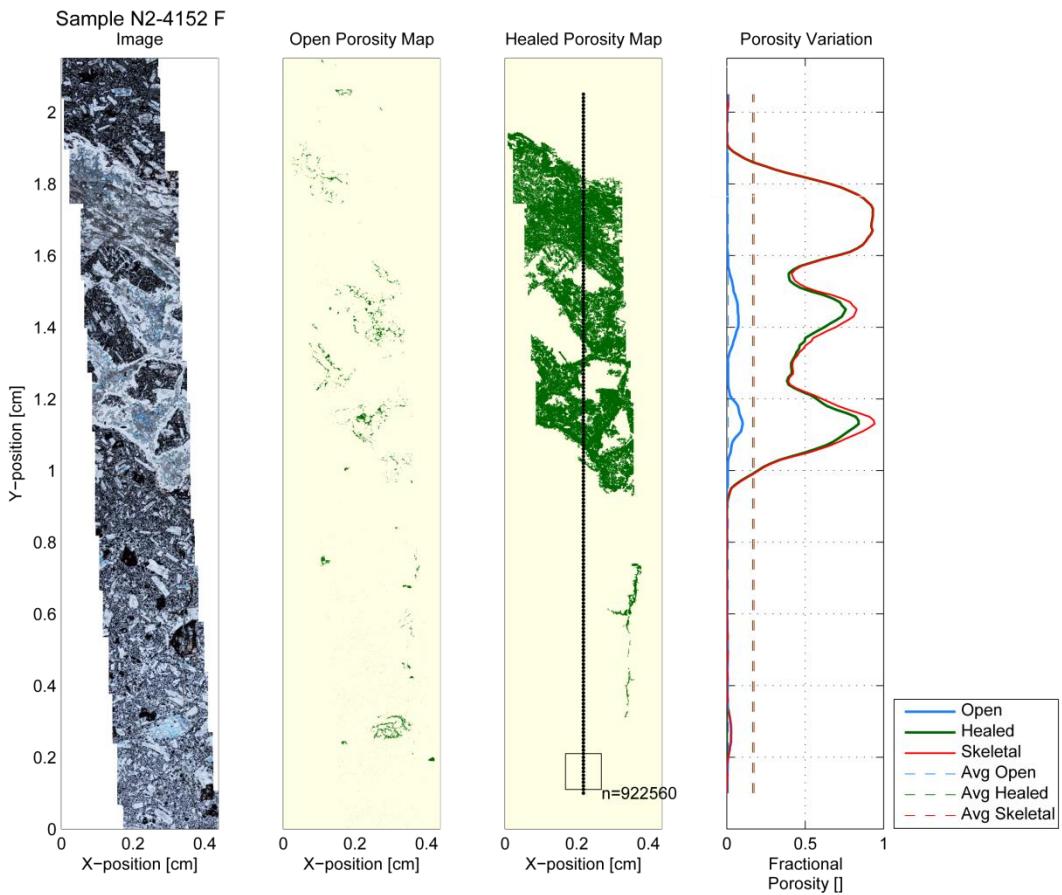


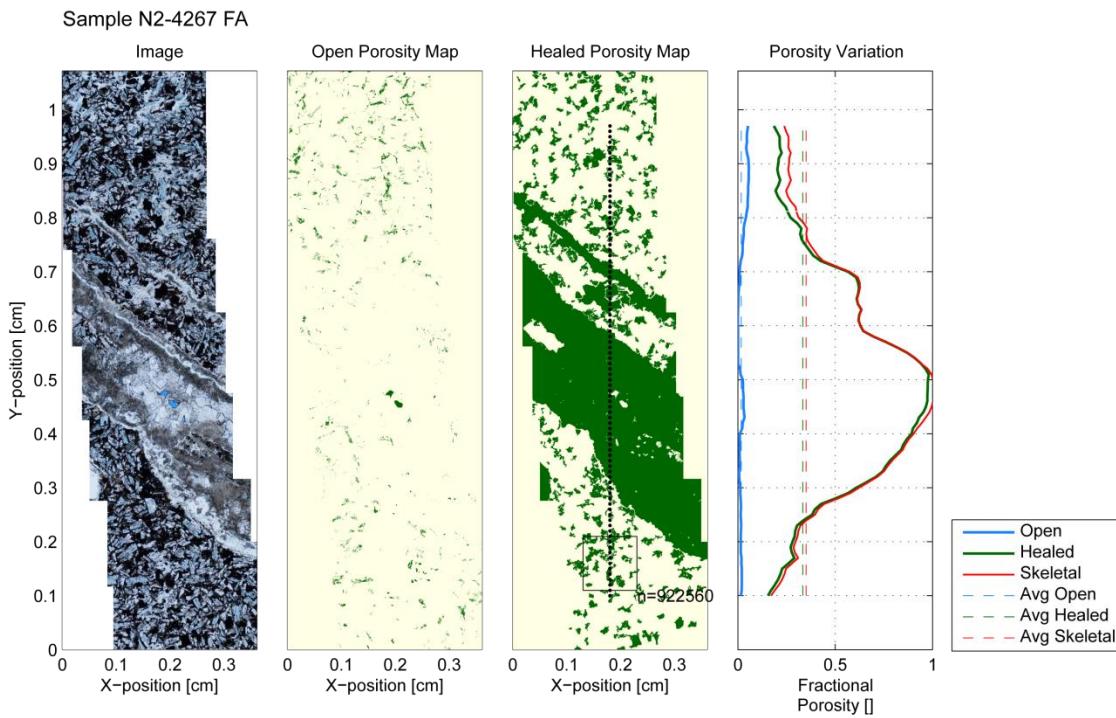
Sample N2-3617 PFWA

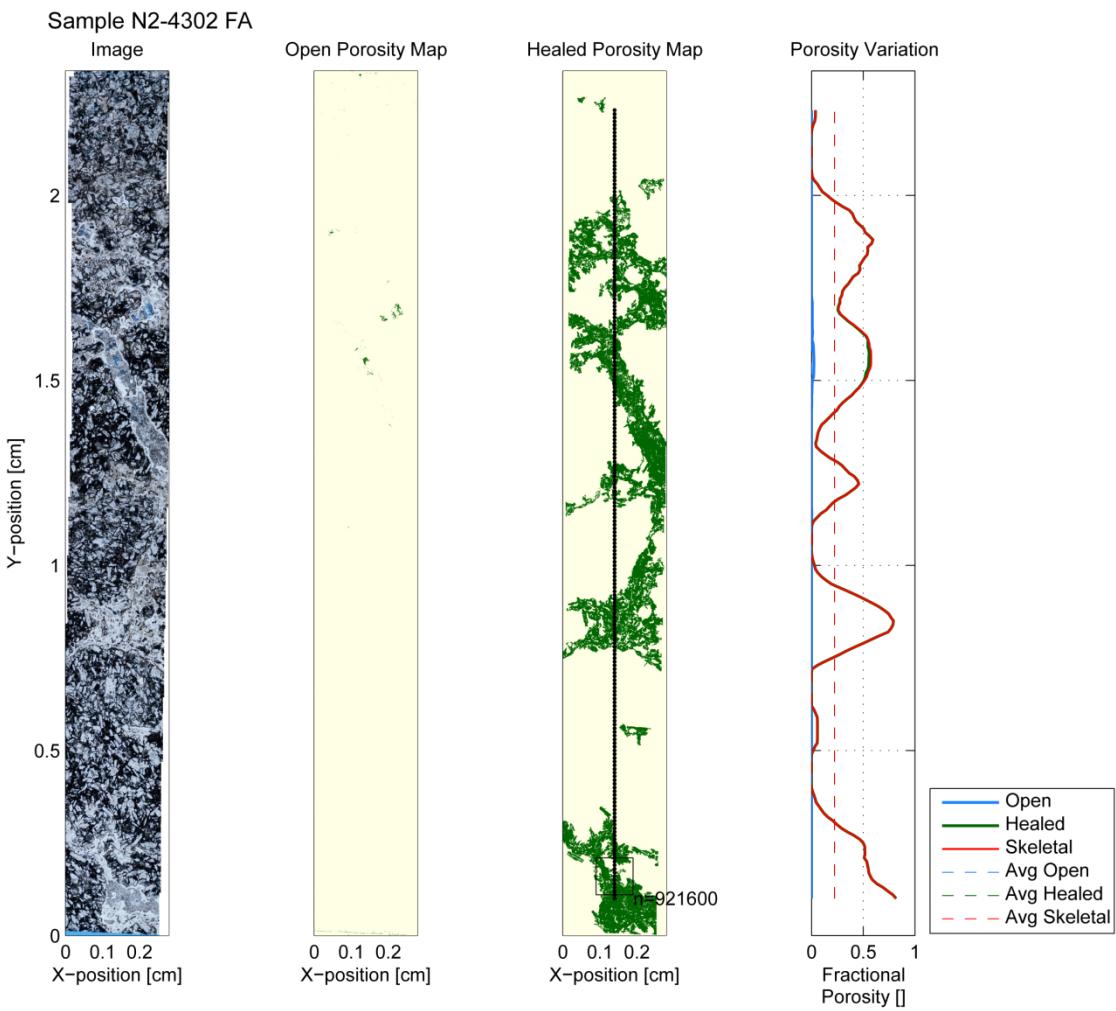


Sample N2-3937 FA

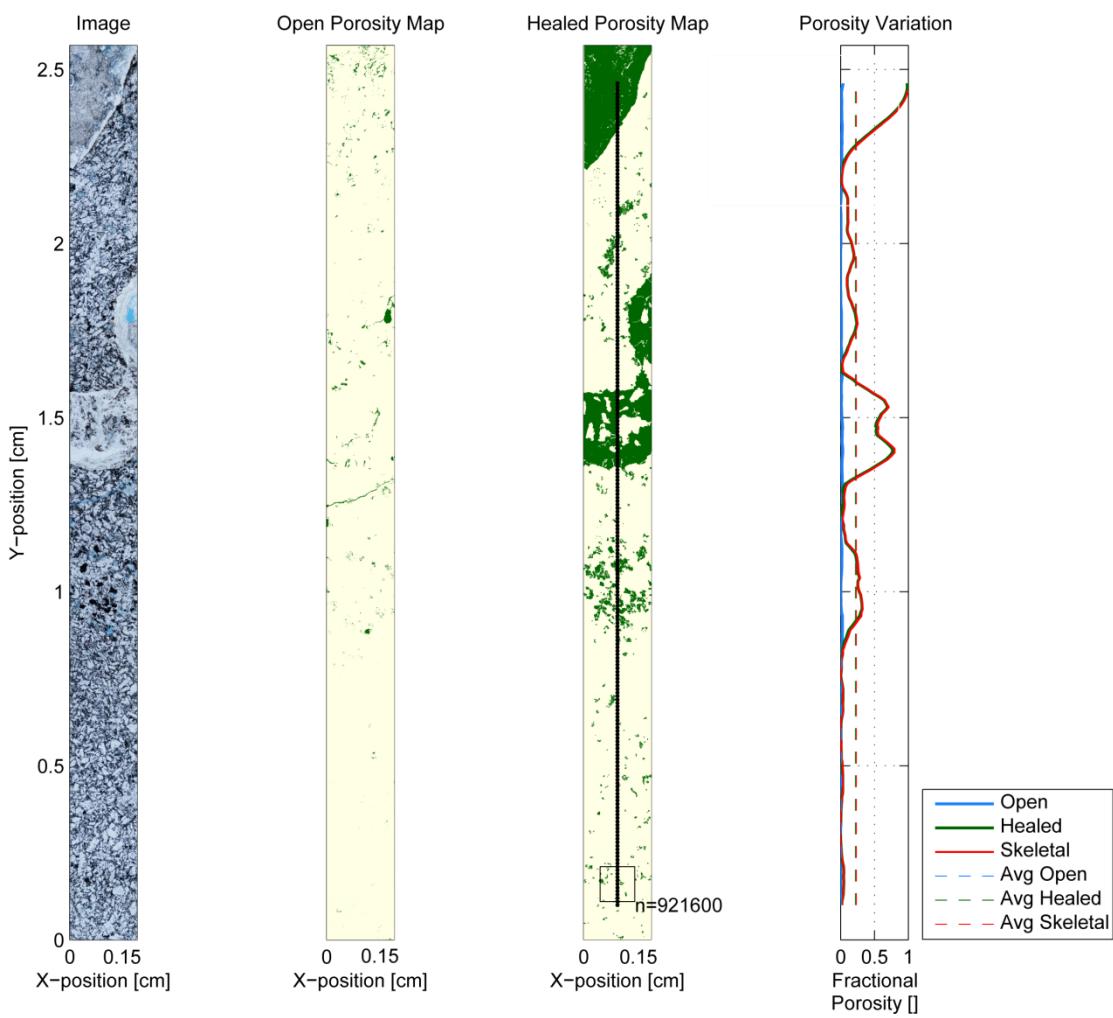




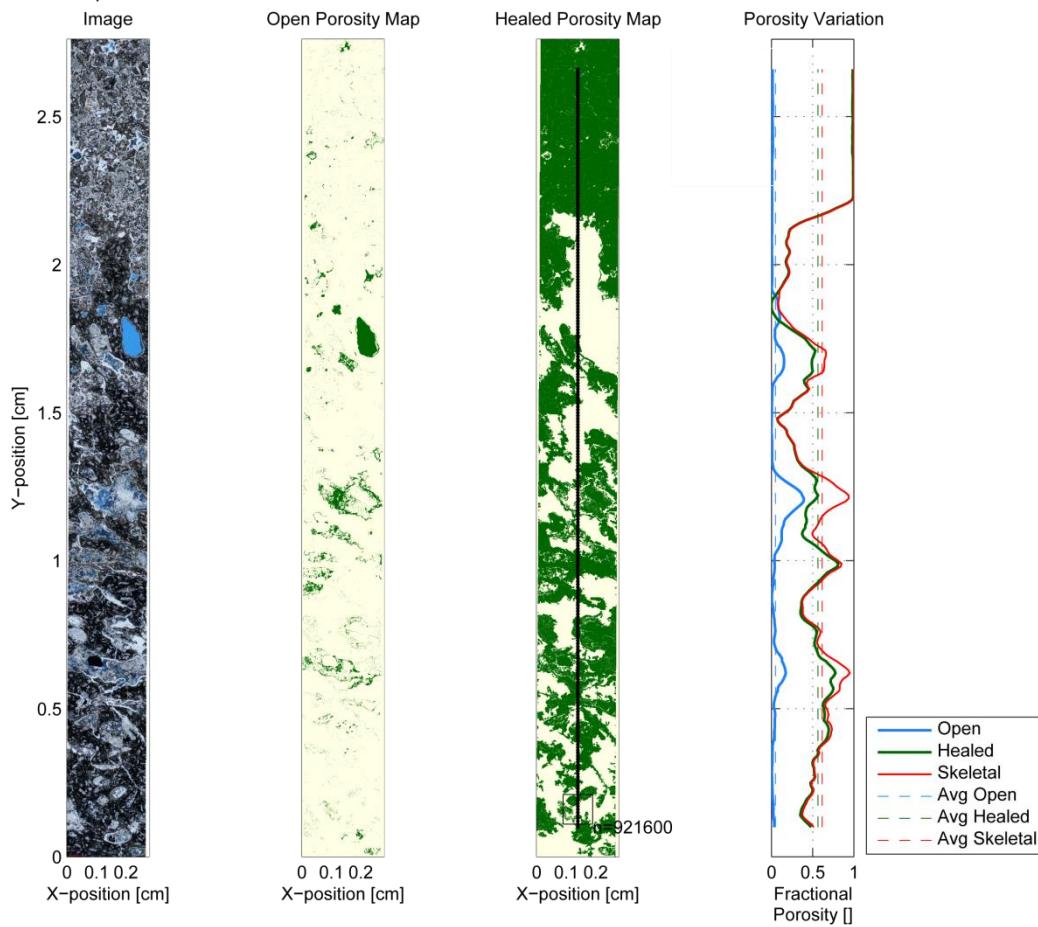




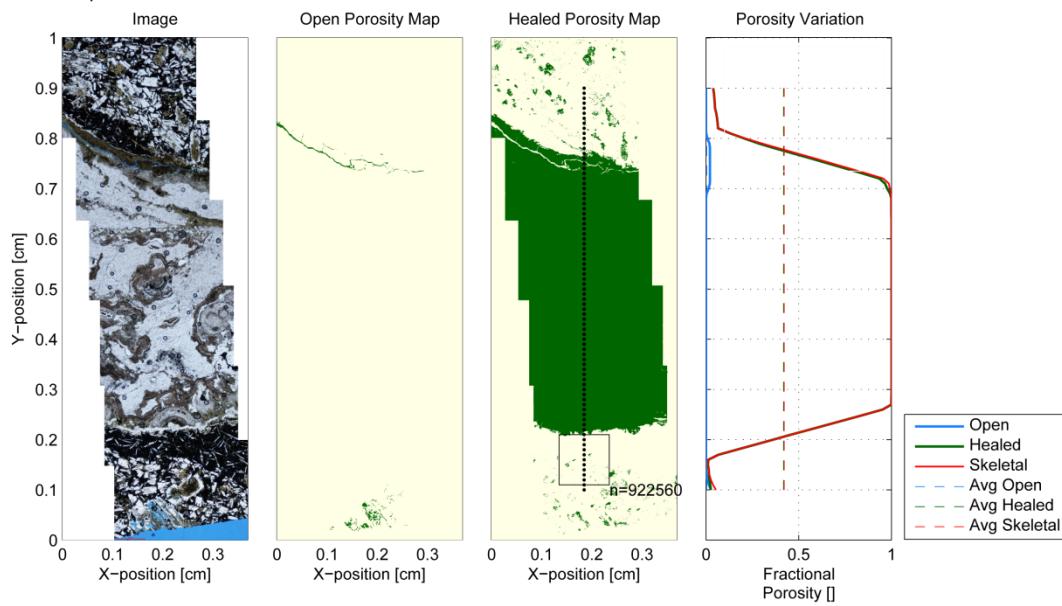
Sample N2-4303 PFWA



Sample N2-4306 FA



Sample N2-3523.5 WFA



APPENDIX E

MICRO CT SCAN RESULTS

